



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

**OFDM TABANLI HABERLEŞME
KANALLARINDA LDPC KODLAMA**

**Elektronik Yüksek Mühendisi Aziz YILMAZ
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Elektrik-Elektronik Programı**

**Danışman
Prof.Dr. Osman Nuri UÇAN**

Temmuz, 2010

İSTANBUL



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

**OFDM TABANLI HABERLEŞME
KANALLARINDA LDPC KODLAMA**

**Elektronik Yüksek Mühendisi Aziz YILMAZ
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Elektrik-Elektronik Programı**

**Danışman
Prof.Dr. Osman Nuri UÇAN**

Temmuz, 2010

İSTANBUL

Bu çalışma 19/07/2010 tarihinde ařağıdaki jüri tarafından Elektrik-Elektronik Mühendisliğı Anabilim Dalı Elektrik-Elektronik Mühendisliğı Programında Doktora Tezi olarak kabul edilmiştir.

Tez Jürisi

Prof.Dr.Osman Nuri UÇAN
İstanbul Üniversitesi
Mühendislik Mimarlık Fakültesi

Prof.Dr.Aydın AKAN
İstanbul Üniversitesi
Mühendislik Mimarlık Fakültesi

Prof.Dr.Sedef KENT
İstanbul Teknik Üniversitesi
Elektronik ve Haberleşme Fakültesi

Prof.Dr.İlhan KOCAARSLAN
İstanbul Üniversitesi
Mühendislik Mimarlık Fakültesi

Prof.Dr.Ali OKATAN
Karatay Üniversitesi
Mühendislik Mimarlık Fakültesi

ÖNSÖZ

Bu tezin hazırlanmasında, değerli bilgi ve tecrübelerini esirgemeyen, öğrenci motivasyonunu en iyi şekilde yansıtan ve bu çalışmaya ilgimin daha da artmasını sağlayan Değerli Hocam, *Prof.Dr.Osman Nuri UÇAN*'a teşekkürlerimi sunarım.

Ayrıca, bu çalışmanın ortaya çıkması için gerekli birikime sahip olmamda, büyük katkıları olan Sayın *Yrd.Doç.Dr.Onur OSMAN*'a ve yardımlarını esirgemeyen çalışma arkadaşlarım ile İstanbul Üniversitesi personeline teşekkürü bir borç bilirim.

Doktora süresince, maddi ve manevi destekleriyle hep yanımda olan aileme teşekkürler.

Temmuz, 2010

Aziz YILMAZ

İÇİNDEKİLER

ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİL LİSTESİ	v
TABLO LİSTESİ	vii
SEMBOL LİSTESİ	viii
ÖZET	ix
SUMMARY	x
1. GİRİŞ.....	1
2. GENEL KISIMLAR.....	3
2.1. LDPC KODLAMA.....	3
2.1.1. LDPC İle İlgili Kavramlar.....	3
2.1.1.1. Entropi ve Bilgi.....	3
2.1.1.2. Minimum Uzaklık.....	4
2.1.1.3. Bayes Teoremi	6
2.1.1.4. Saf Bayes Sınıflandırması.....	7
2.1.1.5. Bayes İnanç Ağları.....	9
2.1.1.6. Shannon Teoremi.....	9
2.1.1.7. Durma, Kod Kelimeleri, Uç Genişleme Kümeleri ve Çevrimler.....	12
2.1.1.8. Çevrimsiz Kümeler.....	15
2.1.1.9. EMD Kodlama.....	18
2.1.2. Kodlama Algoritmaları.....	19
2.1.2.1. Hırslı Algoritmalar.....	20
2.1.2.2. A Hırslı Algoritması.....	21

2.1.2.3. AH Hırslı Algoritması.....	23
2.1.2.4. AHT Hırslı Algoritması.....	23
2.1.3. LDPC Kodlama Yapısı.....	24
2.1.3.1. Hata Azaltımı İçin Parite Kontrol Matris Yapısı.....	26
2.1.3.2. Parite Kontrol (PC) Kodlama.....	27
2.1.3.3. Düşük Yoğunluklu Parite Kontrol Kodları.....	28
2.1.3.4. Düşük Yoğunluklu Parite Kontrol Kodlama.....	32
2.1.3.5. Düşük Yoğunluklu Parite Kontrol Kodlarının Çözülmesi.....	34
2.1.3.6. Kapasite Yaklaşımı Düzensiz LDPC Kodlarının Tasarımı.....	39
2.1.3.7. Kod Çözme Teorileri.....	42
2.1.3.8. İnanç Yayılımı.....	42
2.1.3.9. İnanç Yayılımının ve Yoğunluk Evriminin Asimptotik Analizi.....	45
2.1.3.10. Mesaj Geçen Yapı.....	46
2.1.3.11. Sert Karar Kod Çözme.....	48
2.1.3.12. Yumuşak Karar Kod Çözme.....	49
2.2. ÇOK SEVİYELİ LDPC.....	51
2.3. OFDM YAPISI.....	52
2.3.1. OFDM İle İlgili Kavramlar.....	52
2.3.1.1. Doppler Saçılımı.....	52
2.3.1.2. Kablosuz Ortamda İletim.....	53
2.3.1.3. Çok Yollu Sönümlenme Modeli.....	54
2.3.1.4. Gecikme Saçılması.....	55
2.3.1.5. Uyumluluk Bant Genişliği.....	55
2.3.1.6. Uyumluluk Süresi.....	56
2.3.1.7. PSK Yapıları.....	56
2.3.1.8. Semboller ve Taşıyıcılar Arası Girişim.....	58
2.3.2. OFDM.....	60
2.3.2.1. OFDM'in Tercih Nedenleri.....	60
2.3.2.2. Seri/Paralel Veri İletim Sistemleri.....	61
2.3.2.3. OFDM Tekniği.....	62
2.3.2.4. Alt Taşıyıcıların Üretilmesi.....	67
2.3.2.5. Koruma Süresi ve Periyodik İlaveler.....	69
2.3.2.6. Pencereleme.....	71

2.3.2.7. OFDM ve Diklik.....	73
2.3.2.8. OFDM Parametrelerinin Seçimi.....	74
2.3.2.9. OFDM Standartları.....	75
2.4. OFDM HABERLEŞME KANALLARINDA ÇOK SEVİYELİ	
LDPC KODLAMA.....	76
2.4.1. LDPC Kodlamalı OFDM.....	76
2.4.2. Çok Seviyeli LDPC Kodlamalı OFDM.....	79
3. MALZEME VE YÖNTEM	81
4. BULGULAR.....	82
4.1. ÇOK SEVİYELİ LDPC KODLAMA BENZETİMİ.....	82
4.1.1. Benzetim Ortamı.....	82
4.1.2. Programlama Akışı.....	82
4.1.3. Benzetim Sonucunda Elde Edilen Grafikler	85
4.1.4. Elde Edilen Bulgular.....	90
4.2. OFDM KODLAMA BENZETİMİ.....	92
4.2.1. Benzetim Ortamı.....	92
4.2.2. Programlama Akışı.....	92
4.2.3. Benzetim Sonucunda Elde Edilen Grafikler	96
4.2.4. Elde Edilen Bulgular.....	103
4.3. ÇOK SEVİYELİ LDPC KODLAMALI OFDM BENZETİMİ.....	105
4.3.1. Benzetim Ortamı.....	105
4.3.2. Programlama Akışı.....	106
4.3.3. Benzetim Sonucunda Elde Edilen Grafikler	110
4.3.4. Elde Edilen Bulgular.....	115
5. TARTIŞMA VE SONUÇ.....	117
KAYNAKLAR.....	122
ÖZGEÇMİŞ.....	125

ŞEKİL LİSTESİ

Şekil 2.1:	İki kanal örneği (BEC ve BSC).....	11
Şekil 2.2:	Matris ve (9,3) kodun grafiksel gösterimi.....	12
Şekil 2.3:	Katkılı mesaj ve Graf genişlemesi.....	14
Şekil 2.4:	C_d, S_d, W_d, E_d arasındaki ilişkiyi gösteren Venn Diyagramı.....	15
Şekil 2.5:	Birçok çevrimi ortadan kaldıran geleneksel çevre koşullandırması.....	15
Şekil 2.6:	Diyagonal Matris.....	16
Şekil 2.7:	v_1 'in iki tane 2-dereceli düğümler tarafından yer değiştirmesi.....	17
Şekil 2.8:	Çevrim içindeki demet tarafından v_1 'in yer değiştirmesi.....	18
Şekil 2.9:	Örnek hızlı algoritma yapısı.....	20
Şekil 2.10:	Hırslı Algoritma.....	21
Şekil 2.11:	A Hırslı Algoritması.....	21
Şekil 2.12:	Doğrusal blok kodlayıcı yapısı.....	25
Şekil 2.13:	Düzenli (3,4) LDPC, “Tanner Graph” gösterimi.....	29
Şekil 2.14:	H matrisi ve “Tanner Graph” gösterim.....	29
Şekil 2.15:	Düzenli-Düzensiz LDPC, “Tanner Graph” karşılaştırma.....	31
Şekil 2-16:	Bir LDPC Kodun matematiksel ifadesi.....	38
Şekil 2.17:	Statik bir kanalda LDPC kodlama performansı.....	38
Şekil 2.18:	Kod çözme algoritması.....	47
Şekil 2.19:	Yumuşak karar çözme yapısı.....	49
Şekil 2.20:	Doppler saçılımı.....	53
Şekil 2.21:	Kablosuz haberleşmede çoklu yol gecikmesi.....	53
Şekil 2.22:	BPSK, QPSK ve 8-PSK sembol konumları.....	57
Şekil 2.23:	Semboller arası girişim (ISI).....	58
Şekil 2.24:	“Cyclic Prefix” ile bir sembolün periyodik genişlemesi.....	59
Şekil 2.25:	Çoklu yol gecikme süresinin ‘CP’ den kısa olması.....	59
Şekil 2.26:	Tek taşıyıcılı sistem.....	61
Şekil 2.27:	Çok taşıyıcılı system.....	62
Şekil 2.28:	Genel OFDM yapısı.....	63
Şekil 2.29:	Modüleli sinyale IFFT uygulanması.....	64
Şekil 2.30:	Geleneksel ve dik çok-taşıyıcı yapısı.....	65
Şekil 2.31:	OFDM sinyal spektrumu.....	66
Şekil 2.32:	OFDM modülatörü.....	68
Şekil 2.33:	Üç alt-taşıyıcılı OFDM sinyali.....	68
Şekil 2.34:	Koruma zamanı olmamasının çok-yollu ortamlardaki etkisi.....	69
Şekil 2.35:	Periyodik örnekli OFDM sembolü.....	70
Şekil 2.36:	İki yollu üç alt taşıyıcılı OFDM sinyal örneği.....	71
Şekil 2.37:	OFDM periyodik eki ve pencereleme.....	72
Şekil 2.38:	OFDM ve diklik (orthogonality) prensibi.....	73
Şekil 2.39:	Farklı kodlamalı OFDM sistem performansları.....	78
Şekil 2.40:	İki seviyeli LDPC kodlamalı OFDM.....	79
Şekil 2.41:	Üç Seviyeli LDPC Kodlamalı OFDM.....	80

Şekil 3.1:	Benzetim sistemi blok diyagram.....	81
Şekil 3.2:	İki seviyeli LDPC (AWGN, Ga).....	86
Şekil 3.3:	İki seviyeli LDPC (AWGN, Gb).....	86
Şekil 3.4:	İki seviyeli LDPC (AWGN, SED).....	86
Şekil 3.5:	Üç seviyeli LDPC (AWGN, Ga).....	86
Şekil 3.6:	Üç seviyeli LDPC (AWGN, Gb).....	86
Şekil 3.7:	Üç seviyeli LDPC (AWGN, SED).....	86
Şekil 3.8:	İki seviyeli LDPC (Rayleigh, Ga).....	87
Şekil 3.9:	İki seviyeli LDPC (Rayleigh,Gb).....	87
Şekil 3.10:	İki seviyeli LDPC (Rayleigh,SED).....	87
Şekil 3.11:	Üç seviyeli LDPC (Rayleigh,Ga).....	87
Şekil 3.12:	Üç seviyeli LDPC (Rayleigh, Gb).....	87
Şekil 3.13:	Üç seviyeli LDPC (Rayleigh, SED).....	87
Şekil 3.14:	İki seviyeli LDPC (Rician K=10, Ga).....	88
Şekil 3.15:	İki seviyeli LDPC (Rician K=10, Gb).....	88
Şekil 3.16:	İki seviyeli LDPC (Rician K=10, SED).....	88
Şekil 3.17:	Üç seviyeli LDPC (Rician K=10, Ga).....	88
Şekil 3.18:	Üç seviyeli LDPC (Rician K=10, Gb).....	88
Şekil 3.19:	Üç seviyeli LDPC (Rician K=10, SED).....	88
Şekil 3.20:	İki seviyeli LDPC (Rician K=20, Ga).....	89
Şekil 3.21:	İki seviyeli LDPC (Rician K=20, Gb).....	89
Şekil 3.22:	İki seviyeli LDPC (Rician K=20, SED).....	89
Şekil 3.23:	Üç seviyeli LDPC (Rician K=20, Ga).....	89
Şekil 3.24:	Üç seviyeli LDPC (Rician K=20, Gb).....	89
Şekil 3.25:	Üç seviyeli LDPC (Rician K=20, SED).....	89
Şekil 3.26:	BPSK modülasyonlu OFDM'e Pilot Sembol etkisi.....	97
Şekil 3.27:	BPSK modülasyonlu OFDM'e Doppler etkisi.....	98
Şekil 3.28:	QPSK modülasyonlu OFDM'e Pilot Sembol etkisi.....	99
Şekil 3.29:	QPSK modülasyonlu OFDM'e Doppler etkisi.....	100
Şekil 3.30:	8PSK modülasyonlu OFDM'e Pilot Sembol etkisi.....	101
Şekil 3.31:	8PSK modülasyonlu OFDM'e Doppler etkisi.....	102
Şekil 3.32:	İki seviyeli LDPC kodlamalı OFDM'e Pilot Sembol etkisi.....	111
Şekil 3.33:	İki seviyeli LDPC kodlamalı OFDM'e Doppler Etkisi.....	112
Şekil 3.34:	Üç seviyeli LDPC kodlamalı OFDM'e Pilot Sembol Etkisi.....	113
Şekil 3.35:	Üç seviyeli LDPC Kodlamalı OFDM'e Doppler Etkisi.....	114

TABLO LİSTESİ

Tablo 2.1: Tüm bit düğümleri için olasılık değerleri.....	47
Tablo 2.2: Sert karar kod çözme.....	48
Tablo 2.3: Sert karar kod çözme akışı.....	48
Tablo 3.1: OFDM parametreleri.....	105

SEMBOL LİSTESİ

LDPC	: Low Density Parity Check - Düşük Yoğunluklu Parite Kontrol
BSC	: Binary Symetric Channel - İkili Simetrik Kanal
BEC	: Binary Channel - İkili Kalıntı Kanal
ML	: Maximum Likelihood – Maksimum Yakın Komşuluk
EMD	: Extrinsic Message Degree – Katkılı Mesaj Derecesi
BP	: Belief Propagation - İnanç Yayılımı
LLR	: Log-Likelihood Ratio - Logaritmik Olasılık Oranları
BPA	: Belief Propagation Algorithm - İnanç Yayılım Algoritması
MPA	: Message Passing Algorithm - Mesaj Geçen Algoritma
AI	: Artificial intelligence – Yapay Zeka
C_d	: Çevrim kümesi
S_d	: Durma kümesi
W_d	: Kod kelime kümesi
E_d	: Uç genişleme kümesi
CPT	: Conditional Probability Table - Koşulu olasılık tablosu
B_D	: Spektral genişleme ölçüsü
f_c	: Taşıyıcı frekansı
T_d	: Doppler kanal zamanı
T_s	: Gönderilen sinyal zamanı
S_b	: Band geçiren sinyal
B_c	: Uyumluluk bant genişliği
PSK	: Phase Shift Keying
BPSK	: Binary Phase Shift Keying
QPSK	: Quaternary Phase Shift Keying
SNR	: Signal to Noise Ratio
ISI	: Inter Symbol Interference
ICI	: Inter Carrier Interference – Dahili Taşıyıcı Girişimi
CP	: Cyclic Prefix
FDM	: Frequency Division Multiplex – Frekans Bölümlemeli Çoğullama
OFDM	: Orthogonal Frequency Division Multiplex – Dik Frekans Böl. Çoğ.
ADSL	: Asymmetric Digital Subscriber Line – Asimetrik Dijital Abone Hattı
ANSI	: American National Standards Institute – Amerikan Ulusal St.Enst.
FEC	: Forward Error Correction – İleri Hata Düzeltme
FFT	: Fast Fourier Transform
IFFT	: Inverse Fast Fourier Transform
DFT	: Discrete Fourier Transform
IDFT	: Inverse Discrete Fourier Transform
VLSI	: Çok-geniş-ölçülü bütünleşme

ÖZET

OFDM TABANLI HABERLEŞME KANALLARINDA LDPC KODLAMA

İletilmesi istenen verinin, farklı gürültülerden oluşan bir ortamdan geçerek vericiye en az hata ile ulaşması ve kullanılabilir frekans bandına ilişkin kısıtlamalara rağmen daha fazla miktarda veri gönderilmesi ihtiyacı nedeniyle, her geçen gün yeni teknik ve uygulamaların geliştirilmesi gerekmektedir.

Tek-taşıyıcılı sistemlerde, bir olumsuz etki tüm iletim hattının etkileyerek iletişimi bozabilecek iken, çok-taşıyıcılı sistemler (OFDM) kullanılarak, sadece alt-taşıyıcıların çok az bir bölümünün etkilenmesi ve veri iletiminin daha az riskli bir şekilde gerçekleştirilebilmesi sağlanmaktadır. Ayrıca çok yollu sönümleme ve gecikme yayılımı ile frekans spektrumundaki alt taşıyıcıların birbirine dikliği sağlanarak çakışmayı en az seviyeye indiren bir tekniğin avantajlarından yararlanılmaktadır.

Gönderilmesi gereken verinin bit seviyesinde kontrol edilmesi ile veri iletim hata oranını en iyi seviyeye indirebilme imkanı sağlayan LDPC kodlama tekniğinin, çok seviyeli olarak kullanılması ile paralel kollardan gönderilen verinin sembollere dönüştürülmesi sağlanmakta ve neredeyse aynı zaman dilimi içinde daha fazla verinin iletilebilmesi sağlanmaktadır.

Bu çalışmada, öncelikle LDPC kodlama benzetimi yapılarak hata başarımları gözlenmiştir. Daha sonra OFDM yapısının benzetimi gerçekleştirilmiş ve hata başarımları elde edilmiştir. Nihayetinde, OFDM tabanlı haberleşme kanallarında LDPC kodlama yapılmış, hata başarımlarının daha iyi düzeyde elde edilmesine ve daha fazla verinin iletilebilmesine olanak sağlayan, Çok Seviyeli tasarım gerçekleştirilmiştir.

SUMMARY

LDPC CODING OVER OFDM BASED COMMUNICATION CHANNELS

There are more limitations over communication environment related to frequency band and different noise effects. Although having that kind of obstacles, it is desired to transfer more data with good error rates. Because of that, new techniques and applications required to be improved for this purpose.

A negative effect over single carrier communication channels may be able to cause crucial effect of whole communication channel. It is possible to prevent these communication channels from negative effects by using multi-carrier system (OFDM). This system, prevent almost all sub-carriers and provide more safe environment for data transfer and it is also preven from multi-direction fading and delay propagation effects by having orthogonality between sub-carriers in frequency spectrum.

LDPC coding provides bit level control for data which is transferred as blocks. For that reason, it provides best error rate performance. It is possible to send more data in paralell lines by using this technique with multilayer approach.

In this study, first LDPC coding simulation environment established and error rate performans observed. Later, simulation environment of OFDM system realized and error rate performans of this system examined. Finaly, a Multi Level LDPC coded OFDM (COFDM) simulation environment established, which has good error rate performans and more data transfer capability.

1. GİRİŞ

Gelişen teknolojik imkanlar ile birlikte artan veri iletimi ve frekans trafiği, haberleşme sistemlerinin sürekli olarak daha da gelişmesini gerektirmektedir. Alıcıdan gönderilen verinin farklı gürültülerden oluşan bir ortamdan geçerek vericiye en az hata ile ulaşması ve kullanılabilir frekans bandına ilişkin kısıtlamalara rağmen daha fazla miktarda veri gönderilmesi ihtiyacı nedeniyle, her geçen gün yeni teknikler ve uygulamalar geliştirilmektedir.

Günümüzde gezgin iletişim sistemleri, internet, video ve çoklu ortam veri aktarımının desteklenebilmesi için klasik yöntemler ile çözüm bulunması imkansız hale gelmiştir. Yüksek hızdaki verilerin hareketli platformlardan, arzu edilen servis kalitesinde iletilmesi için çağın gelişen alt yapısı da kullanılarak yeni modülasyon tekniklerinin kullanılması kaçınılmaz bir gerçek haline gelmiştir.

Kablosuz iletim kanallarında en az hata ile veri transferi, gelişen yeni modülasyon teknikleri ve kodlama teknikleri ile gün geçtikçe daha iyi seviyeye gelmiştir. Bu kodlama teknikleri; haberleşme ortamlarındaki gürültü ve sönümlenme etkisinin en aza indirilebilmesi ve alıcıda daha iyi bir sinyal elde edilebilmesi amacıyla kullanılmaktadır.

Tek-taşıyıcılı sistemlerde, bir sönümlenme ya da bozucu etki tüm iletim hattının etkileyerek iletimi bozabilecek iken, çok-taşıyıcılı sistemler (OFDM) kullanılarak, sadece alt-taşıyıcıların çok az bir bölümü etkilenmekte ve veri iletiminin daha az riskli bir şekilde gerçekleştirilmesi sağlanmaktadır. Ayrıca çok yollu sönümlenme ve gecikme yayılımı ile frekans spektrumundaki alt taşıyıcıların birbirine dikliği sağlanarak çakışmayı en az seviyeye indiren bir tekniğin avantajlarından yararlanılmaktadır.

Gönderilmesi gereken verinin bit seviyesinde kontrol edilmesi ile veri iletim hata oranını istenen seviyeye indirilebilmesi imkanını sağlayan LDPC kodlama tekniğinin çok seviyeli olarak kullanılması ile paralel kollardan gönderilen verinin sembollere dönüştürülmesi sağlanmakta ve neredeyse aynı zaman dilimi içinde daha fazla verinin iletilmesi sağlanmaktadır.

Bu tez çalışmasında, frekans bandının daha etkin kullanılması amacıyla geliştirilmiş bir teknik olan Dik Frekans Bölümlenmeli Çoğullama (OFDM-Orthogonal Frequency Division Multiplex) ile daha az hatalı veri transferinin daha kısa zamanda sağlanması amacıyla, Çok seviyeli Düşük Yoğunluklu Parite Kontrol (Multilevel LDPC-Low Density Parity Check) kodlamanın birlikte kullanılması amaçlanmış ve bilgisayar ortamında benzetim yapılarak hata başarımları performansları incelenmiştir.

Haberleşme literatüründe, LDPC kodlamalı OFDM ile ilgili çalışmalar bulunmaktadır. Ancak, Çok Seviyeli LDPC Kodlamalı OFDM (MLCOFDM-Multilevel Coded OFDM) ile ilgili bir çalışmaya rastlanmamıştır. Bu nedenle yapılan çalışma ve sonuçlarının dikkat çekici olacağı değerlendirilmektedir.

Tez dokümanında, tez konusu teknikler ile birlikte kullanılan tanımların/yapıların daha iyi anlaşılabilmesine de olanak sağlanmasının yararlı olacağı düşünülmüştür. Bu nedenle söz konusu tanımlar ve yapıların mümkün olduğunca açıklanmasına özen gösterilerek, tez konusunun ve yapılan çalışmaların daha iyi anlaşılması amaçlanmıştır.

Tez konusu ile ilişkili, bazı tanımlar ve yapılar ile ilgili açıklamalar yapıldıktan sonra, ilk aşamada farklı gürültü yapıları, değişik üreteç matrisleri, farklı seviyeler (iki ve üç seviyeli) ve farklı SNR değerlerinde, Çok Seviyeli LDPC Kodlama benzetimi yapılarak hata başarımlarına yönelik elde edilen bulgular değerlendirilmiştir. Daha sonra frekans seçici düzgün sönümlenmeyen kanallarda OFDM benzetimi yapılarak doppler ve pilot sembol etkisiyle elde edilen bulgular değerlendirilmiştir. Son olarak, yeni bir teknik olan Çok Seviyeli LDPC Kodlamalı OFDM benzetimi yapılarak, elde edilen bulgular değerlendirilerek tartışma ve sonuçlar ile tez dokümanı tamamlanmıştır.

2. GENEL KISIMLAR

Sayısal haberleşmenin en büyük avantajlarından biri, değişik matematiksel metodlar ile farklı kodlamaların elde edilebilmesine olanak sağlamasıdır. Bu bölümde tezin konusuna uygun olarak tasarlanmış kodlamalara ilişkin bilgi verilecektir.

2.1. LDPC KODLAMA

LDPC Kodlamanın anlaşılabilmesi için, bu kodlama tekniğinde kullanılan bazı kavram ve tanımların bilinmesi gerekmektedir. Bu nedenle öncelikle, ilgili tanım ve kavramlar açıklanmıştır.

2.1.1 LDPC İle İlgili Kavramlar

2.1.1.1. Entropi ve Bilgi

Entropi bazı olaylar için kesin olmayan durumları en iyi açıklayan ölçü olarak kabul edilir. Herhangi bir rastgele değişkenin belirsizliğidir. Rastgele değişken biliniyor ise entropi değeri sıfır olarak kabul edilir. Eğer ayrık rastgele değişken X olarak düşünüldüğüne Entropi, denklem (2.1.)'de olduğu gibi kabul edilir.

$$H(X) = -\sum_k p_k * \log_2 p_k \quad (2.1)$$

X değişkeninin kesin rastgele olasılık dağılımı (p_1, p_2, \dots, p_n) için, entropi üst sınırı vardır.

Eğer ve sadece $(p_1 = p_2 = \dots = p_n = \frac{1}{n})$ ise bu durumda entropi üst sınırı, denklem (2.2.)'de belirtildiği gibi gösterilebilir.

$$H(X) = H(p_1, p_2, \dots, p_n) \leq \log_2 n \quad (2.2)$$

X ve Y, iki rastgele deęişken ise, $H(X,Y)$ denklem (2.3)'de olduęu gibi ifade edilir.

$$H(X,Y) \leq H(X) + H(Y) \quad (2.3)$$

X ve Y bağımsız ise, bu durumda denklem (2.4) koşullu entropi olarak tanımlanır.

$$H(X | Y) = \sum_j H(X | Y = b_j) P(Y = b_j) \quad (2.4)$$

Kesin durum olan E' nin bilgisi denklem (2.5)'te olduęu gibi tanımlanabilir.

$$I(E) = -\log_2 P(E) \quad (2.5)$$

Bu durumda Entropi' nin bilginin deęeri anlamına geldięi görülür. Entropi ile birlikte koşullu bilgi;

$$I(U|V) = H(U) - H(U|V) \quad (2.6)$$

olarak tanımlanır. Bu eşitlik, entropi ve bilgi kavramlarının birbiriyle ilişkisi olduğunu gösterir. V parametresinin bilinmesiyle U hakkındaki belirsizlik ortadan kaldırılabilir.

2.1.1.2 Minimum Uzaklık

Blok kodlardaki Minimum uzaklık (d_{\min}) herhangi iki kod kelimesi arasındaki uzaklık olarak tanımlanır. Bu deęer sahip olduęu etki nedeniyle, blok kodların kullanıldığı yapılarda önem arz etmektedir. Blok kodlarda;

- Eğer hataların sayısı $d_{\min} - 1$ den az veya eşit ise kod çözücü bütün hataları bulunur.
- Eğer hataların sayısı $\frac{d_{\min} - 1}{2}$ den az veya eşit ise kod çözücü bütün hataları düzeltilir.

Bilgi setleri, $p_0 = p_1 = 0,5$ Shannon olasılık terimleriyle oluşturulmuş rastgele bitlerin içerięidir. Bir BSC kanalının giriş ve çıkışı $\{0,1\}$ olarak gösterilip ikili simetrik kanal için

hata olasılığı f_n olarak adlandırılırsa, $f_n = 0,5$ dir. Bu durumda, $H_2(f)$ ve $H_2^e(f)$ ikili fonksiyonlarının entropi eşitliği denklem (2.7)' deki gibidir.

$$\begin{aligned} H_2(f) &= f \log_2(1/f) + (1-f) \log_2(1/(1-f)) \\ H_2^e(f) &= f \log_e(1/f) + (1-f) \log_e(1/(1-f)) \end{aligned} \quad (2.7)$$

Denklem (2.8) 'de olduğu gibi doğal logaritmaları yazabiliriz.

$$\log(x) \equiv \log_e(x) \quad (2.8)$$

İkili vektör veya matrisin ağırlığı, içindeki 1'lerin varlığına bağlıdır. Bu ağırlık, iki vektör arasındaki örtüşen ortak "1" lerin sayısıdır. Rastgele bitlerin kaynağının yoğunluğu, "1" olan bitlerin beklenen kesridir. Eğer yoğunluğu 0,5 den az ise kaynak seyrek. Eğer uzunluk artarak yoğunluk azalır "v" vektörü bu durumda çok seyrek.

$$R = 1 - H_2(d/N) \quad (2.9)$$

R oranlı N blok genişliği olan kod, denklem (2.9) da belirtlen koşulda, GİLBERT-VARSHAMOV minimum uzaklık sınırını sağlayacaktır.

BSC kanalın kapasitesi $C(f_n)$, Shannon teoreminde, f_n gürültü yoğunluğu parametresi ile denklem (2.10)'da tanımlandığı gibidir.

$$C(f_n) = 1 - H_2(f_n) \quad (2.10)$$

Hesaplanabilir kestrim oranı ise denklem (2.11)'de olduğu gib hesaplanabilir.

$$R_0(f_n) \equiv 1 - \log_2[1 + 2\sqrt{f_n(1-f_n)}] \quad (2.11)$$

Sıralı kod çözme sonsuzluk yaklaşımı kullanılarak hata olasılığının kaybolmasıyla, konvülsiyonel kodun çözümü, beklenen hesaplanabilir orandır.

$$GV(f_n) = \begin{cases} 1 - H_2(2f_n) & f_n < 1/4 \\ 0 & f_n \geq 1/4 \end{cases} \quad (2.12)$$

Denklem (2.12)'de verilen eşitlik ile Gilbert sınırı tanımlanmıştır.

Herhangi N genişliğinde X stringleri üzerinde Olasılık dağılımı, H_x zayıf entropisine sahip olacaktır. Eğer $P(x|N), \varepsilon > 0$ ise $\eta > 0$ için burada tüm $N > N^*$ gibi N^* değeri olacak ve denklem (2.13)'de olduğu gibi sınırlandırılabilir [1].

$$P\left(\left|\frac{1}{N} \log_2 \frac{1}{P(x|N)} - H_x\right| > \eta\right) < \varepsilon \quad (2.13)$$

2.1.1.3. Bayes Teoremi

Bayes iki olayın aynı anda olma olasılığının, koşullu olasılıkla ifadesidir. Bayes ağırları ya da Bayes İnanç Ağırları bir grafiksel olasılık formudur. X, sınıf üyeliği bilinmeyen veri örneği olsun. Y ise veri örneği X in C sınıfına ait olduğunun öngören bir hipotez olsun. Bu sınıflandırma problemi için, $P(Y/X)$ olasılığı, hipotez Y' nin C sınıfına ait olduğu farz edilen veriyi tutmasıdır.

$P(Y/X)$, Y'nin X üzerindeki koşullandırmasına ait sonraki olasılıktır (*sonrasal olasılık*). Örneğin, verinin örnek dünyamızın renk ve şekilleriyle tanımlanan meyveler olduğunu farz edelim. X' in kırmızı ve yuvarlak olduğunu ve H hipotezinin X' in bir elma olduğunu öngören bir hipotez olduğunu farz edelim. $P(Y/X)$, kırmızı ve yuvarlak olarak görünen X' in bir elma olduğu görüşünü yansıtmaktadır. Karşıtlığında, $P(Y)$, Y' ye ait öncelikli olasılıktır (öncesel olasılık). Bu veri örneğinin görünüşüne aldırılmadan verilen veri örneği, herhangi birinin elma olma olasılığıdır. Sonrasal olasılık, $P(Y/X)$, X' den bağımsız olan öncesel olasılıktan ($P(Y)$) daha fazla bilgi üzerine kuruludur.

Benzer şekilde, $P(X/H)$ Y üzerinde koşul sağlayan sonraki olasılıktır. Bilgileri verilen X' in kırmızı ve yuvarlak olma olasılığı biliniyorsa bu elmadır denilebilir. $P(X)$ ise X' in öncelikli olasılığıdır.

$P(X)$, $P(Y)$, ve $P(X/Y)$ verilen veriden aşağıda görüldüğü gibi Bayes teoremi ile kestirilebilir. Bayes teoremi, $P(Y/X)$, $P(Y)$, $P(X)$ ve $P(X/Y)$ 'ı kullanılarak,

$$P(Y / X) = \frac{P(X / Y)P(Y)}{P(X)} \quad (2.14)$$

ile çözüm bulunmasını sağlayan kullanışlı bir teoremdir[1].

2.1.1.4. Saf Bayes Sınıflandırması

Saf Bayes sınıflandırıcı ya da Bayes sınıflandırma belirli kurallar içinde yürütülür. Her veri örneği “n-boyutlu” özellik vektörleri ile gösterilir, $X = (x_1, x_2, \dots, x_n)$, her veri örneği “n” özelliklerden alınan örnek üzerindeki “n” ölçümler ile A_1, A_2, \dots, A_n şeklinde tarif edilir.

M değerinin bir sınıf olduğunu kabul edelim(C_1, C_2, \dots, C_m). Eldeki bilinmeyen bir veri (örneğin, bir sınıf üyeliğine sahip olmayan X değeri), sınıflandırıcı X' in en yüksek sonraki olasılığına sahip sınıfa ait olduğunu öngörecektir. Sonuç olarak, saf Bayes sınıflandırıcı bilinmeyen örnek X'i denklem (2.15) koşuluyla C_i sınıfına atar.

$$P(C_i | X) > P(C_j | X) \text{ için } 1 \leq j \leq m, j \neq i. \quad (2.15)$$

Böylece $P(C_i | X)$ 'i en yüksek büyüklüğe çıkarırız. C_i sınıfı $P(C_i | X)$ en yüksek büyüklüğe çıkarıldığı için denklem (2.16)'de tanımlandığı gibi, *en büyük sonsal hipotez* diye adlandırılır.

$$P(C_i / X) = \frac{P(X / C_i)P(C_i)}{P(X)} \quad (2.16)$$

$P(X)$ bütün sınıflar için sabit ise, sadece $P(X | C_i)P(C_i)$ 'nin en yüksek büyüklüğe çıkarılmasına ihtiyaç duyulur. Eğer sınıf öncelik olasılığı bilinmiyorsa, o zaman genel olarak sınıflar eşit kabul edilir. $P(C_1) = P(C_2) = \dots = P(C_m)$ dir. Bu nedenle, $P(X | C_i)$ ifadesi en yüksek büyüklüğe çıkarılır. Aksi durumda ise, $P(X | C_i)P(C_i)$ ifadesi en yüksek büyüklüğe çıkarılır. Sınıf önceki olasılığı $P(C_i) = \frac{s_i}{s}$ olabilmektedir, burada s_i C_i sınıfına ait eğitilen örnek sayısı ve s ise toplam eğitilen örnek sayısıdır.

Birçok özellik barındıran veri setleri verilirse $P(X | C_i)$ aşırı derece hesaplama gerektirecektir. $P(X | C_i)$ işleminde hesap yükünü azaltmak için, sınıf koşul bağımsızlığına ait saf varsayım uygulanır. Bu varsayımda, özelliklere ait değerler bir diğerinden şartlı olarak bağımsızdır, örneğe ait verilen sınıf üyeliğinde, özellikler arasında bağımlılık ilişkisi yoktur. Bu durum denklem (2.17) ile tanımlanır.

$$P(X | C_i) = \prod_{k=1}^n P(x_k | C_i) \quad (2.17)$$

Olasılıklar $P(x_1 | C_i), P(x_2 | C_i), \dots, P(x_n | C_i)$ eğitim örneklerinde tahmin edilebilirler. Burada;

- Eğer A_k kategorik ise, o zaman $P(x_k | C_i) = \frac{s_{ik}}{s}$, dir. s_{ik} , A_k için x_k değerine sahip olan C_i safına ait eğitim seti sayısı ve s_i ise C_i 'ye ait olan eğitim seti sayısıdır.

- Eğer A_k devamlı-değer ise, özelliklerin tipik olarak denklem (2.18)'de belirtilen Gauss dağılımına sahip oldukları varsayılır. Merkezi limit teoremi ile birden fazla "IID-Independent Identical Distribution (ısı,direnç v.b.)" ye sahip rastgele değişkenlerin toplamı (yaklaşık 10 uniform dağılım) gauss dağılımına yakınsar.

$$P(x_k | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi\sigma_{C_i}^2}} e^{-\frac{(x_k - \mu_{C_i})^2}{2\sigma_{C_i}^2}}, \quad (2.18)$$

burada $g(x_k, \mu_{C_i}, \sigma_{C_i})$ A_k için Gauss yoğunluk fonksiyonudur. μ_{C_i} ve σ_{C_i} ortalama ve standart sapmadır. Bilinmeyen örnek X ' i sınıflandırmak için, her C_i sınıfı $P(X | C_i)P(C_i)$ ifadesi hesaplanır. Denklem (2.19) koşuluyla, Örnek X 'i C_i sınıfına atar. Diğer taraftan, $P(X | C_i)P(C_i)$ en yüksek değeri alması için C_i sınıfına atanır.

$$P(X | C_i)P(C_i) > P(X | C_j)P(C_j), 1 \leq j \leq m, j \neq i. \quad (2.19)$$

Teoride, Bayesian sınıflandırıcılar diğer bütün sınıflandırıcılarla karşılaştırıldıklarında en düşük hata oranına sahiptirler. Bununla birlikte pratik kullanımı için yapılan varsayımlardaki hatadan dolayı olası bir durum değildir. Karşılaştırma alanında yapılan bu sınıflandırıcı ile

ilgili birçok deneysel çalışmalar bu sınıflandırıcının karar ağaçları ve yapay sinir ağları ile birçok alanda yarışabileceğini bulmuştur. Bayesian sınıflandırıcı Bayes teoremini kullanmayan diğer sınıflandırıcıları teorik olarak doğrulamak için de kullanılmaktadır. Örneğin, gerçek varsayım altında, birçok yapay sinir ağı ve eğri uydurma algoritması çıkış olarak, saf Bayesian sınıflandırıcının yaptığı gibi, *en yüksek büyüklük* hipotezini vermektedir.[1]

2.1.1.5. Bayes İnanç Ağları

Saf Bayes sınıflandırıcı, sınıf koşullu bağımsızlığı hakkında varsayım yapar. Verilen örneklere ait sınıf üyelikleri, özelliklere ait değerler birbirlerine koşullu bağımsızdırlar. Bu varsayım hesaplamaları basitleştirir. Bu durumda, Bayes sınıflandırıcının, diğer bütün sınıflandırıcılarla karşılaştırıldığında en doğru sonucu verdiği kabul edilebilir. Bununla birlikte, pratikte değişkenler arasında bağımlılıklar olabilir. Bayes inanç ağları ortak koşullu olasılık dağılımları belirtir ve değişkenlere ait alt kümeler arasındaki sınıf koşullu bağımsızlıkları tanımlamak için olanak sağlarlar. Ayrıca öğrenme sırasında nedensel ilişkilere ait grafiksel model sağlarlar. Bu ağlar inanç ağları, Bayes ağları ve olasılıksal ağlar olarak bilinirler. Genel olarak, bu ağlar inanç ağları olarak adlandırılacaktır.

İnanç ağındaki bileşenlerden biri olan *yönlendirilmiş çevrimsiz çizelge* de, her düğüm bir rastgele değişkenle ve her yay (arc) bir olasılıksal bağımlılıkla gösterilir. Eğer bir yay düğüm Y' den düğüm Z' ye doğru çizilirse, o zaman Y Z' nin ebeveyni veya şu anki selefidir ve Z ise Y nin alt üyesidir. Her değişken, çizelgedeki alt üyeleri olmayanlara koşullu bağımsızdır. Bu değişkenler, ayrık veya sürekli değerler olabilirler. Sunulan veri içindeki gerçek özelliklere karşılık gelebilirler ya da ilişkilerden gelen “gizli değişkenler” olabilirler (örneğin tıbbi belirti tıbbi veri ile ilgilidir).

Bir inanç ağında tanımlı ikinci bileşen her değişken için bir *koşul olasılık tablosu (CPT)* içerir. Sınıf üyelik özellikleri sunulurken, ağ içindeki bir düğüm “çıkış” düğümü olarak seçilebilir. Ayrıca birden çok çıkış düğümü olabilir. Öğrenim için çıkarım algoritmaları ağ üzerinde kabul edilebilir. Sınıflandırma işlemi, tek sınıf üyelik dönüşümünden ziyade, sınıf üyelik özellikleri için bir olasılık dağılımı döndürürler, yani, her sınıfa ait olasılık tahminidir.

2.1.1.6. Shannon Teoremi

Haberleşme teorisinde, iletilen verinin gürültülü ve güvenilir olmayan kanaldan etkileneceği bir gerçektir. 1948 yılında Claude E. SHANNON, güvenilir olmayan kanallar üzerinden güvenilir veri iletimi limitlerini ve bu limitlere hangi yollarla ulaşılacağını gösteren metotlar

hakkında ıęır aıcı sunum hazırlamıřtır. Haberleřme teorisinin temel teoremi olarak adlandırılan bu yaklařım dięer teoriler ile birlikte, bilgi konseptini genel hale getirmiř ve gvenilir olmayan kanallar zerinden iletilebilecek azami bilginin sınırları zilmıřtir [1].

Bir haberleřme kanalının Shannon limiti ya da Shannon kapasitesi; belirli bir grlt seviyesinde teorik olarak iletilebilecek maksimum veri transfer oranı anlamına gelmektedir. kapasite, bant geniřlięi, grlt ve kodlama oranına baęlıdır.

İletim kanalı, giriř , ıkıř ve her ikisinin geiř elemanları iin geiř olasılıęı olmak zere  blmden oluřur. Anlam olarak, geiř olasılıęı, kanal zerinden geirilen giriřin, ıkıř tarafından alınmıř olma olasılıęıdır.

Shannon, verilen bir iletim kanalında limitlerin ortaya ıktıęını, bu limitlerin kanalın kapasitesini verdięini, gvenilir geiřin kapasiteye tesadfi olarak yakın olan miktar iin mmkn olduęunu, kapasitenin zerindeki miktarlar iin ise mmkn olmadıęını kanıtlamıřtır [1].

Kapasitenin anlamı, en gzel bilgi teorisiiyle anlařılır. Bu teori kapasiteye ulařan geiř sistemlerinin ortaya ıkıřını garanti etmemektedir. Shannon ıęır aan konseptini, iletilen vektrlerin toplamı olarak aıklar. řu kesindir ki, eęer bir girdi elementi bile en azından iki deęiřik yoldan (hatta farklı olasılıklarla birlikte) alınabilirse, sadece tekil elementler yollandıęında kanal zerinden gvenilir iletim olanaksızdır.

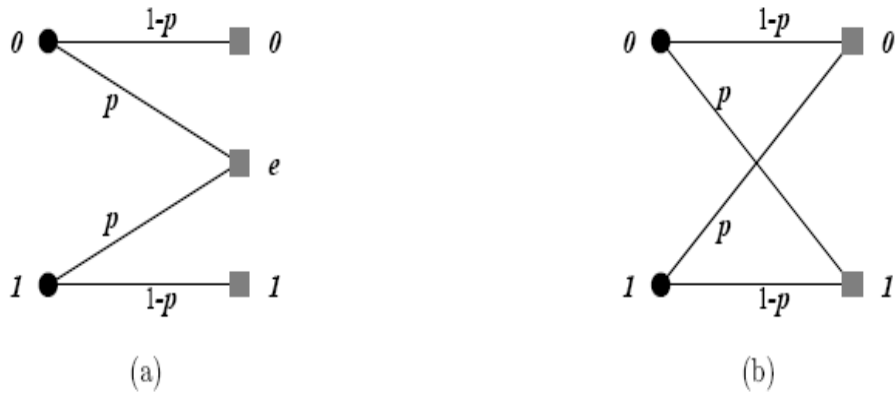
Bu olay, baęlı olmayan oklu elementler yollandıęı zaman bile mmkndr. Gvenilir bir iletim elde etmek iin, baęlı olan giriř elementlerini yollamak nemlidir. Bu da giriř alfabeti zerindeki sonlu vektrler olarak adlandırılan kod konseptini oluřturur. Tm vektrlerin aynı uzunluęa sahip olduęunu ve bu uzunluęa kodun blok uzunluęu adını verdięimizi kabul edelim. Eęer vektrlerin sayısı $K=2^k$ ise, her vektr k parası olarak sayabiliriz. Eęer vektrlerin uzunluęu n ise, k paraları kanallarının n defa kullanımı kadar paralar iletilir. Bu durumda diyebiliriz ki, her kanal kullanımı iin kod, $\frac{k}{n}$ oranına sahiptir.

Kod kelime yollanıp ıkıř alfabeti zerinden bir vektr alındıęını kabul edelim. Bu vektrn, yollanan veri olduęu nasıl anlařılabilir? Eęer kanal yanlıřlıklara izin verirse, hangi kod

kelimesinin doğru şekilde yollandığını söylemek mümkün değildir. Bununla beraber, yollanan kod kelimeye en yakın olan kod kelime; alınan vektörün büyütülmesi sayesinde bunun yollanan kod kelimeyle aynı olma olasılığı sayesinde bulunabilir. Bunu görmek için, birçok kod kelime bulunabilir, basitçe tüm K kod kelimeleri listelenebilir ve teker teker tüm kod kelimelerin olasılığı hesaplanabilir. Daha sonra maksimum olasılığa sahip olan vektörler bulunup bir tanesi geri gönderilebilir. Bu tür kod çözümlere maksimum olasılıklı kod çözümleri denir. Ancak bu kod çözümler mükemmel değildir, kod büyük olduğu zamanlarda çok zaman alır ve yanlış yapılmaması için azami dikkatli olunması gerekmektedir. Kodun blok uzunluğu sonsuza giderken, maksimum olasılıklı kod çözümlerinin sıfıra gitme hatasının olasılığı için olan kapasiteye, rastgele olarak yakın orantıdaki kodlarının varlığını, Shannon kanıtlamıştır (Aslında maksimum olasılık kod çözümlerinin hatasız kod çözme olasılığı, blok uzunluğuyla birlikte ivmeli bir şekilde sıfıra gider).

Kapasiteye ulaşan kodlamalar, iletim yönünden çok iyidir. Fakat Shannon teoremleri bu kodları bulmak için bir ipucu vermemektedir. Daha da önemlisi; alıcı, belirli bir oranda kapasiteye ulaşan kodların sıralarını bile verse, bunları etkili bir şekilde kodlamak mümkün değildir. Kanalın kapasitesine ulaşan etkili kodlama ve kod çözme algoritmaları, gelişen süreçte ortaya çıkmaktadır.

İletişim kanallarına örnek olarak; İkili kalıntı kanalları (BEC) ve çift simetrik kanallar (BSC) Şekil 2-1’ de gösterilmiştir. Her açıdan giriş alfabeti çifttir ve giriş alfabetindeki elemanlara “parça” ya da “bit” denir. İkili kalıntı kanallarında çıkış alfabeti 0,1 ve “e” olarak gösterilen ilave elemanlardan oluşur ve buna kalıntı denir.



Şekil 2.1: İki kanal örneği: a) p silme olasılığıyla ikili silme kanalı (BEC)
b) p hata olasılığıyla ikili simetrik kanal (BSC)

Her parça ya doğru şekilde iletilir (1-p olasılığıyla), ya da silinir (p olasılığıyla). Kanalin kapasitesi ise (1-p)' dir.

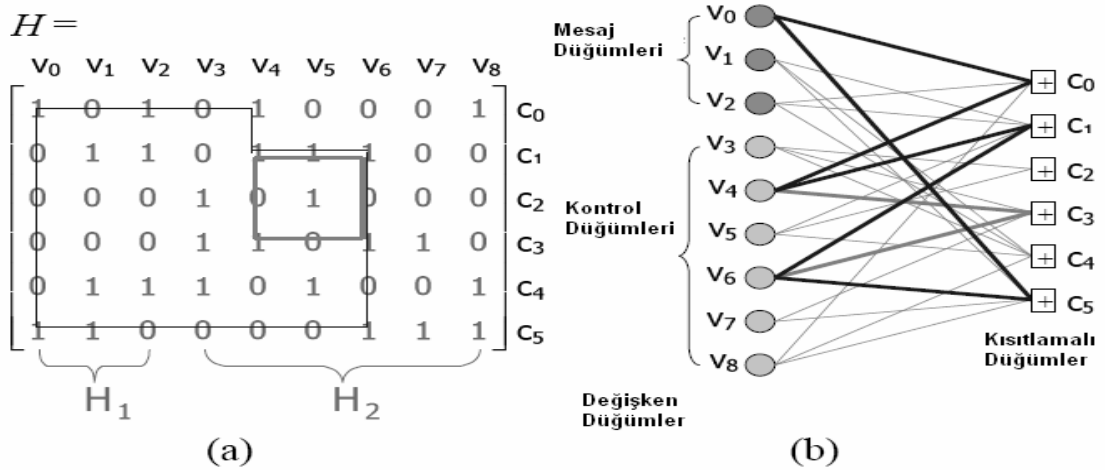
BSC' de hem giriş, hem de çıkış alfabeti F_2 olarak tanımlansın. F_2 her parçaya doğru şekilde iletilir (1-p olasılığıyla) ya da değiştirilir (p olasılığıyla). İlk görünüşte BSC kanalının BEC kanalından daha kolay olduğu düşünülebilir. Ancak pratikte daha karmaşıktır. Bu karmaşa, hangi parçaların değiştiğinin belli olmaması sebebiyle ortaya çıkar (BEC de hangi parçaların silindiği bellidir). Kanalin kapasitesi;

$$1 + p \log_2(p) + (1-p) \log_2(1-p), \quad (C = W \log_2(1 + \frac{S}{N}) = W \log_2(1 + \frac{RE_b}{W_0 N_0})) \quad (2.20)$$

ile ifade edilir. Alınan kelimedeki en küçük Hamming mesafesine sahip olan kod kelimesi F_2 üzerinden verilen vektörün uzunluğu (n), bu kanal için maksimum olasılıklı kod çözücüsünün bulunmasına yardımcı olur [2].

2.1.1.7. Durma, Kod Kelimeleri, Uç Genişleme Kümeleri ve Çevrimler

1/3 oranlı (9,3) bir kodun ikili kümeli grafik açıklamaları ve iyi bilinen matrisi, Şekil 2.2'de verilmiştir. Parite kontrol matrisindeki bir sütun, iki kümeli grafta bir değişkene karşılık gelir. Parite kontrol matrisi H, tersi alınabilen H_2 bölümü hesaba katılarak yapılır, H_2 , H' ın tam derecesi olduğunu gösterir. Sistematik kodlama için H_1 bilgi bitine H_2 ise parite bitlerine karşılık gelir.



Şekil 2.2: Matris ve (9,3) kodun grafiksel gösterimi

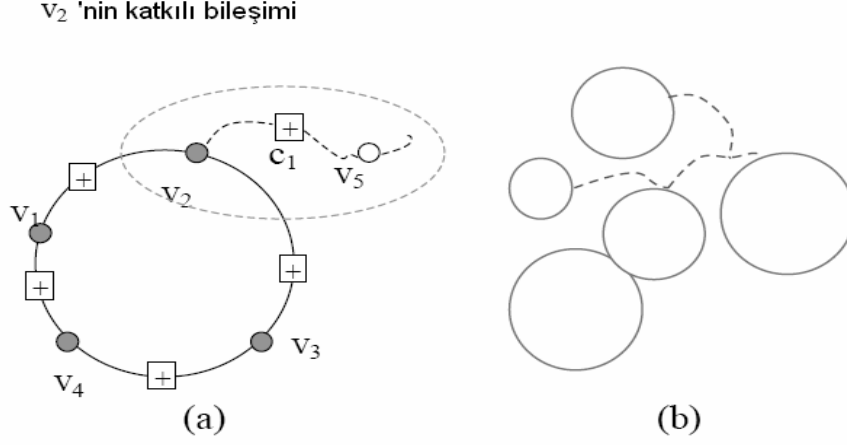
Çevrim, “2d” uzunluğundaki çevrim uçlar tarafından bağlanan “d” değişken düğümleri ve “d” kısıt düğümlerin kümesidir. Bu olay her düğümün kendileriyle aynı uca ikinci kez uğramadan bağlanması sonucu oluşur.

Eğer “ d_1 ” elemanlarına sahip bir veya daha fazla çevrim, bu kümeler ve kendisine bağlı olan kısıtlamalı kümelerin komşuluğu arasında oluşursa, iki kümeli grafta değişken düğümlerin kümesini Çevrim Kümesi (C_d), olarak tanımlarız. Eğer sadece bu değişkenler ve onların kısıtlamalı komşuları arasında hiçbir çevrim içermezse “d” değişkenlerin kümesi C_d ’yi oluşturmaz. C_d için de maksimum çevrim uzunluğu “2d”dir. Şekil 2.3’de altı çevrim uzunluğundaki $(V_0 - c_0 - V_4 - c_1 - V_6 - c_5 - V_0)$ ve dört çevrim uzunluğundaki $(V_4 - c_1 - V_6 - c_3 - V_4)$ örnekleri görülmektedir.

Değişken düğüm kümesi (V_4, V_0, V_6) C_3 kümesidir. V_5 ise 4 çevrim-uzunluğunu içermemesine rağmen değişken düğüm kümesi (V_4, V_5, V_6) ’de aynı zamanda C_3 kümesidir.

Eğer “d” elemanlarına sahip ve bütün komşuları en az iki kez buna bağlandıysa, değişken kümesi durma kümesi (S_d) olarak adlandırılır. Şekil 2.3’de, değişken düğüm kümesi (V_4, V_0, V_6) S_3 kümesidir. Çünkü bütün komşuları (c_0, c_1, c_3 ve c_5) bu kümeye en az iki kez bağlanmıştır.

Durma kümeleri daima çevrimler içerir. Çevrimlerle birlikte graflar üzerinde mesaj geçen kod çözmenin etkinliği, öncelikle durma kümelerini oluşturmak için çevrimlerin nasıl sınıflandırıldığına bağlıdır. Tekil olarak değişken düğümlere bağlanmaksızın iki kümeli grafta her durma kümesi çevrimler içerir. Her durma kümesi (değişken düğümler) ve onun komşuları (kısıtlayıcı düğümler), o düğüme girmeye çalışmaktan ziyade farklı bir uç üzerinden, düğümden ayrılan iki kümeli grafi oluşturur. Bu yolda sonuçlanan iki kümeli grafta, karşıdan karşıya geçme durumunda, kesinlikle herhangi bir düğümden iki defa geçtiğinden çevrim oluşur.



Şekil 2.3: (a) Katkılı mesaj (b) Graf genişlemesi

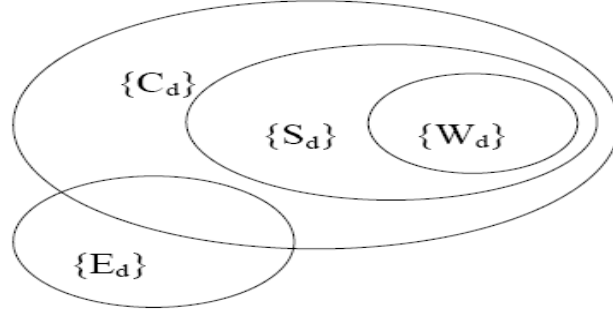
Genel olarak, değişken düğümlerine bağlanmaksızın, iki kümeli grafta durma kümeleri, çoklu çevrimleri kapsar. Durma kümesi bütün 2-dereceli değişken düğümleri içeren tek çevrim tarafından oluşur. Bütün 2- dereceli değişken düğümleri içeren çevrim durma kümesidir. Şekil 2.3 (a), keyfi “2d” uzunluğunda çevrimi gösterir. ($2d=8$). V_2 , bu çevrimin dışında farklı bir kısıtlayıcı düğüme bağlanmalıdır. Bundan dolayı, eğer C_1 düğümü, V_1 , V_3 ya da V_4 ’e bağlanmazsa, durma kümesi en az birden fazla değişken düğümü içermelidir .

Şekil 3.2.(b)’de görüldüğü gibi, iki tip değişken düğüm durma kümesini oluşturur. Kısa çevrimler, en az bir katkılı yol içerdiğinden durma kümesi genişliği artar.

Tamamen “d” elemanlarını kapsayan değişken düğüm kümesi kod kelime kümesi (W_d) olarak adlandırılır. W_3 , bir değişken düğüm kümesi (V_0, V_4, V_6) olup 100010100 kod kelimesine karşılık gelir. d_{\min} minimum uzaklığı ile birlikte doğrusal kod, en az d_{\min} ağırlıksız bir kod kelimesine ve daha küçük ağırlıktayken hiçbiri sıfır-olmayan kod kelimelerine sahiptir. Bundan dolayı en az bir $W_{d_{\min}}$ kümesi vardır. Bütün kod kelime kümeleri durma kümeleridir. Küçük durma kümelerini önlemek aynı zamanda küçük d_{\min} ’ den sakınmaktır. Eğer kod d_{\min} ’e sahipse $s_{d_{\min}}$ durma kümesine sahip olmak zorundadır.

Küçük durma kümeleri, gerekli olan düşük uzaklık olaylarını yansıtmaz. Aslında ML (Maksimum Likelihood) kod çözücü, silinen durma kümesinin kodunu başarıyla çözebilir.

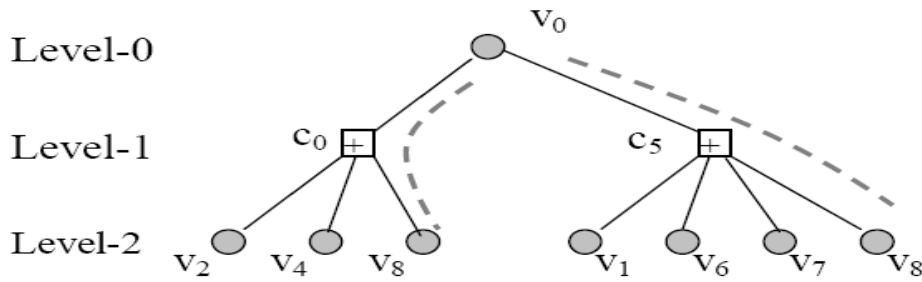
Ancak $\{V_3, V_4, V_5, V_6, V_8\}$ gibi, tam sütun mertebesindeki alt matrisin, durma kümesi değişkenleri tarafından indekslenen parite kontrol matrisi sütunlarından oluşturulmalıdır. Küçük büyüklükteki bütün durma kümeleri ML ile çözülebilir. Ancak, bazıları küçük uzaklıklara neden olur ve tekrarlamalı kod çözme işlemleri ile çözülemeyebilir. İzlenmesi gereken yol, tekrarlanmalı kod çözmeye uygun kodları oluşturmak yerine minimum durma kümesinin büyüklüğünü artırmak ve uzaklığını azaltmaktır. Şekil 2.4, C_d, S_d, W_d ve E_d arasındaki ilişkiyi özetlemektedir.



Şekil 2.4: C_d, S_d, W_d, E_d arasındaki ilişkiyi gösteren Venn Diyagramı

2.1.1.8. Çevrimsiz Kümeler

Küçük durma kümelerinin kaldırılması ile oluşacak etki açıktır. Ancak, çevrimli kümelere basit çevre koşullandırmalar uygulandığında etkisi tartışılabilir. Çünkü her durma kümesi çevrimler içerir. Geleneksel çevre koşullandırma ile ilgili sorun, burada çok çevrim olmasıdır. Şekil 2.5, değişken V_0 düğümünün destek ağacındaki çevrimini göstermektedir. Tek sayılı olan bütün seviyeler, kısıtlamalı düğümü içerir. Çift sayılı olan bütün seviyeler ise değişken düğümleri içerir. Eğer destek ağacındaki iki pozisyon, iki kümeli grafta aynı düğümü gösterirse çevrim oluşur. V_0 Destek ağacında “2d” uzunluğuna ulaşan çevrimlerin bulunması için, destek ağacının “d” seviyesine genişletilmesi gerekmektedir.



Şekil 2.5: Birçok çevrimi ortadan kaldıran geleneksel çevre koşullandırması

Destek ağacındaki düğümlerin sayısı, seviyelerin sayısının genişlemesiyle üstel olarak genişler. Kısa çevrimlerin olması için bütün bu düğümler farklı olmak zorundadır. Bu nedenle, en uzun çevrim boyutu, blok büyüklüğünün ile logaritmik olarak artması ile engellenir. Sonuçta, sonlu uzunluktaki LDPC kodun çevresel koşullandırması, blok uzunluğunun artması ile ciddi olarak sınırlanır.

Yüksek dereceli düğümler bulunduğu zaman, çevre koşullandırma problemlidir. Güncel çevre koşullandırma teknikleri, genellikle yüksek derece düğümlerini içermez. Ancak bu durumda, kapasite yaklaşımı istenen seviyede olamayacaktır.

Bileşen değişkenleri içinde herhangi bir çevrim olmaması durumunda, değişken düğüm kümesi çevrimsiz küme olarak adlandırılır. 2-dereceli değişken düğümler kümesinin çevrimsiz olması için gereken koşul, bu kümenin lineer olarak bağımsız olmasıdır. Doğrusal olarak bağımsız olmayan kümeler kod kelimelerini içerir. Kod kelime kümeleri özel durma kümeleridir ve durma kümeleri de çevrimler içerir. Yeterlilik için, kısıtlanmalı düğümlerinin her biri, iki değişken düğümü tarafından paylaşılan 2-dereceli düğümler arasındaki çevrimlerde yer alır. Bundan dolayı çevrimlerin içinde yer alan sütunların ikili toplamın tümü sıfır vektördür ve bu sütunlar doğrusal olarak bağımlıdır.

$n-k$ uzunluğundaki, $n-k-1$ maksimum 2-dereceli sütunları doğrusal bağımsız olabilir. Örneğin Şekil 2.6'da verilen $[(n-k) \times (n-k-1)]$ diyagonal matrisi düşünelim;

$$\begin{bmatrix} 1 & 0 & \dots & & 0 \\ 1 & 1 & & & \\ 0 & 1 & \ddots & & \\ \vdots & & \ddots & & \\ & & & 1 & 0 \\ & & & 1 & 1 \\ 0 & \dots & & 0 & 1 \end{bmatrix}_{n-k, n-k-1}$$

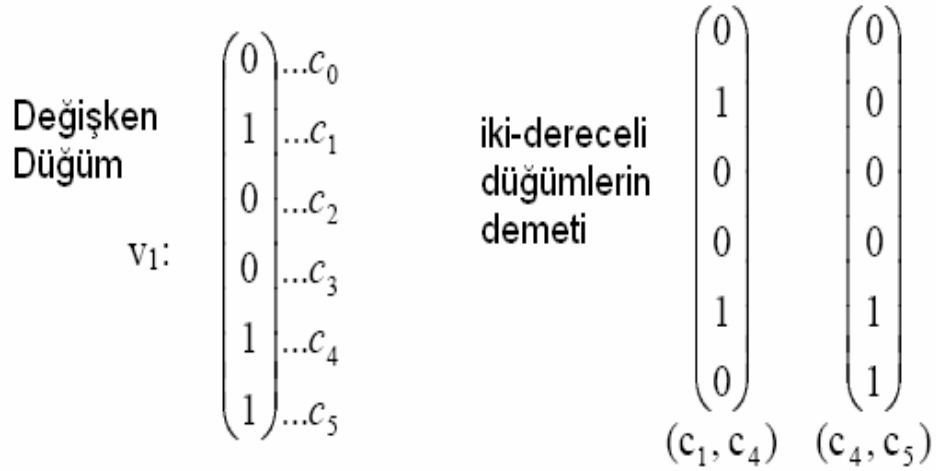
Şekil 2.6: $[(n-k) \times (n-k-1)]$ Diyagonal Matrisi

“n-k” boyutuyla 2-dereceli sütünların her biri “n-k-1” mertebesini oluşturur. “n-k” boyutunda mümkün olan herhangi 2-dereceli sütün, tüm sütünların lineer kombinasyonu ile şekillendirilebilir. 2-dereceli sütünlara ilişkin sınırlama, mümkün olan en iyi d_{\min} değerini “n-k” dan “n-k-1”e düşürebilir.

1-dereceli deęişkenlerden yoksun bir (n,k) kodu içinde, çevrimsiz deęişken düęüm kümesi v_1, v_2, \dots, v_s , denklem (2.21)’de verilen koşulu sağlamalıdır (d_i , v ’nin derecesidir).

$$\sum_{i=1}^s (d_i - 1) \leq n - k - 1 \quad (2.21)$$

Kısıtlamaları $\{c_1, c_2, \dots, c_d\}$ olan “d” dereceli deęişken düęüm kısıtlamaları $\{c_1, c_2\}, \{c_2, c_3\}, \dots, \{c_{d-1}, c_d\}$, 2-dereceli düęümlerin “d-1” kümesi (topluluęu) tarafından deęiştirilir. Örnek olarak Şekil 2.7 ve 2.8’de, deęişken düęüm v_1 ’in, 2-dereceli düęümlerin kümesi tarafından deęiştii görülmektedir.

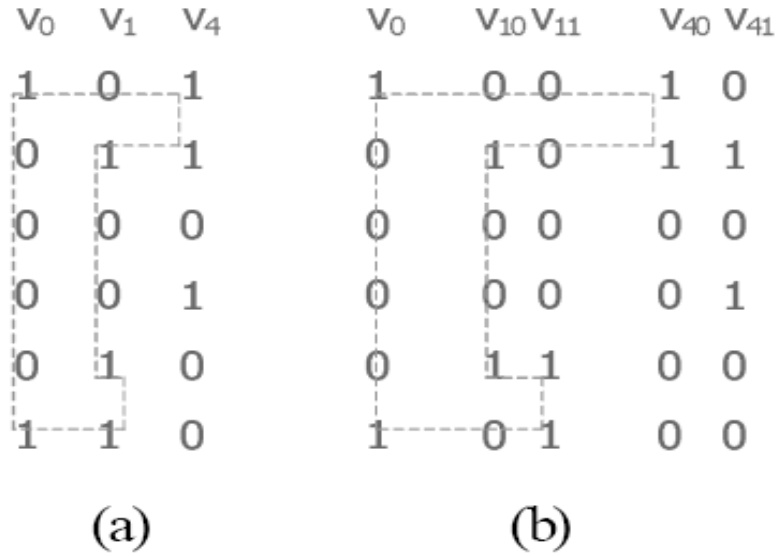


Şekil 2.7: v_1 ’in iki tane 2-dereceli düęümler tarafından yer deęiştirmesi

“1-dereceli” deęişkenlerden yoksun “(n,k)” kodu içinde önemlilięi “n-k-1”den geniş olan deęişken düęümlerin hiçbir kümesi çevrimsiz olamaz.

Yüksek dereceli düęümler iki veya daha fazla 2-dereceli düęümlerin süper pozisyonunu alarak düşünebilir ve çevrim oluşturmaya daha eğilimlidir. Sonuç olarak yüksek dereceli

düğüm üzerinde çevre koşullandırması, düşük dereceli düğümlere göre daha zordur. Ama çevrimler, tamamen daha az problemlili olan yüksek dereceli düğümler içerir. Çünkü daha yüksek EMD (Katkılı Mesaj Derecesi)'ne sahiptir.



Şekil 2.8: Çevrim içindeki demet tarafından v_1 'in yer değiştirilmesi

2.1.1.9. EMD (Extrinsic Message Degree –Katkılı Mesaj Derecesi) Kodlama

Bütün durma kümeleri, en azından değişken düğümlerin minimum sayısına sahip olmalıdır. Ancak herhangi bir polinomsal-zamanlı algoritmanın, küçük durma kümelerini ortadan kaldırdığı tam olarak bilinmemektedir. Kompleks olan durma kümelerinin büyüklüğünün direkt olarak kontrol edilmesi ile mantıklı bir zaman süresinde, çok küçük durma kümelerinin önlenmesi amaçlanmaktadır. Avantaj olarak, graf koşullandıran algoritma, bütün küçük çevrimleri küçük durma kümelerinin parçası olmasını önleyen uç genişleme özelliğine sahiptir. Örneğin, L parametresi, herhangi sol düğümlerin (değişken düğümler) alt kümesi olsun. $E(L)$ 'yi L 'ye bağlanan uçların sayısı olarak tanımlayan $N(L)$, L 'nin komşularının sayısı olsun. " (n,k) " vektörü, kodun (α, γ) uç genişleticisi $E(L) \leq \alpha$ " n " ile birlikte bütün alt kümeler için $N(L) > \gamma E(L)$ 'ye sahiptir.

Eğer uçların $\frac{1}{2}$ sayısı bundan çıkan uçlara bağlayan komşuların sayısından daha az ise " d " değişkenli düğümlerin kümesi uç genişleme kümesi (E_d) olarak adlandırılır. E_d kümesinin komşuları, ortalamanın üzerinde olarak iki defadan daha az kümeyle bağlanmıştır, bundan

dolayı en az bir komşu, bu E_d kümesine tekil olarak bağlanmıştır. S_d setlerinin kümesi ve E_d setlerinin kümesi bağlantısızdır. Verilen “d” değeri için, E_d kümesinin sayısı arttırılırsa, mümkün olan sol kümelerin sayısı azalacaktır. Bundan dolayı iyi uç genişlemesi için koşullandırma yapılması, küçük durma kümesinin sayısını azaltır. Durma kümeleri birbiriyle bağlantılı çevrimlerin kapsamıdır. Küçük durma kümelerini bastırmak için etkili yol, düzensiz LDPC kodunda çevrimlerin uç genişleme özelliklerinin geliştirilmesidir. Değişken düğüm kümesine tekil bağlanan kısıtlamalı düğümler, iyi uç genişlemesi sağladığı bilinmektedir. Çünkü bu kısıtlamalı düğümler kullanışlı mesaj akışı sağlarlar. Algoritma katkılı mesaj derecesi (EMD) olarak adlandırdığımız değişken düğüm kümelerinin parametresi üzerinde yoğunlaşarak başarılabılır.

Değişken düğüm kümesinin katkılı kısıtlama düğümü, bu kümeye tekil bağlanan kısıtlama düğümüdür. Değişken düğüm kümesinin katkılı mesaj derecesi, bu değişken düğüm kümesinin katkılı kısıtlama düğümlerin sayısıdır. Tekrarlamalı olarak çözülen LDPC kodun yüksek SNR bölgelerindeki performansı, kod içinde en küçük durma kümesinin büyüklüğüyle sınırlanır. EMD’ nin ise durma kümesi sıfırdır. Geniş EMD ile değişken düğümler kümesini durma kümesi yapmak için eklenen düğümlerin birbirine bağlanması yeterli olacaktır. .

Bu teknik istatistiksel olarak en küçük durma kümesinin büyüklüğünü arttıracaktır. Aynı zamanda d_{\min} ’i de arttıracaktır. Çünkü kod kelime kümesi durma kümesinin özel durumudur.

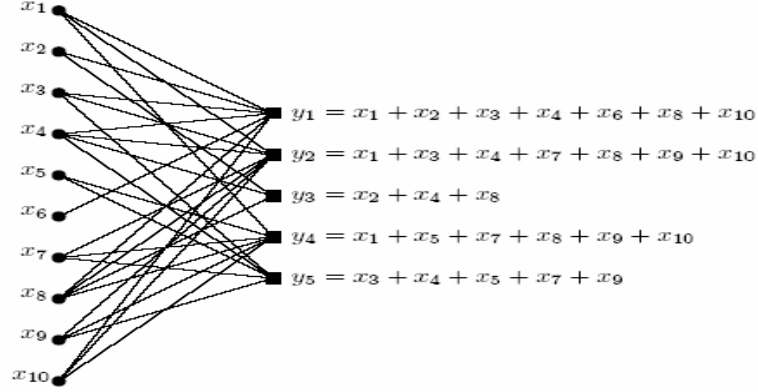
Geniş çevrimler, zorunlu olarak birçok değişken düğüm içerir. Ayrıca istatistiksel olarak küçük çevrimlerin daha iyi olmasını sağlayan bağlantı, graf seviyesi tarafından yüksek EMD’ ye eğilimlidir. Bu nedenle, düşük EMD olması durumunda yüksek SNR’ da hatalar artacaktır.

2.1.2. Kodlama Algoritmaları

“k” boyutunda, “n” blok genişliğinde ikili doğrusal blok kod için kodlama algoritması, “k” orjinal (x_1, x_2, \dots, x_k) bitten kod kelimelerini hesaplayan algoritmadır. Bu algoritmaları işlemler ve maliyetler bakımından birbiriyle karşılaştırmak önemlidir. Algoritmanın maliyeti, F_2 üzerinde yapılan aritmetik işlemler sayısı ile orantılıdır. Doğrusal bir kod için temel g_1, \dots, g_k biliniyorsa, $x_1 g_1 + \dots + x_k g_k$ hesaplanarak kodlama yapılabilir. Hesaplama oluşturmak için yeterli işlemlerin sayısı Hamming ağırlıklarının temel vektörlerine dayanır.

Eğer vektörler uygun ise, kodlama maliyeti “ nk ” derecesindeyken sabit oranlı kodlar için ise “ n^2 ” ile orantılı olacaktır.

LDPC kodlar, matris satırları tarafından oluşan boşluktan daha çok, seyrek matrislerin boş uzayı olarak görülmektedir. Bu eksikliği gidermek için, otomatik olarak hızlı algoritma olarak tanımlanan LDPC kodların modifikasyonu ele alınabilir ya da hızlı kodlama algoritmaları kullanılabilir.

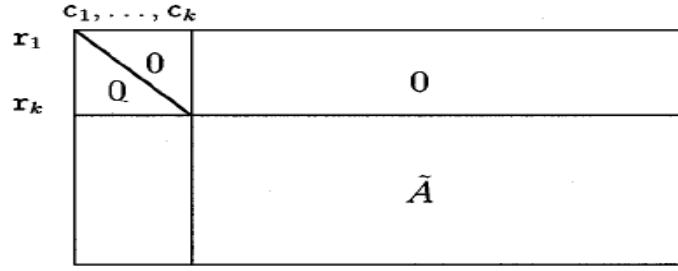


Şekil 2.9: Örnek hızlı algoritma yapısı

Hızlı algoritmayla, seyrek graflardan kodları elde etmenin basit yolu, Şekil2.9’da olduğu gibi her bir kontrol düğümü değerinin, ona bitişik mesaj düğümlerinin toplamı olarak LDPC kodlarının yapısının modifiye edilmesidir. Bu yapıda, ihtiyaç duyulan toplamların sayısı uç sayıları tarafından üst sınır olacaktır. Bundan dolayı eğer graf seyrek ise etkili kodlama mümkündür. Bu kodlamada kod kelimeleri mesaj değerlerini içermekte ayrıca kontrol düğümlerinin değerleri tarafından eklenmektedir. Bu yapı aynı zamanda doğrusal zamanlı kodlayıcıya bir örnektir.

2.1.2.1. Hırslı Algoritmalar

Bu algoritmanın özü diyagonal uzatma basamağıdır. Diyagonal uzatma basamağı olan A matrisi ve bilinen olarak adlandırılan sütunların alt kümeleri verilmiş olsun. Bu bilinen sütunların her bir derecesi bir olan satırlara bağlı olması dikkat edilecek noktadır. c_1, \dots, c_k bilinen sütunları işaret etsin, r_1, \dots, r_k derecesi “1” olan satırlara ve c_i de r_i ’ye bağlı olsun. Yeniden düzenleyerek, A ’nın satır ve sütunları c_1, \dots, c_k , A matrisinin k satırını r_1, \dots, r_k ve A matrisinin k sütununu oluşturalım. (Şekil 2.10)

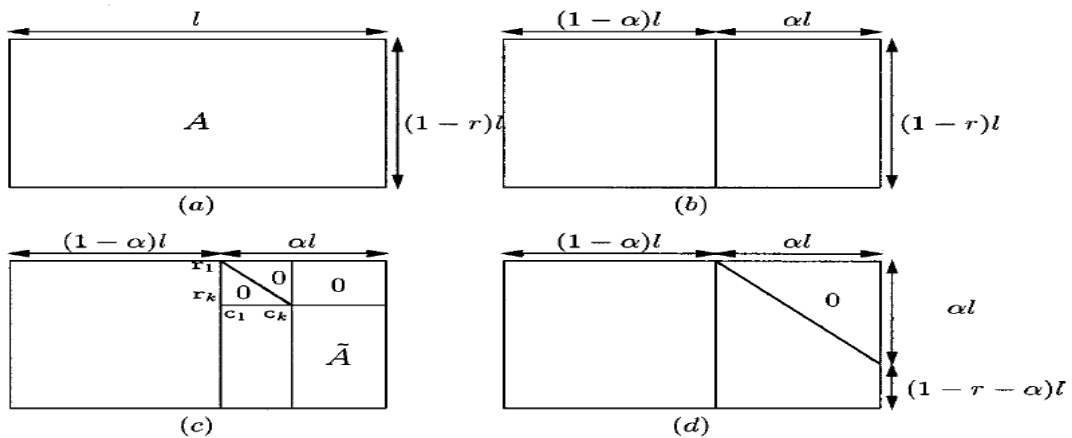


Şekil 2.10: Hırslı Algoritma

\tilde{A} matrisi A 'nın alt matrisidir ve c_1, \dots, c_k ve r_1, \dots, r_k endekslenerek satır ile sütunları silinmiştir. Tekrar düzenlendikten sonra, sol üst köşedeki $k \times k$ diyagonal formda alt matrisi, A 'nın üst k satırları sıfır olmayan girişe sahiptir. Giriş olarak, A matrisi ve bilinen sütunlar verilmiş olsun. Algoritmada bazı satır ve sütun permütasyonları oluşturulur ve kalan A matrisi belirtilir. Diğer bir deyişle, eğer bilinen sütunların hiçbiri, derecesi "1" olan satırlara bağlanmadıysa, sütun permütasyonu oluşturulur. Bundan dolayı bilinen bütün sütunlar A matrisine yön veren sütunu oluşturur. Bu bilinen sütunlar orjinal matristen silinir, kalan matris, \tilde{A} olarak sonuçlandırılır. Daha sonra, orjinal matristen bilinen c_1, \dots, c_k sütunları ve bilinen r_1, \dots, r_k satırlarını sileriz ve matris \tilde{A} olarak kalır.

2.1.2.2. A Hırslı Algoritması

Başlangıç olarak A matrisi verilsin, $1 - \alpha$ olasılığıyla bilinen olarak her sütun bağımsızca bulunabilir ya da diğer bir yolla silme de yapılabilir. $\tilde{A} = A$ olarak kabul edelim. Eğer \tilde{A} bilinen sütun veya 1 dereceli satırdan birini içermezse, varolan matris çıkarılır. Aksi takdirde diyagonal uzama basamağı oluşturulur. Bir dereceli satıra bağlanan \tilde{A} 'nın içinde herhangi bir sütun, bilinen olarak belirtilir ve yeniden diyagonal uzatma basamağı oluşturulur.



Şekil 2.11: A Hırslı Algoritması

“A” hırslı algoritmasını görmek için Şekil 2.11(a)’da gösterilen $(1-r)l \times l$ boyutunda bir A matrisiyle başlanıldığını düşünelim. Başlangıç basamağında, bütün sütunların beklenen $(1-\alpha)$ kesri bilinen olarak sınıflandırılır ve kalanlarda silme olarak belirtilir. İlk başta, birinci basamak oluşturulur. $(1-\alpha)l$ ile bilinen sütunlar Şekil 2.11(b)’de gösterildiği gibi A matrisine yön veren sütunları oluşturmak için yeniden düzenlenir. Artık matrisin bir dereceli satırlara sahip olduğunu varsayarak, bu bir dereceli satırlara bağlanan sütunlar tespit edilir. c_i ’nin r_i ’ye bağlanmış olduğu gibi c_1, \dots, c_k sütunlar, r_1, \dots, r_k ise satırlar olsun. Birinci basamağın ikinci uygulaması süresince bu bilinen sütunlar ve onunla ilişkili satırlar Şekil 2.11(c)’de gösterildiği gibi diyagonal boyunca düzenlenir. Daha sonra, her eklenen tekrarlama daha da genişler. Eğer bu prosedür doğal olarak durmazsa sonuçlanan diyagonal αl beklenen uzaklığına sahip olacaktır. Bundan dolayı satır aralığı $(1-r-\alpha)l$ beklenen genişliğine ve sütun aralığı $(1-\alpha)l$ beklenen genişliğe sahip olacaktır. Sonuç Şekil 2.11(d)’de gösterilmiştir.

Diğer bir durumda, prosedür bütün sütunlar tükenmeden önce biterse, ondan sonra kalan sütunları sola doğru yeniden düzenleyerek yaklaşık üçgenimsi şekle sahip olunur. Sütunların kalan kesrini εl eşit olarak kabul edersek, sonuçlanan satır aralığı $(1-r-\alpha+\varepsilon)l$ ve sonuçlanan sütun aralığı $(1-\alpha+\varepsilon)l$ eşit olur.

(μ, ν) derece çifti ve $\alpha < \alpha^*(\mu, \nu)$ değerlerini küçük seçelim. $C^l(\mu, \nu)$ topluluğundan rastgele graf seçelim, onun genişletilmiş bitişik matrisi \tilde{A} olsun. Hırslı A algoritmasını \tilde{A} genişletilmiş bitişik matrisine uygulayalım. Bundan sonra satır aralığı $(1-r-\alpha)l$ değerine yaklaşmaya ve sütun aralığı $(1-\alpha)l$ değerine yaklaşmaya başlar. Böylece A Hırslı Algoritmanın performansı gözlenebilir. $\alpha \uparrow \alpha^*$ kabul edilirse, A hırslı algoritması ile başarılabilir minimum sütun aralığının $(1-r-\alpha^*)l$ değerine ve minimum sütun aralığının $(1-\alpha^*)l$ değerine eşit olduğu görülür.

2.1.2.3. AH Hırslı Algoritması

AH hırslı algoritması ile, verilen LDPC kodu \hat{H} genişletilmiş parite kontrol matrisine A matrisinin direk uygulanmasını ifade eder. Dikkat edilmesi gereken aralık, sonuçlanan satır aralığıdır.

$r(\lambda, \rho) > 0$ ile birlikte (λ, ρ) derece distribüsyon çifti olsun. $\alpha < \alpha^*(\lambda, \rho)$, u seçelim. $C^n(\lambda, \rho)$ topluluğundan rastgele kod seçelim ve \hat{H} ilişkilendirilen genişletilmiş parite kontrol matrisi olsun. A hırslı algoritmasını \hat{H} , a uygulayalım. Daha sonar, aralık $(1 - r(\lambda, \rho) - \alpha)n$ beklenen değerine yakınsar. $\alpha \uparrow \alpha^*$ Sağlanarak A hırslı algoritması yapılabilir minimum aralık olan $(1 - r(\lambda, \rho) - \alpha^*)n$, a eşit olur.

2.1.2.4. AHT Hırslı Algoritması

A algoritması, direkt olarak LDPC kodun genişletilmiş parite kontrol matrisine uygulanması yerine genişletilmiş parite kontrol matrisinin yerlerinin değiştirilmesi sağlanır. Bu durumda beklenen aralık, sonuçlanan sütun aralığıdır.

$r(\lambda, \rho) \geq 0$ ile birlikte (λ, ρ) derece distribüsyon çifti olsun. $\alpha < \alpha^*(\rho, \lambda)$, u seçelim. $C^n(\lambda, \rho)$ topluluğundan rastgele kod seçelim ve \hat{H} ilişkilendirilen genişletilmiş parite kontrol matrisi olsun.

A hırslı algoritmasını \hat{H}^T , ye uygulayalım. Bu, $C^{\frac{n}{1-r(\rho, \lambda)}}(\rho, \lambda)$ topluluğundan rastgele seçilen genişletilmiş bitişik matrise A algoritmasını uygulamaya denk düşer. Daha sonra aralık

$\frac{1 - \alpha}{1 - r(\rho, \lambda)} n$ beklenen değerine yakınsar. $\alpha \uparrow \alpha^*(\rho, \lambda)$ Sağlanarak AHT hırslı algoritması

yapılabilir minimum aralık olan $\frac{1 - \alpha^*(\rho, \lambda)}{1 - r(\rho, \lambda)} n$ a eşit olur.

(3,6) düzenli kod ve AH hırslı algoritması için aralık örneği ele alındığında; $r(x^5, x^2) = -1$ ve

$\alpha^*(x^5, x^2) = \frac{3^{18}}{2^{17}5^5}$ olduğunu biliyoruz. Bu durumda, AHT hırslı algoritmasının minimum beklenen aralığı denklem (2.22) ile ifade edilir.

$$g = \frac{1 - \frac{3^{18}}{2^{17}5^5}}{2} n \approx 0.0270746 \quad (2.22)$$

2.1.3. LDPC Kodlama Yapısı

Düşük Yoğunluklu Parite Kontrol (DYPK - LDPC) kodları ilk olarak 1962 yılında Robert Gallager tarafından tanımlanmış, ancak bilgisayar teknolojisinin gelişmesiyle doğru orantılı olarak bellek kapasitesinin artmasıyla, yeniden gündeme getirilebilmiştir. Gallager, parite kontrol matrisinin sütun ve satırlarında d_v ve d_r adet 1 biti bulunan n blok uzunluklu (d_v, d_r, n) düşük yoğunluklu parite kontrol kodunu tanımlamıştır. (d_v, d_r, n) kodu düzenli yapısından dolayı, Düzenli Düşük Yoğunluklu Parite Kontrol kodu olarak adlandırılmaktadır. Gallager ayrıca “mesaj geçiş” algoritmasına dayanan yumuşak ve sert kararlı iteratif kod çözücü yapısını önermiştir. Daha uzun kodlarla benzetim yapmak için gerekli alt yapı olmadığından, o dönemlerde hata denetimi için maksimum blok uzunluğu yaklaşık 500 bit civarında olan düzenli kodlar için, sert kararlı kod çözme algoritmasını kullanarak çeşitli benzetimler yapmış ve bu benzetimler sonucunda düşük yoğunluklu parite kontrol kodlarının hata düzeltmede çok iyi bir hata başarımla potansiyeline sahip olduğunu göstermiştir [3][4].

Diğer kod sınıfların aksine çok bellek kullanılması ihtiyacı olmasına rağmen, LDPC kodlama çok hızlı kodlama ve kod çözme algoritmalarına sahiptir. Yüksek miktarda gürültüde bu algoritmaların orijinal kod kelimelerini saklayıp saklamayacağı bir sorundur. Yeni analitik ve birleştirilebilir algoritmalar bu tasarım sorununu halletmemizi sağlamıştır. Bu ise LDPC kodlamayı hem teorik olarak çekici hale getirmiş hem de kullanım için mükemmel olmasını sağlamıştır. Yapılan çalışmalarda, bu kodlama tekniği “Shannon” Limitlerine yaklaşmakta en uygun kodlama tekniği olarak görülmektedir[5].

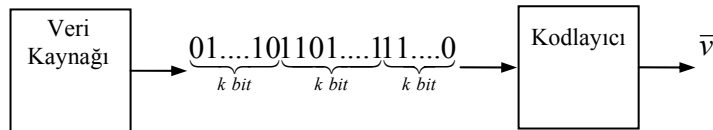
Birçok LDPC kodlayıcı yapıları, sadece bir tek LDPC tipini ya da sabit blok uzunluklu ve kod oranlı LDPC ailesini desteklemektedir. Ancak yapılan çalışmalar ile farklı blok uzunluklarında ve farklı kodlama oranlarındaki LDPC guruplarını destekleyen esnek yapıda kodlayıcılar geliştirilmiştir[6]. Ayrıca paralel olarak ta ölçeklendirilebilen LDPC Encoder/Decoder yapıları da mevcuttur [7].

LDPC kodlamaya olan ilginin canlanmasında bir diğer önemli etken ise Turbo kodların bulunmasıdır. Turbo kodlar birçok noktada LDPC kodlama ile benzeşmektedir. En büyük benzerlikleri mesaj geçiş algoritmasının iteratif kod çözme sürecinde gerçekleşmesidir. LDPC kodlamanın çözülmesi, turbo kodlardan farklı olarak daha az karmaşık ve daha kısa bir süreçtir. Ancak kod sözcük uzunluğuna bağlı olarak kodlama işlemi daha uzun zaman almaktadır. Ayrıca LDPC kodlama yapısındaki paralellikten dolayı turbo kodlara göre yüksek veri hızının sağlanabileceği değerlendirilmektedir.

LDPC kodlar üzerindeki çalışmalar seyrek, düzenli veya düzenliye yakın iki taraflı çizgelere sahip kodlar üzerinde yoğunlaşırken uygun yapılandırılmış düzensiz kodlar kullanılarak hazırlanan LDPC kodlar, sönmülemeli kanallarda, Shannon limitine oldukça yakın bir performans göstermektedirler. Düzensiz kodların tanımları gereği, koda ait parite kontrol matrisinin satır ve sütunlarında farklı sayıda 1 biti bulunmaktadır. Kodun sembolik gösterimi olan iki taraflı çizgede sağ ve sol düğümlerin derece dağılımları farklılık göstermektedir. Derece dağılımının dikkatlice seçilmesiyle kapasite arttırımı sağlanabilmektedir.

Özel yapıları ve kod çözme algoritmalarıyla LDPC kodlama, Şekil 2.12’de görülen yapıda bir çeşit blok kod olarak tanımlanabilir. Koda ilişkin parite kontrol matrisi az sayıda, sıfır olmayan eleman (1 biti) içeriyorsa, bu kod Düşük Yoğunluklu Parite Kontrol Kodu olarak adlandırılır. Herhangi bir satır veya sütunun derecesi parite kontrol matrisinin ilgili satır veya sütunundaki 1 bitlerinin sayısı ile ifade edilir. LDPC kodlama sırasıyla bit ve kontrol düğümlerinin derecelerini gösteren ρ ve λ dağılım çifti yardımıyla analiz edilir.

(N, K) blok kodlayıcı; K adet bilgi bitini N adet kodlanmış bit dizisine çeviren bir kodlayıcıdır ($N > K$). Bu yapıda, blok kodun kodlama oranı bilgi bitlerinin toplam bit sayısına oranı olan K/N ’dir. Bu orandan açıkça görülüyor ki kodlayıcı çıkışında bilgi biti sayısından daha fazla bit bulunmaktadır. Bir başka deyişle kodlanmış bit dizisi, bilgi bitlerini tekrar geri elde etmek için gereğinden fazla bilgiye sahiptir.



Şekil 2.12: Doğrusal blok kodlayıcı yapısı

2.1.3.1 Hata Azaltımı İçin Parite Kontrol Matris Yapısı

İki kümeli graflar kaçınılmaz olarak, çevrimlere sahip, birbirine bağlanmış düğümleri olmayan sonlu uzunluktaki kodları gösterir. Bu durumda ağaç benzeri yapı olmayacaktır. İki kümeli graflarda çevrimler, ortakça uygulanan BP (inanç yayılımı) kod çözme ile uygundur. Eğer çevrimler var ise, bir düğümün komşuları, genel olarak koşullu bağımsız değildir. Bundan dolayı graf ayrımı olmamaktadır. Ancak bu durum, pratik olarak bütün çevrimlerde aynı derecede sorun oluşturmaz. Grafta mevcut olan çok çevrim kodlama zorluğu tekrarlamalı(iterative) kodlama ile kolaylaşabilmektedir.

İlişkili koşulsuz kodlarda hata oranı istenen seviyede olabilmektedir. Düzensiz kodlar hemen hemen düzenli kodlara göre genellikle düşük hata zeminlerinden hoşlanır. Görünürde, komşuluklu kod kelimeleri arasındaki daha fazla Hamming uzaklığı ve daha yüksek minimum uzaklıkları nedeniyle bu sonuca varılmıştır[3].

Mackay düzenli LDPC kodlamaya karşı, düzensiz LDPC kodları için SNR eşiği ve hata zemini BER arasında etkileşim olduğunu ilk olarak ortaya çıkarmıştır. Benzer etkileşim Turbo kodlar için de bulunmuştur. Maksimum olasılık (ML) altında LDPC' nin hata zemini, kodun d_{\min} değerine ve d_{\min} ile ilgili hata olaylarına bağlıdır. Ancak rastgele yapılanmış kodlar için geniş minimum uzaklıkları mevcut ise, söz konusu durumun kontrol edilebilmesi için herhangi bir algoritma bilinmemektedir.

Minimum uzaklığı geliştirmek için uygulanan ortak yaklaşım; kısa kod çevrimlerinin ortadan kaldırılması gibi şartlandırma tekniklerine yönelik dolaylı işlemlerin geliştirilmesi yönündedir (çevre koşullandırması). Bu tür koşullandırmaların kullanışlı olduğu gözlemlenmiştir. Kesin kısa çevrimler, geniş anlama sahipse ML kod çözmesi için sorun olmasa bile, tekrarlamalı kod çözmeyle birlikte birleşme kısıtlı performansa sebep olur. Ama bütün kısa çevrimler eşit değerde sorun yaratmamaktadır. Standart çevrim koşullandırması, belirlenen uzunluktan daha kısa bütün çevrimlerin ortadan kaldırılması negatif etki yaptığından, kullanılması ile kod yapısı ciddi olarak sınırlanabilir. Kısa çevrimlerin kaldırılması, grafin kalan kısmından ayrı tutulan alakalı çevrimleri etkisiz kılan teknik kullanılarak yapılabilir. Bu durum, küçük durma kümelerine katkı sağlar. Durma kümelerindeki bütün değişken düğümler silindiğinde, Mesaj Geçen kod çözme kullanılamayacak ancak ML kullanılabilir.

QPSK ve BPSK modülasyon yöntemlerinde, öklid ve Hamming uzaklığı doğrusal olarak ilişkilidir. Hamming uzaklık, spektrum üzerinde yoğunlaşması nedeniyle bu modülasyon türlerinde kod tasarlamak mantıklı olacaktır. Minimum Hamming uzaklığı (d_{\min}), güvenli olarak düzeltebilecek hata (t) sayısına bağlıdır ($d_{\min} = "2t + 1"$). Minimum Hamming uzaklığı aynı zamanda düzeltebilir silmeler " u " sayısına da bağlıdır ($d_{\min} = "u + 1"$).

Minimum durma kümeleri genişliği, tekrarlamalı kod çözme tarafından düzeltilmeyen silmelerin en küçük sayısına eşittir. Dolayısıyla Hamming uzaklığı ile ilişkilidir. Çünkü durma kümeleri ağırlık dağılımı, öklit uzaklık spektrumuyla çok yakın alakalıdır. İkili silme kanalı (BEC) kadar iyi olan ve AWGN kanalları için de uygun olan durma kümeleri, ağırlık dağılımı üzerinde yoğunlaşan LDPC tasarımında kullanılabilir [4].

2.1.3.2. Parite Kontrol (PC) Kodlama

Parite kontrol kodlama, blok giriş bilgi dizisinin 0 ve 1 lerden oluşan rastgele G üreteç matrisi ile çarpılması sonucu oluşan kod sözcükleridir. $N \times K$ boyutlu G üreteç matrisi;

$$G = \begin{bmatrix} 1 & 1 & \dots & 0 & 1 \\ 0 & 1 & \dots & 0 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 1 \end{bmatrix} \text{ şeklinde tanımlanır.}$$

Üreteç matrisi, $P = K \times (N-K)$ ikili bir matris ve $I_K = K$ boyutlu birim matris olmak üzere oluşturulan parite kontrol kodlarına sistematik parite kontrol kodları denir.

$$G = [P | I_K] = \begin{bmatrix} 1 & 1 & \dots & 0 & 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 1 & 0 & 0 & \dots & 1 \end{bmatrix} \text{ olarak gösterilir.}$$

Giriş bit dizisi \bar{u} , K boyutlu bir satır vektörü olarak düşünülürse; K boyutlu \bar{v} çıkış bit dizisi, giriş bit dizisi ile üreteç matrisinin çarpılması sonucu elde edilir. Her bir \bar{u} giriş dizisine karşı elde edilen \bar{v} çıkışları kod sözcüğü olarak adlandırılır. \bar{v} çıkış dizisinin ilk $(N-K)$ tane biti parite kontrol bitleridir. Bu dizi $\bar{v} = \bar{u}.G$ şeklinde tanımlanır. Geri kalan K tane bit ise \bar{u} giriş bit dizisi ile aynıdır.

Tüm olası kod sözcüklerini içeren kümeye, genel olarak kod denir. Kod sözcükleri G üreteç matrisinin satırlarının farklı kombinasyonlarının modulo-2 aritmetiğine göre toplanması ile oluşturulur. Bir ikili blok kod, ancak ve ancak herhangi iki kod sözcüğünün modulo-2 toplamı da bir blok kod sözcüğü ise doğrusaldır.

Kodlayıcı çıkışından alınan \bar{v} vektörüne kanalda \bar{n} gürültü vektörü eklenir. Bu durumda alıcı tarafından alınan \bar{r} işareti;

$$\bar{r} = (\bar{u}.G) + \bar{n} \quad (\text{modulo-2}) \quad (2.23)$$

Eşitliği ile tanımlanır.

Kod çözücünün görevi gürültünün özelliklerini de hesaba katarak, alınan \bar{r} işaretinden \bar{u} bilgi dizisini geri elde etmektir. Optimum kod çözücü;

$$P(\bar{u} | \bar{r}, G) = \frac{P(\bar{u} | \bar{r}, G)P(\bar{u})}{P(\bar{r}, G)} \quad (2.24)$$

eşitliği ile olasılığı asgariye indirerek \bar{u} bilgi dizisine karar verir. \bar{u} 'nin önsel olasılığının tekdüze olduğu ve \bar{n} 'nin olasılığının da \bar{u} 'dan bağımsız olduğu varsayılırsa $(N-K) \times N$ boyutlu parite kontrol matrisi;

$$H = [P^T | I_{N-K}] \quad (2.25)$$

gibi bir sistematik yapıda tanımlanabilir.

Yukarıda tanımlanan parite kontrol matrisi ile üreteç matrisinin evriğinin çarpımı;

$$H.G^T = 0 \quad (\text{modulo-2}) \quad (2.26)$$

eşitliğinde görüleceği gibi tüm sıfır matrisini verir.

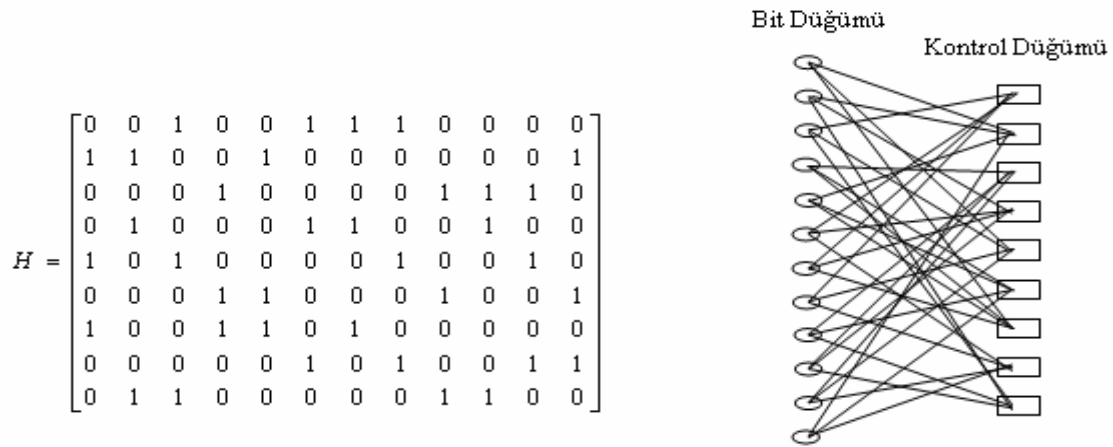
2.1.3.3. Düşük Yoğunluklu Parite Kontrol Kodları

Gallager tarafından varsayılan LDPC kodlamaya ilişkin H matrisi oldukça seyrek (satır ve sütunlarındaki 1 biti yoğunluğu düşük) olup şu özellikleri sağlamaktadır:

- Her bir sütunda ortalama j adet 1 biti bulunur. Başarımı iyileştirmek için j sayısı 3 ten büyük veya 3 e eşit olmalıdır.
- Her bir satır ortalama $k (> j)$ adet 1 biti içermektedir.

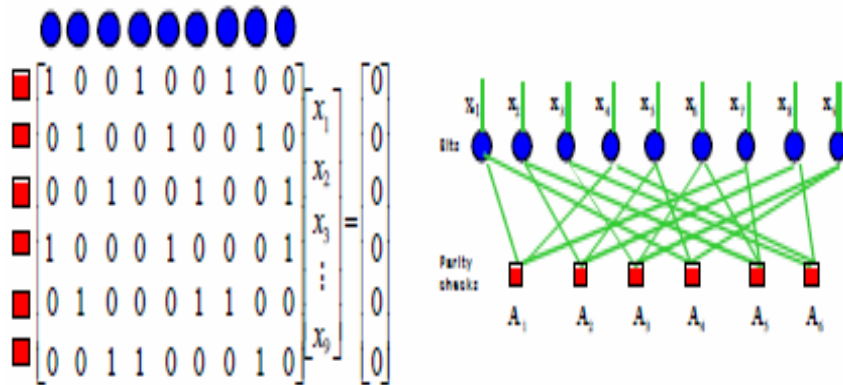
Bu sınırlamaların dışında H rastgele üretilir. Herhangi iki sütun sadece bir satır elemanı dışında örtüşmüyorsa LDPC kodunun başarımı iyileşir. Doğrusal bir blok kod, ikili bilgi bitleri ve bu bilgi bitlerinin modulo-2 toplamlarından oluşan kontrol bitleri kümesinden oluşmaktadır. Bu nedenle doğrusal bir blok kod K uzunluklu bir bilgi vektöründen ve $N-K$ kontrol bitinden N uzunluklu kod sözcükleri üretir.

LDPC kod topluluğu, H matrisinin satır ve sütunlarının sıfır olmayan elemanları ile tanımlanır. Düzenli bir parite kontrol matrisi (H) ve “Tanner graph”ı Şekil 2.13’deki gibidir.



Şekil 2.13: Düzenli (3,4) LDPC, “Tanner Graph” gösterimi

H matrisi ve iki taraflı çizgesi Şekil 2.13’de gösterilen kod düzenli (3,4) LDPC koduna aittir. Burada parite kontrol matrisinin satır ve sütun ağırlıkları sırasıyla 4 ve 3’ tür. Parite matrisi ile “Tanner Graph” ilişkisi Şekil 2.14 ile daha iyi anlaşılacaktır.



Şekil2.14: H matrisi ve “Tanner Graph” gösterimi

“Tanner graf” (iki taraflı çizge), iki değişik tip düğüm içermektedir. Çizgenin sol tarafı (bit düğümleri) kodlanmış bitleri gösterir. Bu düğümler, parite kontrol matrisinin sütunlarını temsil etmektedir. Çizgenin sağ tarafı (kontrol düğümleri) ise H tarafından belirlenen kontrol bitlerini gösterir. Sağ taraftaki düğümler de parite kontrol matrisinin satırlarını temsil etmektedir. Her bir bit düğümü, bir kod sözcüğü içindeki kod sembollerini, ve her bir kontrol düğümü de o koda ilişkin parite kontrol denklemlerini göstermektedir. Herhangi bir kod sembolünün bir parite kontrol denkleminde yer alması, o bit düğümünden kontrol düğümüne “Tanner Grapph” üzerinde bir hat olması ile gösterilir. H matrisinin N sütun ve L satıra sahip olması durumunda çizgede N tane bit düğümü ve L tane kontrol düğümü bulunur. Bu durumda kod oranı $R = 1 - (L/N)$ olarak ifade edilebilir.

H matrisinin sıfır olmayan elemanlarının sayısı (λ, ρ) derece dağılımı denklem (2.27) ve denklem (2.28)’te belirtilen üreteç fonksiyonları yardımıyla tanımlanır.

$$\lambda(x) = \sum_{i=2}^{d_{l_{maks}}} \lambda_i x^{i-1} \quad (2.27)$$

$$\rho(x) = \sum_{i=2}^{d_{r_{maks}}} \rho_i x^{i-1} \quad (2.28)$$

Burada λ_i ; i dereceli bit düğümlerinin oranı, ρ_i ise i dereceli kontrol düğümlerinin oranıdır.

$d_{l_{maks}}$ iki taraflı çizge üzerinde bit düğümlerine ait düğümlerin en yüksek derecesini, $d_{r_{maks}}$ ise kontrol düğümlerinde ait düğümlerin en yüksek derecesini göstermektedir. Bir düğümün derecesi ona bağlı hatların sayısı ile ifade edilir. Kodun derecesi;

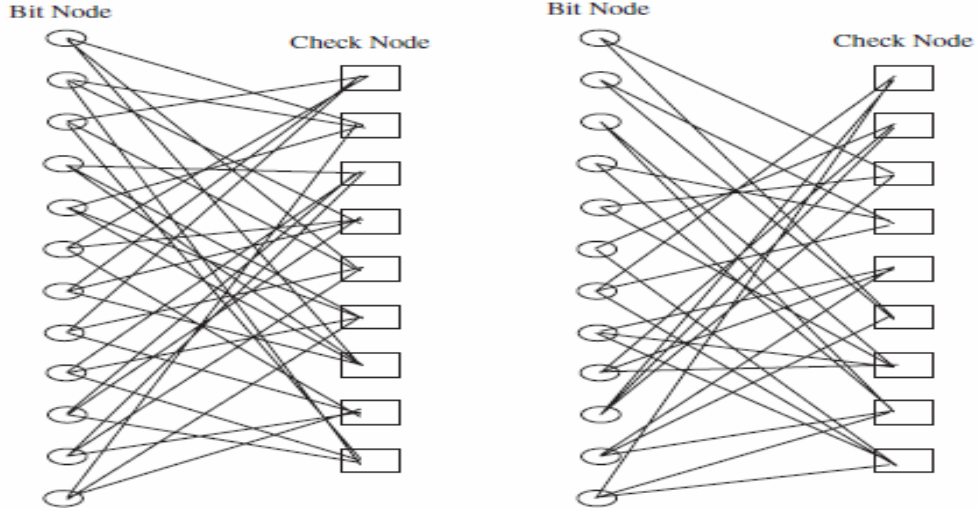
$$1 - \frac{\int \rho(x)}{\int \lambda(x)} \quad (2.29)$$

üreteç fonksiyonları cinsinden de ifade edilebilir. Kullanılacak olan kanal tipine uygun olarak optimize edilen derece dağılımı göz önüne alınarak kodlayıcı ve kod çözücü tasarlanır.

LDPC kodları, parite kontrol matrisinin satır ve sütunlarındaki 1 değerli bitlerin sayısına göre veya derece dağılımının karakteristiğine göre ikiye ayrılır (düzenli LDPC kodları ve düzensiz

LDPC kodları). Düzenli LDPC kodunun parite kontrol matrisinin satırlarındaki ve sütunlarındaki 1 değerindeki bitlerin sayısı sabit, düzensiz LDPC kodunda ise değişkendir. $A(j,k)$ düzenli LDPC koduna ait ikili grafta her bir bit düğümünden j tane çıkış, her bir kontrol düğümüne ise k tane giriş vardır. Bir başka deyişle $A(j,k)$ kodunun parite kontrol matrisinin tüm satırları j ve tüm sütunları k ağırlığına sahiptir.

Düzensiz LDPC kodları kullanarak düzenli kodlara oranla daha iyi bir Bit Hata Oranı (BER) başarımı elde edilebileceği fikri ortaya atıldıktan sonra tüm çalışmalar düzensiz kodlar üzerinde yoğunlaşmıştır. Düzensiz bir LDPC kodlamaya ilişkin yapının daha iyi anlaşılabilmesi için hem düzenli hem de düzensiz yapılara ait “Tanner Graph” gösterimleri Şekil 2.15’te verilmiştir.



Şekil 2.15: Düzenli-Düzensiz LDPC, “Tanner Graph” karşılaştırma

Düzenli ve Düzensiz LDPC Kodlara ilişkin karşılaştırma yapıldığında;

- Düzenli LDPC kodun parite kontrol matrisinin satırlarındaki ve sütunlarındaki “1”lerin sayısı sabit olduğu,
- Düzensiz LDPC kodun parite kontrol matrisinde satır ağırlıkları ve sütun ağırlıklarınının değişken olduğu,
- Düzensiz LDPC kodlaman satır ve sütunlarındaki optimal dağılımın, yoğunluk evrimi (density evolution) ile bulunabileceği,
- Düzensiz kodların performansının, düzenli kodlara göre daha iyi olacağı söylenebilir.

2.1.3.4. Düşük Yoğunluklu Parite Kontrol Kodlama

LDPC kodları, çoğunlukla rastgele yapıda oluşturulmuş parite kontrol matrislerine sahiptirler. Rastgele yapıdaki bu H matrisleri Gauss eleme yöntemi ile sistematik hale dönüştürülür. Çok uzun kod sözcükleri ile çalışıldığından, Gauss eleme yöntemi hesaplama yoğunluğu ve karmaşıklığı bakımından zorlayıcı bir yöntemdir. Kodlama işlemi sistematik yapıdaki parite kontrol matrisine göre yapılırken, kod sözcüklerinin çözülmesi sürecinde seyrek yapıdaki parite kontrol matrisi kullanılır. H matrisi sistematik biçime sokulduğunda, matrisin parite kısmının genellikle çok seyrek olduğu kolaylıkla görülebilir. Gauss eleme yöntemi uygulandıktan sonra oluşan H matrisi, I (N-K)x(N-K) boyutlu birim matris ve P (N-K)xN boyutlu matris olmak üzere;

$$H = [I_{N-K} | P] \quad (2.30)$$

şeklinde tanımlanabilir. Bu durumda LDPC kodunun üreteç matrisi de denklem (2.31) ile ifade edilebilir.

$$G = [P^T | I_K] \quad (2.31)$$

LDPC kodları (N, K) doğrusal blok kodunda olduğu gibi, bilgi bitleri dizisi ile G üreteç matrisinin çarpılmasıyla kodlanmış bit dizisine dönüştürülmüş olur. K uzunluklu \bar{u} bilgi vektörü, KxN boyutlu G üreteç matrisi ile çarpılır ve sonuçta N uzunluklu \bar{v} kod sözcükleri elde edilir.

Parite kontrol matrisi H'ın sistematik bir yapıda olmadığı durumu inceleyelim. A_1 ve A_2 sırasıyla (N-K)x(N-K) ve Kx(N-K) boyutlu iki matris olmak üzere;

$$H^T = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \quad (2.32)$$

olarak ifade edilir. \bar{u} K adet bilgi biti dizisini, p de N-K adet parite kontrol bitini ifade etmekteyken kodlanmış bit dizisi \bar{v} ; $\bar{v} = (p | \bar{u})$ şeklinde gösterilebilir. Kodlanmış bit dizisi ile o koda ait parite kontrol matrisinin evriğinin çarpımı;

$$\bar{v}.H^T = (p | \bar{u}). \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = 0 \quad (2.33)$$

Bize tüm sıfır dizisini verir. Bu eşitlik düzenlenecek olursa,

$$\bar{p}.A_1 + \bar{u}.A_2 = 0 \quad (2.34)$$

denklemini elde edilir. H parite kontrol matrisi ile tanımlanmış olan LDPC kodunun üreteç matrisi G 'nin,

$$G = \left[\bar{p} | I_k \right] \quad (2.35)$$

olduğunu varsayalım. Burada \bar{p} $K \times (N-K)$ boyutlu bir matristir. Bu durumda ;

$$\bar{v} = \bar{u}.G = \bar{u} \left[\bar{p} | I_k \right] \quad (2.36)$$

olduğundan,

$$\bar{p} = \bar{u}.\bar{p} \quad (2.37)$$

eşitliği elde edilebilir.

$$\bar{p}.A_1 + \bar{u}.A_2 = 0 \quad (2.38)$$

denkleminde \bar{p} 'nin değeri yerine konulursa ;

$$\bar{u}(\bar{p}A_1 + A_2) = 0 \quad (2.39)$$

ifadesi elde edilir. Üreteç matrisi G 'nin bulunmasında kullanılan parametre, denklem (2.40)'da gösterilmiştir.

$$\bar{p} = A_2 A_1^{-1} \quad (2.40)$$

Bulunan bu tanımlama G üreteç matrisinde yerine konulacak olursa denklem (2.41) elde edilir.

$$G = [\bar{p} | I_k] = [A_2 A_1^{-1} | I_K] \quad (2.41)$$

Parite kontrol matrisi bilinen bir kodun üreteçini, bu şekilde elde etmek mümkündür. Üreteç matrisi ile giriş bilgi bit dizisi çarpılarak yine kodlama işlemi gerçekleştirilmiş olur.

LDPC kodları doğrusal kodlar olduğundan, yani rastgele iki kod sözcüğünün toplamı yine başka bir kod sözcüğü olduğundan dolayı, bu kodların başarımı herhangi bir kod sözcüğü üzerinden incelenebilir.

LDPC kodlarının kodlanma sürecinde $(N-K)$ adet parite kontrol biti elde etmek için $(N-K) \times K$ kez ikili çarpma işlemi ve $(N-K) \times (K-1)$ kez ikili toplama işlemi yapılmaktadır. N uzunluklu kod sözcükleri elde etmek için de $(N-K)$ tane parite kontrol bitine K uzunluklu bilgi bit dizisi eklenmektedir. Bu sebeple LDPC kodlarının kodlama karmaşıklığı N^2 ile doğru orantılıdır. Dikkat edilmesi gereken bir diğer önemli nokta ise bilgi bitlerinden kodlanmış kod sözcüklerini elde etmek için üreteç matrisinin parite kontrol bitlerinin saklanma zorunluluğudur. Parite kontrol bitlerini saklamak için $K \times (N-K)$ bitlik bir belleğe ihtiyaç duyulmaktadır. Bu da büyük N değerleri için oldukça büyük bir bellek gereksinimi yaratan bir dezavantaj olarak görülebilir.

2.1.3.5. Düşük Yoğunluklu Parite Kontrol Kodlarının Çözülmesi

Blok kodlarda kod çözme, demodülatör çıkışında \hat{x} dizisine karar verilmesi, ardından bu bitler yardımıyla $p = \hat{x}H$ parite kontrol bitlerinin bulunması ve son olarak da başvuru çizelgesi yardımıyla hatalı bitlerin konumlarını belirlenmesi işlemlerini içerir. Kod sözcüklerinin uzunluğu arttıkça, kod çözünün karmaşıklığı ve bellek gereksinimi üstel olarak artmaktadır. LDPC kodları genellikle çok uzun kod sözcüklerine sahip olduklarından aşağıda anlatılacak olan kod çözme algoritması oldukça iyi bir hesaplama duyarlılığına sahip olmakla birlikte çok geniş bir belleğe ihtiyaç duymaktadır.

Mesaj geçiş algoritması, kodlanmış bitlerin yumuşak bilgisinin kullanılmasını olanağını sunar. Mesaj geçiş algoritması, işlenmiş bilgilerin bit düğümleri ve kontrol düğümleri arasında değiş tokuş edilmesi prensibine dayanır. Her bir düğüm kendisine bağlı hatlardan gelen dışsal

bilgileri alır, bilgilerin çeşidine göre bunları işler ve yine aynı hatlar üzerinden işlenen bilgileri diğer düğümlere geri yollar.

Bit düğümlerinin derecesi d_b ve kontrol düğümlerinin derecesi de d_k olan (d_b, d_k) LDPC kodunu ele alalım. Bit düğümleri $d_k - 1$ adet kontrol düğümünden, kontrol düğümleri de $d_b - 1$ adet bit düğümünden gelen dışsal bilgileri alır. Bu bilgiler düğümlerde işlendikten sonra aynı hatlar yardımıyla geri yollanır. Düğüme bağlı herhangi bir hat üzerinden yollanan bilgi karşı taraftaki ilgili düğüme yeniden işlenir ve tekrar diğer düğüme geri yollanır. Bu süreç bir çok kez tekrarlanır. Düğümler arasında iletilen bilgiler logaritmik olasılık oranlarıdır (Log-Likelihood Ratio, LLR). Örneğin;

$$\log \frac{P(y|x=1)}{P(y|x=0)} \quad (2.42)$$

şeklinde ifade edilen LLR'da x düğüme ilişkin bitin değerini y de o ana kadar olan iterasyonlar sonucu elde edilen bilgidir.

$$\theta(x) = \log \left(\tanh \left(\left| \frac{x}{2} \right| \right) \right) \quad (2.43)$$

$$L(x_i) = \log \frac{P(x_i=0|r)}{P(x_i=1|r)} \quad (2.44)$$

Denklem (2.29) ve (2.30) kod çözme algoritmasında kullanılan iki fonksiyondur. Bu fonksiyonlarda görülen r parametresi ise uyarlanabilir filtre çıkışıdır.

(λ, ρ) derece dağılımına ve H parite kontrol matrisine sahip bir LDPC kodunu ele alarak mesaj geçiş algoritmasını inceleyelim.

$$\begin{aligned} R_j &= \{i \mid H_{j,i} = 1\} \\ C_i &= \{j \mid H_{j,i} = 1\} \end{aligned} \quad (2.45)$$

R_j , H matrisinin j . satırındaki 1 değerlikli bitlerin sütun numaralarının kümesini göstermekteyken C_i de H matrisinin i . sütunundaki 1 değerlikli bitlerin satır numaralarının kümesini göstermektedir. R_j kümesinin elemanları ile β_j 'ler $R_{j,0}, R_{j,1}, \dots, R_{j,\beta_j-1}$ olarak ifade edilsin. Aynı şekilde C_i kümesinin elemanları olan v_i 'ler de $C_{i,0}, C_{i,1}, \dots, C_{i,v_i-1}$ şeklinde gösterilsin.

İki taraflı çizge gösterimi düşünülecek olursa $R_{j,0}, R_{j,1}, \dots, R_{j,\beta_j-1}$ terimleri j . kontrol düğümüne bağlı bit düğümleridir (sol düğümler). Görülüyor ki j . kontrol düğümünden çıkan β_j tane hat vardır. Benzer şekilde $C_{i,0}, C_{i,1}, \dots, C_{i,v_i-1}$ terimleri de i . bit düğümüne bağlı kontrol düğümlerine karşılık gelmektedir. Buradan da i . bit düğümünden çıkan kontrol bitlerinin sayısının v_i olduğu kolayca anlaşılmaktadır.

Gerekli tanımlamaları yaptıktan sonra mesaj geçiş algoritması aşağıda belirtildiği gibi özetlenebilir.

Birinci adımda kanaldan alınan bitlerin LLR (Log Likelihood Ratios) 'leri hesaplanır. Hesaplanan bu olasılıklar LDPC kodunun olasılıkları olarak kullanılır. İkinci adımda da sonuçta karar verilecek bit dizinini bulmak için gerekli LLR 'lerin, bit ve kontrol düğümleri arasında güncellenmesi ile değiş tokuşu yapılır.

$$1. i = 0, 1, \dots, N-1, \forall j \in C_i, \quad L_j^{(0)}(x_i) = L_{ch}^i; L_{ch}^i = \frac{-2}{\sigma^2} y_i$$

$$2. q = 1, 2, \dots, Q$$

$$\bullet \quad j = 0, 1, \dots, L-1$$

$$M1 = \sum_{k=1}^{\beta_j} \theta(L_j^{(q-1)}(x_{R_{j,k}}))$$

$$S1 = \prod_{k=1}^{\beta_j} \text{sign}(L_j^{(q-1)}(x_{R_{j,k}}))$$

$$i = 0, 1, \dots, \beta_j - 1$$

$$M2 = M1 - \theta(L_j^{(q-1)}(x_{R_{j,i}}))$$

$$S2 = S1 \times \text{sign}(L_j^{(q-1)}(x_{R_{j,i}}))$$

$$L_{e_j}^{(q)}(x_{R_{j,i}}) = -S2 \times \theta(M2)$$

- $i = 0, 1, \dots, N-1, \forall j \in C_i$ ise $L_j^{(q)}(x_i) = L_{ch}^i + \sum_{l \in C_i, l \neq j} L_{e_l}^{(q)}(x_i)$

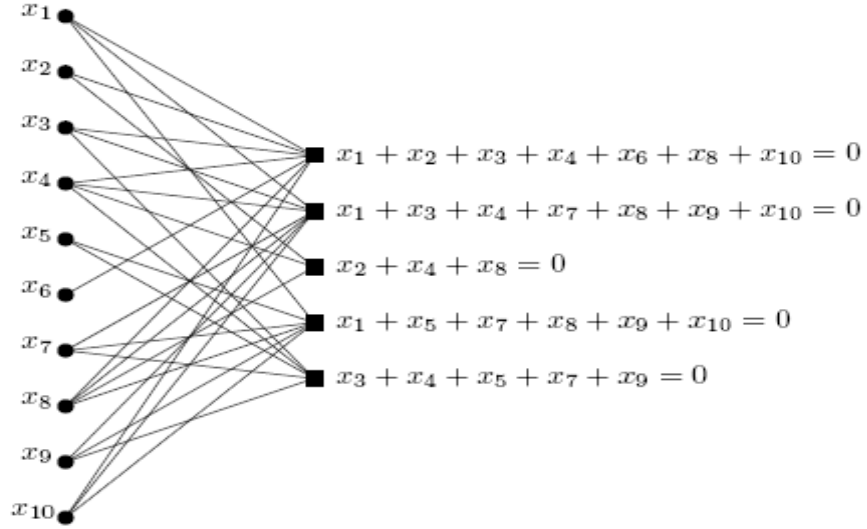
- Yumuşak çıkış: $L(x_i) = L_{ch}^i + \sum_{j \in C_i} L_{e_j}^{(q)}(x_i)$

- Sert karar verme: $\hat{x}_i = \begin{cases} 0 & \text{diğer} \\ 1 & L(x_i) < 0 \end{cases}$

- Durma kriteri: $\hat{x} = [\hat{x}_0, \hat{x}_1, \dots, \hat{x}_{N-1}]$ geçerli bir kod sözcüğü oluşturmuşsa yani $\hat{x}H^T = 0$ 'sa ya da iterasyon sayısı q maksimum iterasyon sayısı olan Q 'ya ulaşmışsa algoritma son bulur.

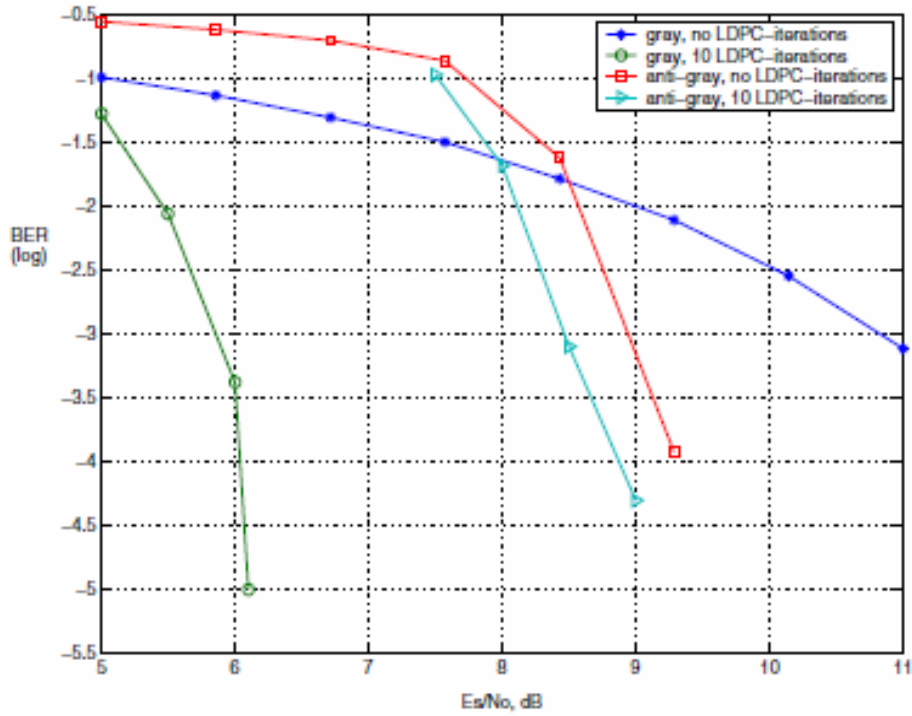
LDPC kodlarının başarımı, çevrimlerin az olduğu çizge varsayımı altında mesaj geçiş algoritmasının yakınsamasını öngören yoğunluk değerlendirme algoritması yardımıyla incelenebilir. Bu varsayım düğümlere gelen olasılıkların, birbirinden bağımsız olmasını sağlamak için oldukça gerekli bir varsayımdır. Çizgenin az sayıda çevrim içermesi teoride sonsuz uzunlukta kod sözcüğü olması durumuna denk gelmektedir. Ancak pratikte sonlu uzunlukta bir kod sözcüğü için kodun başarımı asgari düzeyde çevrim içeren çizgeler tasarlanması ile iyileştirilebilir. Çizgeler çoğunlukla rastgele tasarlanırken dikkat edilecek en önemli nokta aynı düğüm kümesine iki tane hattın bağlanmamış olmasıdır.

Kod kelimeleri, bütün kontrol düğümleri için (c_1, \dots, c_n) gibi vektörlerdir. Ayrıca mesaj düğümleri arasında komşu olanların toplamı sıfırdır. Bu durum grafiksel ve matematiksel bir örnek olarak Şekil.2-16'da gösterilmiştir.



Şekil 2-16: Bir LDPC kodun matematiksel ifadesi

Bilimsel olarak yapılan çalışmalarda, LDPC kodlama performansının analizine yönelik bazı sonuçlar elde edilerek yayınlanmıştır. Şekil 2.17’de verilen sonuçlar değerlendirildiğinde, LDPC kodlamanın gün geçtikçe veri transferinde tercih edilecek bir yapı olacağını göstermektedir. İdeal Bit Hata Oranlarına altı desibel seviyelerinde yaklaşılmaktadır. İterasyon sayısının artırılması ile daha iyi sonuçlar elde edilmektedir[8].



Şekil 2.17: Statik bir kanalda LDPC kodlama performansı

2.1.3.6. Kapasite Yaklaşımli Düzensiz LDPC Kodların Tasarımı

LDPC kodları, turbo kodlar üstünde belirgin avantajlara sahiptir. LDPC için (inanç yayılımı) kod çözme, tamamen paralelleşebilir ve potansiyel olarak önemli yüksek hızlarda başarılabilir. Çok düşük karmaşıklıkta, performans bakımından inanç yayılımına (BP) yaklaşık yakınlıktaki dekoderler, bu kodlar için tasarlanabilir. Ayrıca LDPC kod çözümünde düzeltilmiş kod kelimesi, algılanabilir olay anlamıyla gerçekleştirilebilir.

LDPC kodlar pratiksel olarak iyidir. Olumsuz tarafı ise kodlama karmaşıklığının yüksek olmasıdır. Bu sorunu aşmanın bir yolu, iki kümeli graflardan kaskat graflara kodların yapısını modifiye etmektir. Bu durum, pratik amaçlar için farklı çözümler gerektirmeyecektir.

Tanner tarafından önerilen LDPC kodları, iki kümeli grafik düğümlerinden oluşur. Bir kümesi kod kelimelerinin ögesine karşılık gelen değişken düğümleri, diğer taraftaki düğüm kümesi ise, kodu tanımlayan parite kontrol sınırlamalarına karşılık gelen kontrol düğümleri tarafından gösterilmiştir. Düzenli LDPC kodlarında aynı tipteki bütün düğümler aynı dereceye sahiptir. Örnek olarak (3,6) düzenli LDPC kodu değişken düğümlerin derecesi 3, kontrol düğümlerin derecesi 6 'dır. Düzensiz LDPC kodu için bazı dağılımlara göre düğümlerin her biri için setin derecesi seçilir.

Kod topluluk dizisi $C^n(\lambda, \rho)$ buna bağlı derece dağılım çifti (λ, ρ) olarak kabul edelim. n ise kod uzunluğu olarak tanımlanmış olsun derecelerin dağılımı denklem (2.46) ile elde edilir.

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1} \quad (2.46)$$

$$(\rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1})$$

Değişken düğümünün, derece dağılımı (kontrol düğümünün derece dağılımı) belirtilir. Daha açık olarak $\lambda_i(\rho_i)$ i derecesinde değişken(kontrol) uçlarından çıkan uçların kesrini simgeler.

Örnek olarak (3,6) düzenli kod $\lambda(x) = x^2$ ve $\rho(x) = x^5$ olsun. Maksimum değişken ve kontrol derecesi d_v ve d_c olarak gösterilmiştir bu kodun “n” tane değişken düğümlerine sahip olduğunu düşünelim. i derecesinde değişken düğümlerin sayısı denklem (2.47) ile ifade edilir.

$$n \frac{\lambda_i / i}{\sum_{j \geq 2} \lambda_j / j} = n \frac{\lambda_i / i}{\int_0^1 \lambda(x) dx} \quad (2.47)$$

Bundan dolayı bütün değişken düğümlerinde, çıkan düğümlerin toplam sayısı olan E değeri denklem (2.48) ile ifade edilir.

$$E = n \sum_{j \geq 2} \frac{\lambda_j / j}{\int_0^1 \lambda(x) dx} = n \frac{\lambda_i / i}{\int_0^1 \lambda(x) dx} \quad (2.48)$$

Benzer durumda, kod “m” tane kontrol düğümlerine sahip olsun. Bu durumda E aynı zamanda denklem (2.49) deki gibi ifade edilir.

$$E = m \frac{1}{\int_0^1 \rho(x) dx} \quad (2.49)$$

Her iki ifade de E olduğundan, denklem (2.50)’deki gibi her iki ifade eşitlenerek elde edilir.

$$m = n \frac{\int_0^1 \rho(x) dx}{\int_0^1 \lambda(x) dx} \quad (2.50)$$

Genel olarak, bütün bu kontrol eşitlikleri lineer olarak bağımsız olduğu kabul edilirse, tasarım oranı denklem (2.51)’de olduğu gibidir.

$$r(\lambda, \rho) := \frac{n-m}{n} = 1 - \frac{\int_0^1 \rho(x) dx}{\int_0^1 \lambda(x) dx} \quad (2.51)$$

Herhangi $\varepsilon > 0$ değeri için, $\delta < \delta^*$, $n(\varepsilon, \delta)$ uzunluğu $P(\varepsilon, \delta)$ içerir ve hemen hemen bütün kodlar için $C^n(\lambda, \rho)$ $n > n(\varepsilon, \delta)$ ve mesaj geçen algoritma kod çözümünü oluşturan $P(\varepsilon, \delta)$ tekrarlamalarla ve δ parametresiyle kanal üzerinden akan iletim varsayımı ε den daha küçük bit hata olasılığına sahiptir. Eğer tekrarlamaların sayısı sabitlenirse, grafin ve kanalın çeşitli farkındalıklarının performansı beklenen değer etrafında toplanır, bu toplanma kod genişliğiyle üsteldir (bu üstellik genel olarak derece distribüsyon çiftine, seçilen mesajı geçiren dekodere ve kanal parametresine bağlıdır). Bu nedenle yeterli uzunluktaki kodların performansını karakterize etmek yerine ortalama performansını bulmak yeterlidir. Ancak, sonlu uzunluktaki kod kelimeleri için ortalama performansı hesaplama işlemi düşünüldüğü kadar kolay değildir.

Başta yayılan mesajların dağılımı, kanal tarafında belirtilir. Bu dağılım birleşmiş hata olasılığına sahiptir. Eşik değeri, tekrarlama sayısı sonsuza yaklaştığında hata birleşmiş olasılığı sıfıra yaklaşacaktır. Mesaj dağılımının yavaş yavaş gelişmesi de kanal parametresinin en kötüsü olarak tanımlanmaktadır. Mesaj dağılımının evrimini izleme işlemi de yoğunluk evrimi olarak adlandırılır. Etkili sayısal işlemler, BP dekoderlerin önemli durumlarında yoğunluk evrimini uygulamak için geliştirilmiştir. Uygun olarak seçilen düzensiz kodlar ile kayda değer performans başarılabilen sonuçlar mevcuttur. Bir çok hafızasız kanallar için daha çok karmaşık ve yeni metotlar optimizasyon sorununu aşmak için geliştirilmektedir.

$C \in C^n(\lambda, \rho)$ oranlı LDPC kodu olsun. C , BSC üzerinde δ çapraz geçiş olasılığıyla kullanılsın ve her kod kelimesinin eşit olasılıkla kullanıldığını farz edelim. İkili entropi fonksiyonu ve parametre tanımı denklem (2.52)'deki gibidir.

$$r > 1 - h(\delta) / h(p^*), \quad p^* = \frac{1 + (1 - 2\delta)^{d_c}}{2} \quad (2.52)$$

Bit veya blok hata olasılığı 0'dan uzak, n' den bağımsız, bir sabit tarafından sınırlandırılmıştır. BSC' nin kapasitesi herhangi bir sonlu kod için $p^* > \frac{1}{2}$ ve $h(p^*) < 1$ olsun bundan dolayı denklem (2.53)'de belirtilen hızda hesaplamayla kapasiteye olan aralık denklem (2.54) ile ifade edilen değere yaklaşmakta olup d_c üstel olarak 0'a yaklaşmaktadır.

$$1 - \frac{h(\delta)}{h(p^*)} < 1 - h(\delta) \quad (2.53)$$

$$\frac{h(\delta)}{2 \ln 2} (1 - 2\delta)^{2d_c} \quad (2.54)$$

Gallager yukarıda belirtilen durumu düzenli kodlar için ortaya koymuştur. Maksimum kontrol düğüm derecesine geçilmesiyle birlikte düzensiz kodlar için de aynı durumun söz konusu olduğu ortaya çıkmıştır.

2.1.3.7. Kod Çözme Teorileri

LDPC kodlarının çözülmesinde, çokça kullanılan teoriler; BPA (İnanç Yayılım Algoritması) ve MPA (Mesaj Geçen Algoritma) algoritmaları farklı isimler altında ve çeşitli defalar bağımsızca denenmiştir.

2.1.3.8. İnanç Yayılımı

LDPC kodları için kullanılan kod çözücü algoritmalara mesaj geçen algoritmalar denir. Her algoritma mesajlarının çemberi, mesaj bağlantısından doğruluk bağlantısına ve doğruluk bağlantısından tekrar mesaj bağlantısına yollandığından, bu adı alır. Mesaj bağlantısından doğruluk bağlantısına geçen mesajlar, mesaj bağlantısının değerine bağlı ve komşu doğruluk bağlantısından bu mesaj bağlantısına geçen bazı mesajlardır. V mesaj bağlantısından, C doğruluk bağlantısına geçen mesajlar bir önceki çemberdeki, C' den V' ye geçen mesajlara eklenmemelidir. Bu durum, doğruluk bağlantısından mesaj bağlantısına yollanan mesajlar içinde geçerlidir.

Mesaj geçen algoritmaların önemli alt sınıflarından biri de inanç yayılımı algoritmasıdır. Bu algoritma Gallager' in çalışmasında vardır, ayrıca bu algoritma AI (Artificial Intelligence) topluluğunda da kullanılır. Bu algoritmadaki köşelerden yollanan mesajlar olasılıktır veya yayılımdır. Tam olarak söylemek gerekirse, V mesaj bağlantısından, C doğruluk bağlantısına

yollanan mesajın şu olasılığı vardır. V , bu mesaj bağlantısının sahip olduğu değere eşittir. Diğer yandan C' den V' ye yollanan mesajlar, V' nin değerinin, V' den başka diğer önceki çemberdeki C' ye geçen tüm mesajların değerine eşit olma olasılığını verir.

Bu durumda, daha sonra açıklayacağımız üzere, bağımsızlık varsayımı denilen belli bir varsayımın ışığı altında formüller üretilebilir. Değişken x çifti için $L(x)=\Pr [x=0] / \Pr [x=1]$ in x ile aynı olduğunu düşünelim. Değişken y çifti için, $L(x | y)=\Pr [x=0|y] / \Pr [x=1|y]$ olarak yazılır. Benzer şekilde x 'in benzerleri $\ln L(x)$ ' dedir ve çevresel x 'in benzerleri olan y 'nin özellikleri $\ln L(x|y)$ ' dedir.

Eğer x eşit bir rastgele değişken ise, Bayes kuralına göre $L(x|y)=L(y|x)$ ' tir. Bununla beraber, eğer y^1, \dots, y^d bağımsız rastgele değişken, denklem (2.55)'deki eşitlik ile kurulabilir.

$$\ln L(x|y_1, \dots, y_d) = \sum_{i=1}^d \ln L(x|y_i) \quad (2.55)$$

x^1, \dots, x^l nin çift rastgele değişken ve y^1, \dots, y^l nin rastgele değişken olduğunu farz edelim. Hesaplamalar yapılırsa denkem(2.56) elde edilir.

$$\ln L(x_1 \oplus \dots \oplus x_l | y_1, \dots, y_l) = \ln \frac{1 + \prod_{i=1}^l \tanh(l_i / 2)}{1 - \prod_{i=1}^l \tanh(l_i / 2)} \quad (2.56)$$

LDPC kodları için inanç yayılımı algoritmaları bu iki tanımlamadan elde edilir. doğruluk bağlantıları tanımlanan değerinde "0" çemberinde, eşdeğerlerine yani tüm harici köşelere gönderir. Örneğin kullanılan kanal, p hata olasılığına sahip BSC ise, mesaj bağlantısının yanındaki tüm doğruluk bağlantısına yollanan ilk mesaj $\ln(1 - p) - \ln p$ 'dir (eğer bağlantının değeri "0" ise). Eğer bağlantının değeri "1" ise bu değer negatiftir. Algoritmanın tüm sıradaki çemberlerinde doğrulama bağlantısı (c)' ye göre yakın mesaj bağlantısına (v) yollar. V mesaj bağlantısı 1'nci bağlamdaki c ' nin yerine yakın doğrulama bağlantılarından giriş yapan benzerlerindeki ve belirtilen değerdeki benzerlerine, c doğrulama bağlantısını yollar.

$m_{vc}^{(l)}$ 'nin, algoritmanın l ' nci çemberindeki mesaj bağlantısından (v) doğrulama bağlantısına yollanan mesaj olduğunu düşünelim. Aynı şekilde $m_{cv}^{(0)}$ 'yi açıklayalım. "0" çemberinde, $m_{vc}^{(l)}$ c ' den bağımsız olarak belirlenen v değerdeki mesaj bağlantısının benzeridir. Bu değeri m_v ile adlandırırız. Daha sonra inanç yayılımı altındaki mesajlar için güncellenen eşitlik denklem(2.57) ile ifade edilir.

$$m_{vc}^{(l)} = \begin{cases} m_v & l = 0 \\ m_v + \sum_{d \in C_v / (c)} m_{cv}^{(l)} & l \geq 1 \end{cases} \quad (2.57)$$

$$m_{cv}^{(l)} = \ln \frac{1 + \prod_{v^l \in V_c / (v)} \tanh(m_{v^l c}^{(l)} / 2)}{1 - \prod_{v^l \in V_c / (v)} \tanh(m_{v^l c}^{(l)} / 2)}$$

C_v mesaj bağlantısında (v) hatalı olan doğrulama bağlantısının toplamı ise V_c 'de c doğrulama bağlantısında hatalı olan mesaj bağlantılarının toplamıdır.

Doğrulama bağlantısındaki hesaplamalar fazladan köktekini göstererek denlem (2.58)'deki gibi basite indirgenebilir. Ancak $\tanh(x)$ 'in değeri negatif olduğundan, işaretin yolunu ayrı ayrı elde bulundurmamız gerekir.

$$m_{cv}^{(l)} = \tau^{-1} \left(\sum_{v^l \in V_c / (v)} \tau(m_{v^l c}^{(l-1)}) \right) \quad (2.58)$$

Pratikte, birinci gerçekleştiğinde, geçen benzerlerin kesin sonuca yakın olduğu durumda, inanç yayılımı maksimum çember sayıları için son bulabilir. Kesin benzerlik, $\ln L(x|y) \rightarrow \infty$ olduğunda $\Pr[x=0|y]=1$ ve $-\infty$ olduğunda ise $\Pr[x=1|y]=1$ şeklinde olur.

İnanç yayılımının en önemli konularından biri ise zaman geçmesidir. Algoritma grafikteki kenarlarda dolaştığı zaman ve grafik az ise, dolaşılacak traversler çok küçüktür. Ayrıca, algoritma mesaj bağlantılarının sayısında lineer olan birçok operasyonu kullanır. İnanç yayılımı ile ilgili en önemli notlardan biri ise, algoritma sırasında geçen mesajların tamamen kanala bağımlı olmalarına rağmen, kullanılan kanala, algoritmanın bağımsız olmasıdır. Önemli sorunlardan biri ise inanç yayılımı ile maksimum benzerlik kod çözümü arasındaki

ilişkidir. Sorunun sebebi ise, inanç yayılımının genelde maksimum benzerlik kod çözümünden daha güçsüz olmasıdır. Aslında inanç yayılımından daha fazla hata çözen maksimum benzerlik kod çözümü iken, LDPC kodlarının sınıflarını yapmak kolaydır.

2.1.3.9. İnanç Yayılımının ve Yoğunluk Evriminin Asimptotik Analizi

İnanç yayılım algoritmasının her çemberinden geçen mesajlar, rastgele değişkenlerdir. Eğer algoritmadaki her çemberde giren mesajlar bağımsızsa, güncellenen denklem, tanımlamaya bağlı olan denklemi doğru olarak hesaplar.

İnanç yayılımı, olasılık teorisi ve toplamadan elde edilen kombinasyon kullanılarak analiz edilebilir. Eğer l tamir edilmiş, n ve r yeteri derecede büyükse, o zaman rastgele grafikler için l derinliğinin komşuluğu (mesaj bağlantılarının) ağaçtır. Öte yandan l çemberleri için bu bağlantıdaki inanç yayılım algoritması bağlantının benzerliğini hesaplar. Bu bağlantılar iyi bağlantılardır. Diğer bağlantılar sıkıntı yaratabilecektir. İnanç yayılımının beklenen diğer bir davranış, ağaçtaki algoritmaların analiz edilerek hesaplanmasıdır. Analizin bu basamağı genelde sabittir.

Martingale tartışmaları ve ağaç farz etmesi (büyük grafikleri tutan, l İterasyon numarası) ağaçtaki inanç yayılımı öğrenim analizi, birçok iterasyonlar için tüm grafikteki gerçek davranışları gösterdiğini kanıtlar. Grafikteki güzel mesaj bağlantıları arasındaki hata olasılıkları inanç yayılımının davranışına göre hesaplanabilir. Uygun derece dağıtım için grafikteki güzel mesaj bağlantılarının hata olasılığı bir şekilde azaltılabilir. Küçük parçaları sabit ve küçük olduğu sürece, hata olasılığına çok kısa sabit bir dönemde katkı yaparlar ve etkileri görülmeden kaybolurlar. Bu durum, asimptotik analizler ile ilgilidir.

Ağaçtaki inanç yayılımının beklenen davranışının analizi, köşelerden geçen mesajların yoğunluk fonksiyonlarının yeniden oluşumunu açıklar. Genel makine gösterir ki gerçekten geçen mesaj yoğunluğu beklenen yoğunluğa çok yakındır. İterasyonlar sırasında beklenen yoğunluğu araştırırken algoritmanın gerçek davranışının güzel bir örneği görülür.

Eğer $x_1 \dots x_d$, bazı fazlardan gruplar (G) üzerindeki i ve d rastgele değişkenler ise ve f sadece x ' in genel yoğunluğu ise $x_1 + \dots + x_d$ 'nin yoğunluğu (F) f 'nin d -fold dönüşümlü gücüne eşittir. Algoritmanın i çemberinde doğrulama bağlantılarından mesaj bağlantılarına gönderilen mesajların (m_{cv}^i) genel yoğunluk fonksiyonunun (g_i) olduğunu ve algoritmadaki "0"

çemberinden yollanan mesajların benzerlerinin m_v , mesaj yoğunluğunun f olduğunu düşünelim. d derecesindeki λ_d olasılığına sahip mesaj bağlantılarının her köşeyle bağlantılı olduğu ve ρd olasılığına sahip bir grafiğin rastgele olduğunu farz edelim. Buna göre $i+1$ çemberindeki mesaj bağlantılarından doğrulama bağlantılarına yollanan mesajların beklenen yoğunluğu ise $f \otimes \lambda(g_i)$ 'dir.

Doğrulama bağlantılarındaki yoğunlukların dönüşümünü test etmek için γ operatörünün kullanılması gerekir. Bu durumda f_i yoğunluğunda denklem (2.59)'da belirtilen yoğunluk dönüşümü elde edilir.

$$f_{i+1} = f \otimes \lambda(\Gamma^{-1}(\rho(\Gamma(f_i)))) \quad (2.59)$$

Gelen mesajların, bağımsız rastgele değişken olmasına göre yoğunluk dönüşümü uzun olabilir. Sonsuz bir grafik için bu sadece küçük sayıdaki çemberler için geçerlidir.

2.1.3.10. Mesaj Geçen Yapı

Tanner graf tabanlı bir kod çözme işlemidir. i 'inci bit düğümüne ait hatlardaki mesaj, $\Pr[c_i=1]$ olasılığının kestirimleridir. Bu olasılıklar düğümlerde çeşitli yollarla birleştirilir. Her bir bit düğümündeki ilk olasılık kestirimi, kanalın yumuşak çıkışlarından elde edilmiş olasılıklardır. Bit düğümü bu olasılığı kendine bağlı hatlar yardımıyla parite kontrol düğümlerine iletir. Parite kontrol düğümleri, temsil ettiği parite kontrol denklemindeki bitler için yeni bir olasılık kestirimi yaparak elde ettiği bu olasılığı, bit düğümüne geri yollar. c' kod sözcükleri $c'H^T=0$ eşitliğini sağlayana kadar (geçerli bir kod sözcüğü elde edilene kadar) iterasyonlara devam edilir.

Bilgiler sırasıyla parite düğümlerinden bit düğümlerine ve bit düğümlerinden parite düğümlerine geçer. Gerekli koşul sağlandığında iterasyonlara son verilir. Son adımda her bir bit düğümde elde edilen olasılıklar normalize olarak çarpılır.

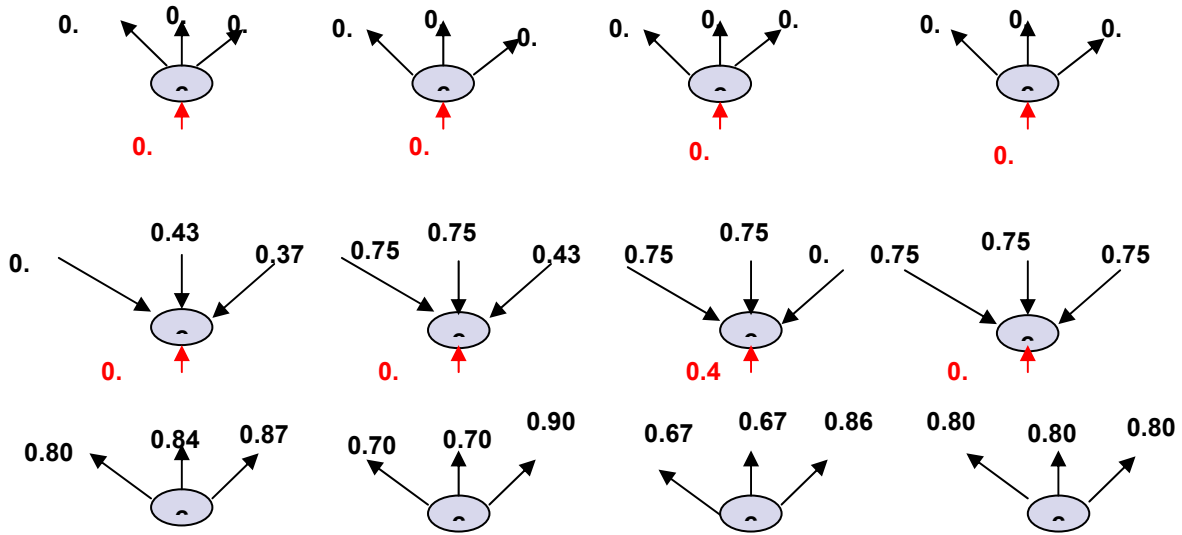
(3,4) LDPC koduna ait kod sözcükleri aşağıda verilen sıralı dizinler olarak kabul edildiğinde,

```
000000000000  111111111111
010110000100  101001111011
000011100001  111100011110
010101100101  101010011010
```

$Pr[c_i=1]$ ($i=1,2,3,\dots,12$) kanaldan alınan olasılıklar sırasıyla

0.9 0.5 0.4 0.3 0.9 0.9 0.9 0.9 0.9 0.9 0.9 0.9 olarak tanımlanabilir.

İlk dört bit düğümü için kod çözme algoritması ise Şekil 2.18’de görülmektedir.

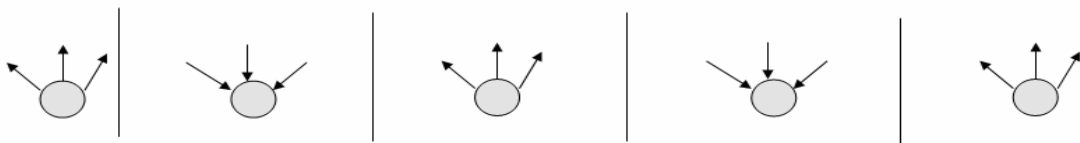


Şekil 2.18: Kod çözme algoritması

Şekil 2.18’de verilen kod çözme algoritmasında elde edilen tüm bit düğümleri için olasılık değerleri Tablo 2.1’de görülmektedir.

Mesaj Geçiren Algoritma
Tüm bit düğümleri için olasılık değerleri

	yukarı	aşağı	aşağı	aşağı	yukarı	yukarı	yukarı	aşağı	aşağı	aşağı	yukarı	yukarı	yukarı
C1	0.900	0.500	0.436	0.372	0.805	0.842	0.874	0.594	0.640	0.656	0.968	0.962	0.959
C2	0.500	0.756	0.756	0.436	0.705	0.705	0.906	0.640	0.690	0.630	0.791	0.751	0.798
C3	0.400	0.756	0.756	0.500	0.674	0.674	0.865	0.790	0.776	0.644	0.807	0.820	0.897
C4	0.300	0.756	0.756	0.756	0.804	0.804	0.804	0.749	0.718	0.692	0.710	0.742	0.765
C5	0.900	0.500	0.372	0.372	0.759	0.842	0.842	0.611	0.694	0.671	0.976	0.966	0.970
C6	0.900	0.436	0.500	0.756	0.965	0.956	0.874	0.608	0.586	0.643	0.958	0.962	0.952
C7	0.900	0.436	0.500	0.372	0.842	0.805	0.874	0.647	0.628	0.656	0.967	0.969	0.965
C8	0.900	0.436	0.436	0.756	0.956	0.956	0.843	0.611	0.605	0.656	0.963	0.964	0.956
C9	0.900	0.372	0.372	0.500	0.842	0.842	0.759	0.722	0.694	0.703	0.980	0.982	0.981
C10	0.900	0.372	0.500	0.500	0.900	0.842	0.842	0.690	0.614	0.654	0.964	0.974	0.970
C11	0.900	0.372	0.436	0.756	0.956	0.943	0.805	0.667	0.608	0.676	0.967	0.974	0.965
C12	0.900	0.500	0.372	0.756	0.943	0.965	0.842	0.565	0.642	0.657	0.969	0.957	0.955



Tablo 2.1: Tüm bit düğümleri için olasılık değerleri

2.1.3.11. Sert Karar Kod Çözme

Zor Karar Çözme Algoritmasının daha iyi anlaşılabilmesi için bu algoritma ile çalışan basit bir türünü tanımak yararlı olacaktır. Hata serbest alınacak şekilde, kod kelimesini örnek olarak $C=[10010101]$ kabul edelim. İlk aşamada bütün V-düğümüleri C_1 'in biri için doru olduğunu düşünürken biti içeren kendilerini C-düğümüleri f_j mesajı gönderir. C_1, f_j . Bu durumda V-düğümü sahip olduğu yalnız bilgi C'ini 1'inci biti ile alınan eşittir. Bu durumda, C_0 1 içeren mesajları Tablo 2.2'de görüleceği gibi f_1 ile f_3 'e gönderir, C_1 düğümü ise $y_1(1)$ içeren mesajları f_0 ile f_1 'e gönderir.

c- düğümü	alınan / gönderilen			
f_0	alınan : $c_1 \rightarrow 1$	$c_3 \rightarrow 1$	$c_4 \rightarrow 0$	$c_7 \rightarrow 1$
	gönderilen : $0 \rightarrow c_1$ $0 \rightarrow c_3$ $1 \rightarrow c_4$ $0 \rightarrow c_7$			
f_1	alınan : $c_0 \rightarrow 1$	$c_1 \rightarrow 1$	$c_2 \rightarrow 0$	$c_5 \rightarrow 1$
	gönderilen : $0 \rightarrow c_0$ $0 \rightarrow c_1$ $1 \rightarrow c_2$ $0 \rightarrow c_5$			
f_2	alınan : $c_2 \rightarrow 0$	$c_5 \rightarrow 1$	$c_6 \rightarrow 0$	$c_7 \rightarrow 1$
	gönderilen : $0 \rightarrow c_2$ $1 \rightarrow c_5$ $0 \rightarrow c_6$ $1 \rightarrow c_7$			
f_3	alınan : $c_0 \rightarrow 1$	$c_3 \rightarrow 1$	$c_4 \rightarrow 0$	$c_6 \rightarrow 0$
	gönderilen : $1 \rightarrow c_0$ $1 \rightarrow c_3$ $0 \rightarrow c_4$ $0 \rightarrow c_6$			

Tablo 2.2: Sert karar kod çözme

İkinci basamakta bütün kontrol düğümüleri f_j ' ler bağlandıkları bütün değişken düğümlere cevap hesaplarlar. Cevap mesajı Tablo 2.3'de görülebileceği gibi f_j ' ye bağlı diğer V-düğüm C_1 için f_j ' ye doğru olduğunu düşünerek bit içerir. Bir C-düğümü f_j üç tane V-düğümünden aldığı mesaja bakar ve dördüncü V-düğümün parite kontrol eşitliğini doldurması için gerekli bit değerini hesaplar.

v- düğümü	y_i	alınan	kontrol düğümlerinden mesajlar		karar
c_0		1	$f_1 \rightarrow 0$	$f_3 \rightarrow 1$	1
c_1		1	$f_0 \rightarrow 0$	$f_1 \rightarrow 0$	0
c_2		0	$f_1 \rightarrow 1$	$f_2 \rightarrow 0$	0
c_3		1	$f_0 \rightarrow 0$	$f_3 \rightarrow 1$	1
c_4		0	$f_0 \rightarrow 1$	$f_3 \rightarrow 0$	0
c_5		1	$f_1 \rightarrow 0$	$f_2 \rightarrow 1$	1
c_6		0	$f_2 \rightarrow 0$	$f_3 \rightarrow 0$	0
c_7		1	$f_0 \rightarrow 1$	$f_2 \rightarrow 1$	1

Tablo 2.3: Sert karar kod çözme akışı

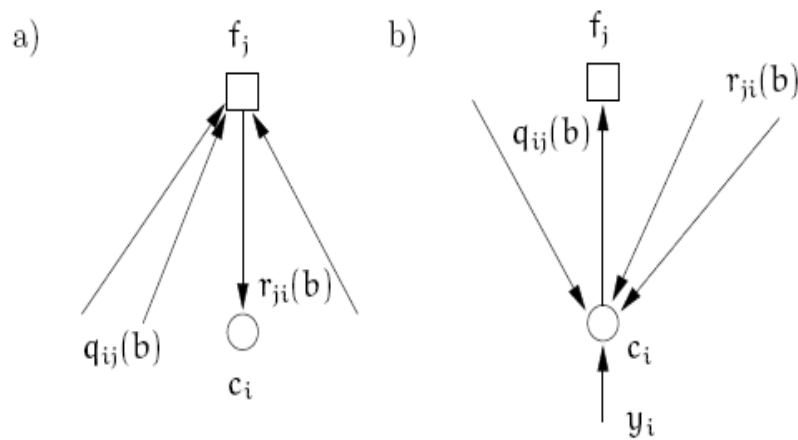
Eğer bütün kontrol eşitlikleri dolarsa bu durum ortaya çıkar. Bütün algoritmalar çevrim içerir, bundan dolayı diğer olasılıkları bitirmek için oluşacak çevrim sayıları için eşik değeri vardır.

Daha sonra V-düğümü kontrol düğümlerinden mesaj alır ve bu ek bilgiyi algoritma alınan bitin tam olduğunda karar vermek için kullanır. Bunu yapmanın basit yolu ise yoğunluk oyudur. Örnekten de görüleceği gibi her bir V-düğümü üç bitleriyle alırken üç tane bilgi kaynaklarına sahiptir. Kontrol düğümlerinden alınan orijinal bit ve iki öneri şimdi V-düğümü kontrol düğümlerine doğru değer için zor karar ile mesaj gönderebilir. C_1 son basamakta sıfır için seçilirse kod çözme işlemi sona erer. Böylece istediğimiz iletim hataları ve bütün kontrol eşitlikleri düzeltilir.

2.1.3.12. Yumuşak Karar (Soft Decision) Kod Çözme

LDPC kodlarının BP kavramlarına dayanan Yumuşak-Karar kod çözümü, kod çözme işlerinde daha iyi performans gösterir ve tercih edilen metottür. Aslında tamamen zor-karar kod çözme ile benzerdir.

$P_i = P_r (c_i = 1 | y_i)$ ve q_{ij} v-düğümünden c_i c-düğümüne f_j gönderilen mesajdır. Her mesaj daima iki çift içerir $q_{ij}(0)$ ve $q_{ij}(1)$. Bunun nedeni de y_i 0 veya 1 olma inancına ayanmaktadır. r_{ij} c-kontrol düğmesinden f_j v-değişken düğümüne c_i gönderilen mesaj olarak adlandırılır. Aynı şekilde her mesaj $r_{ij}(0)$ ve $r_{ij}(1)$ çiftine sahiptir. Şekil 2.19'da görüleceği gibi bütün değişken düğümler kendi q_{ij} mesajlarını gönderir. Bu basamakta hiçbir bilgi uygun olmadığı sürece; $q_{ij}(1) = P_i$ ve $q_{ij}(0) = 1 - P_i$ olarak kabul edilir.



Şekil 2.19: Yumuşak karar çözme yapısı

Daha sonra kontrol düğümü kendi cevap mesajlarını (r_{ij}) denklem (2.60) de belirtildiği gibi hesaplar. Ayrıca c_i hariç değişken düğümler arasında “1” lerin alt sayısının çift bulunma olasılığını hesaplar.

$$r_{ij}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in V_{j,i}} (1 - 2q_{i'} j(1)) \text{ ve } r_{ji}(1) = 1 - r_{ji}(0) \quad (2.60)$$

Bu olasılık c_i sıfır olduğunda $r_{ji}(0)$ ‘nin olasılığına eşittir. Bu bilgi elde edilen cevapları kontrol etmek için kullanılır. Değişken düğüm de, kontrol düğümünde kendi cevap mesajını geliştirir. Bu geliştirme denklem (2.61)’de görülen eşitlikler sağlanarak yapılır.

$$q_{ij}(0) = K_{ij}(1 - P_i) \prod_{j' \in \mathcal{C}_i / j} r_{j',i}(0) \text{ ve } q_{ij}(1) = K_{ij} P_i \prod_{j' \in \mathcal{C}_i / j} r_{j',i}(1) \quad (2.61)$$

K_{ji} ise $q_{ij}(0) + q_{ij}(1) = 1$ sağlayan değerlerden seçilir. \mathcal{C}_i / j ise f_j hariç bütün kontrol düğümleri demektir. Bu noktada v -düğümü kendi değişken c_i ’nin kesin tahmini \hat{c}_i geliştirir. Bu daha büyüğünü seçerek “0” ve “1” için olasılığı denklem (2.62) ‘de olduğu gibi hesaplayarak yapılır.

$$Q_i(0) = K_i(1 - P_i) \prod_{j \in \mathcal{C}_i} r_{ji}(0) \text{ ve } Q_i(1) = K_i P_i \prod_{j \in \mathcal{C}_i} r_{ji}(1) \quad (2.62)$$

Bu denklem q_{ij} ‘yi hesaplamaya benzer şekildedir. Ancak bütün c -düğümlerinde de bilgi kullanıldığı için;

$$\hat{c}_i = \begin{cases} 1 & \text{eğer } Q(1) > Q_i(0) \\ 0 & \text{diğer durumda} \end{cases}$$

koşulu ile Yumuşak Karar verme gerçekleştirilir. Eğer kesin tahmin edilen kod kelimesi parite kontrol eşitliklerini doldurursa, algoritma sonlandırılır. Yumuşak Karar kod çözme algoritmaları basit çevirimlerdir. Genelde BSC kanalları için uygundur. Performans geliştirmek amacıyla modifiye edilebilir. Bunun yanında performans olasılıklarının çok çarpımları nedeniyle sayısız durağan problemler ortaya çıkarmaktadır. Geniş blok uzunluklarında sonuç sıfıra çok yaklaşmaktadır.

2.2. ÇOK SEVİYELİ LDPC

Birden fazla LDPC Encoder/Decoder'in kullanılması ile blok şeklindeki verilerin paralel kollardan bir modülatöre gönderilip, demodülatörden ayrıştırılarak paralele kollarda çıkış işareti alınması ile çok seviyeli LDPC kodlama yapısı elde edilir.

Çok Seviyeli kodlama mantığı, Imai ve Hirakawa [9] tarafından bilimsel literatüre kazandırılmıştır. Buna ek olarak, bölümlene ayrıştırma fikri ise ilk olarak Ungerboeck [10] tarafından gündeme getirilmiştir. Çok seviyeli kodlamanın amacı, 2^l genişliğinde sembol yerleştirme (signal constellation) durumunda, minimum öklid uzunluklarını ifade eden dahili setlerin sayısını arttırmaktır. Seviye dereceleri (l) farklı dahili öklid uzaklığı setlerine sahiptir. Çok seviyeli kodlamada, kodlama seviyelerinin farklı etkisinden korunma amacı güdülmektedir. Minimum öklid uzaklığına sahip yüksek seviye güçlü kodlama seviyelerinden (düşük kodlama) korunurken, güçlü kodlamalarda zayıf kodlamadan korunabilecektir.

Bu kodlamalar için, Konvolisyonel çok seviyeli dekoder en üst seviyeden başlarken, bit seviyesine kadar işlemi tamamlayıp bir sonraki seviyede işleme devam eder. Bu işlem tüm seviyeler tamamlanana kadar devam eder. Alıcı açısından bu durum, yüksek seviyeli kodların her seviyede başarılı bir şekilde kodlanmasını sağlayan bir avantaja dönüşmektedir. Her sıralı (l) kadar bitin fiziksel kanallarda transfer işlemi (l) kadar farklı kanalda paralel olarak transferine eşittir. İteratif kodlamaya dayanan hata kontrol amaçlı kodlamada, düşük seviyede Sinyal-Gürültü (SNR) oranına rağmen iyi derecede sonuçlar alınabilmektedir. Ancak bu sonuçların alınabilmesi için, blok uzunluğu ile lineer olarak artan bir kodlama karmaşıklığı ile karşılaşılması söz konusudur. Bit başına harcanan gücün etkinliğini koruyarak bant genişliğini arttıran sistemler tasarlanması ile çok seviyeli kodlama, diğer kodlama türleriyle yarışabilecek düzeye gelmiştir.

Turbo kodlar kapasite sınırlarına yakın derecede performansa gösterirken seviyeler arasında optimallik problemine ilişkin farklı görüşler ortaya atılmıştır. Bu görüşler sonucunda, çok seviyeli kodlamanın kapasite sınırlarına yaklaşabilmesi için seviyelerin oranının, birbiriyle uyumlu denk kanal kapasitelerine eşit olması halinde gerçekleşebilecektir. Ancak pratikte sonlu uzunluktaki kod ile, yüksek seviyelerde mükemmel kodlama başarısı geçersiz duruma gelmekte ve çok seviyeli kodlama optimal olma şansını kaybetmektedir [10].

Kodlanmış verinin transferinde, iteratif kodlama yapılarak, sert karar (hard decision) verilebilmesinin yanı sıra ayrıca verilerin güvenilir olarak kodlanarak iletilebilmesi sağlanmaktadır. Diğer bir avantajı ise, her bir bitin güvenilirlik bilgisinin yüksek seviyede her iterasyonda gönderilebilmesine olanak sağlamasıdır. Her iterasyonda bilgi değişimi sağlanmakta ve bu değiştirme işlemi demodülatörde yapılmaktadır. [11],[12],[13].

2.3. OFDM YAPISI

2.3.1 OFDM İle İlgili Kavramlar

OFDM tekniğinin anlaşılabilmesi için, bu teknik ile kullanılan bazı kavram ve tanımların bilinmesi gerekmektedir. Bu nedenle temel tanım ve kavramlar açıklanmıştır.

2.3.1.1. Doppler Saçılımı

Alıcı ve verici antenlerin sabit hızda birbirleriyle bağıl olarak hareket etmeleri halinde alıcı tarafında sabit frekans kayması oluşacaktır. B_D hareketli birimin ve baz istasyonunun göreceli hareketi veya kanaldaki cisimlerin hareketiyle meydana gelen spektral genişlemenin ölçüsüdür. Şekil 2.20 ile doppler saçılımı daha iyi anlaşılacaktır.

Bu yayılma en büyük Doppler kayması miktarına eşittir ($B_d=f_m$). Alınan işaretin toplam band genişliği, temel band işaretin band genişliği ve doppler kayması miktarı ile belirlenir. Eğer temel band işaretin band genişliği B_D 'ye göre çok büyük ise alıcıya ulaşan işaretle doppler yayılması etkisi ihmal edilebilir.

$$f_d = f_c \frac{v}{c} \quad (2.63)$$

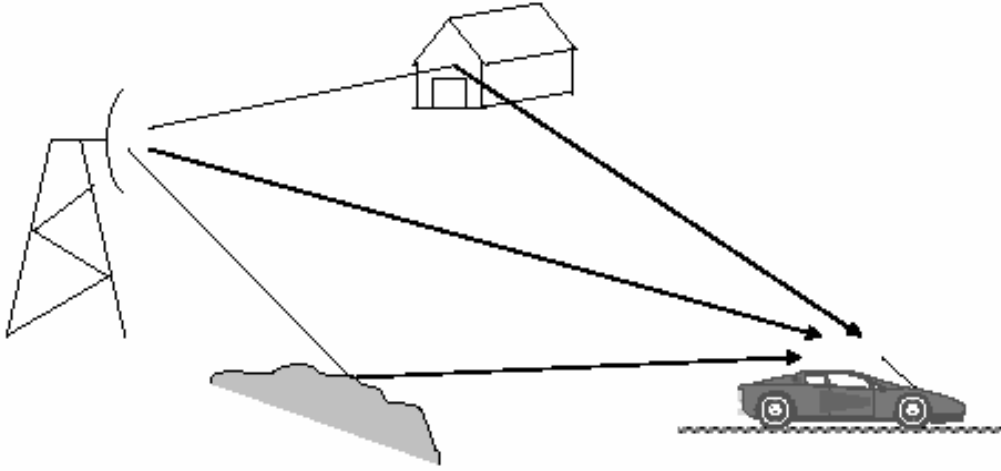
f_c : Taşıyıcı frekansı, v : Aracın hızı, $c = 3.10^8 m/s$:Işık hızı.

Doppler saçılımı değişik yollarda oluşan en geniş frekans kayması olarak tanımlanır. Kanal zamanı (T_d) doppler saçılımının tersi olarak alınır ve ilgili zaman seçici aralığının ölçülmesinde kullanılır.

$$T_d = 1/f_d \quad (2.64)$$

T_s 'yi gönderilen sinyalin zamanı olarak kabul edelim. Şayet $T_s \gg T_d$ ise gönderilen sinyal zamanla değişen sönmülemeye maruz kalır. $T_s \ll T_d$ olması durumunda ise, iletim esnasında önemli değişiklik olmaz ve gönderilen sinyalin değişik zaman dilimlerinde aynı tipte

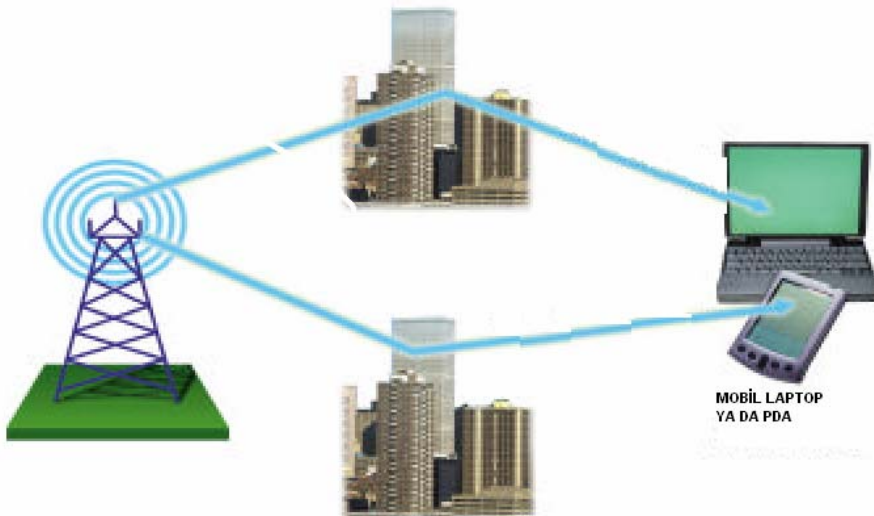
sönümlenme gerçekleşir. Bu durumda kanal zamanda düzgün-seçici olmayan sönümlenmeli kanal olarak adlandırılır.



Şekil 2.20: Doppler saçılımı

2.3.1.2 Kablosuz Ortamda İletim

Telsiz radyo kanallarında, verici tarafından iletilen işaret alıcı antene çeşitli yolları izleyerek ulaşır. Yayılan elektromanyetik dalgalar, gezgin radyo iletişimi durumunda çevrede bulunan binalardan ve engellerden yansır, kırınır ya da saçılır. Kablosuz haberleşmede; Şekil 2.21’de görüldüğü gibi, değişik engellerden oluşan yansımalar nedeniyle gönderilmiş semboller alıcılarda çoklu gecikmelere neden olmaktadır. Hatta bazı durumlarda alıcı ile verici arasında görüş bile olmamaktadır.



Şekil 2.21: Kablosuz haberleşmede çoklu yol gecikmesi

Yansımalarından oluşan her yol farklı güç kaybı, zaman gecikmesi ve faz kaymasına neden olmaktadır. Bazen bu bağıl faz kaymalarından dolayı alıcıda oluşan sinyal de toplanma ya da birbirlerinin etkisini kaybederek sinyalin zayıflamasına ya da kaybolmasına neden olmaktadır. Sonuçta, alıcı antenine ulaşan işaret çok sayıda yoldan gelmiş olan işaretlerin toplamından oluşan birleşik bir kuvvettir. Bazı durumlarda alıcının vericiyi gördüğü doğrudan bir yol bulunabilir[14].

Alıcıya ulaşan çok yollu işaretin bileşenleri değişik gecikmelere ve genlik değerlerine sahiptir. Sinüzoidal işaretler kullanılması durumunda değişik gecikmeler değişik fazlara karşı düşer. Alıcıya ulaşan birleşik işaret genlik ve faz anlamında oldukça geniş ve hızlı değişimlere sahip olabilir. Bu olay sönümlenme olarak adlandırılır.

Yayılm ortamında meydana gelebilecek yağmur veya kar yağışı gibi değişiklikler de farklı sönümlenmeye neden olurlar. Bu tip sönümlenmeye uzun süreli sönümlenme denir. Çok yollu iletim sayısal işaretler için semboller arası girişime de neden olarak iletim kalitesini düşürmektedir.

Gezgin radyo kanalları için ortaya çıkan bir etki de doppler etkisidir. Doppler frekans yayılması, taşıyıcı frekansının kaymasına ve işaret bant genişliğinin yayılmasına neden olur. Tüm bu bozucu etkiler, kullanılan modülasyon tekniğinin hata başarımını, AWGN Gauss gürültülü kanaldakine göre oldukça azaltabilir.

2.3.1.3. Çok Yönlü Sönümlenme Modeli

Sönümlenmeli bir çok yollu kanal çeşitli parametrelerle belirlenebilir. Bunlardan bazıları gecikme yayılımı, uyumluluk band genişliği, doppler yayılması ve uyumluluk süresidir[14]. Bant-geçiren bir sinyali,

$$s_b = \text{Re} \left\{ s(t) e^{j 2 \pi f_c t} \right\} \quad (2.65)$$

şeklinde tanımlayalım. Burada $s(t)$ alt geçiren sinyal f_c 'de taşıyıcı frekansıdır. Kanalın N tane yoldan oluştuğunu kabul edelim. Bu durumda, alıcıda oluşan bant-geçiren işaret denklem (2.66)'daki gibi modellenmiş olur.

Doppler kaymasından oluşan sonuç, denklem (2.66)'da, $\tau_k(t)$ gecikmesinden oluşan faz kayması ise denklem (2.67)'de belirtilen şekilde tanımlanır.

$$r_b = \text{Re} \left\{ r(t) e^{j2\pi f_c t} \right\} \quad (2.66)$$

$$r(t) = \sum \rho_k(t) e^{j\theta_k(t)} s(t - \tau_k(t)) + \eta(t) \quad (2.67)$$

Burada; $\rho_k(t)$: k_{inci} yolun zayıflama katsayısı, $\tau_k(t)$: k_{inci} yoldaki gecikme , f_k : k_{inci} yolun “doppler” kayması olarak kullanılmıştır.

Herhangi bir k_{inci} yoldaki faz kayması ise denklem (2.68)’te belirtildiği şekilde ifade edilir.

$$\theta_k(t) = 2\pi f_k t - 2\pi(f_c + f_k)\tau_k(t) \quad (2.68)$$

2.3.1.4. Gecikme Saçılması

Çok yollu kanal üzerinden alıcıya ulaşan işaretin gücü belli bir zaman aralığına yayılmıştır. i . işaret bileşeninin alıcıya ilk ulaşan işaret bileşenine göre gecikme farkına ek gecikme denir ve τ_i ile gösterilir. τ olurlu değerleri τ_i ler olan bir rastlantı değişkeni olarak tanımlanırsa, ortalaması τ^- ortalama ek gecikme olarak adlandırılır. Benzer şekilde varyansının karekökü, σ_T ek gecikmenin standart sapmasını gösterir.

Alınan işaretin enerjisinin en yüksek değerinin X dB aşağısına düştüğü en yüksek zaman gecikmesine ise ek gecikme yayılımı denir. Tipik σ_T değerleri açık alan radyo iletimi için mikro saniyeler seviyesindeyken, kapalı alan kanallarında nano saniyeler ile gösterilebilir. Gecikme saçılması τ_d farklı yollar içinde meydana gelen en geniş gecikmeyi göstermektedir. B_d kanalının uygun bant genişliği, gecikme saçılmasının tersine eşittir. Bu değer ($B_d 1/\tau_d$), ilgili seçici kanalın sinyal bant genişliğinin ölçülmesinde kullanılmaktadır.

B_s ‘nin yayınlanan sinyalin band genişliği olarak kabul edilmişti. Şayet $B_s \ll B_d$ ise kanal transfer fonksiyonu sabit olarak aynı tip gecikmelerde farklı frekans elemanı olarak alınır. Bu kanal, frekansta düzgün seçilemeyen sönümleme olarak adlandırılır. Şayet $B_s \sim B_d$ ya da $B_s > B_d$ ($\tau_d \sim T_s$: sinyal süresi) ise, çeşitli gecikmeli kanalların karışması ve sembol kaybı oluşur. Bu kanal ise frekans seçici (frequency-selective) kanal olarak adlandırılır.

2.3.1.5. Uyumluluk Bant Genişliği

Uyumluluk bant genişliği B_c tüm işaretlerin neredeyse aynı kazanç ve doğrusal fazla geçebildikleri (düzgün) bant genişliğine denilmektedir. Bu band içerisindeki işaretlerin genlik

değerleri oldukça güçlü bir ilişkiye sahiptir. Öte yandan, aralarındaki fark uyumluluk band genişliği B_c 'den büyük olan iki sinüzoidal işaret kanaldan geçerken oldukça değişik bir şekilde etkilenirler. B_c 'nin bir değerini elde edebilmek için iki işaret arasındaki zarf ilişki katsayısı hesaplanır. Uyumluluk band genişliği denklem (2.69)'de olduğu gibi kabul edilir.

$$B_c = \frac{1}{2\pi\sigma_\tau} \quad (2.69)$$

2.3.1.6. Uyumluluk Süresi

Uyumluluk band genişliğine benzer bir şekilde, uyumluluk süresi de bozulmanın fark edilebileceği zaman aralığına denir. Uyumluluk süresi ile doppler yayılması arasındaki ilişki denklem (2.70)'deki gibi verilebilir:

$$T_c \approx \frac{9}{16\pi f_m} = \frac{9}{16\pi B_D} \quad (2.70)$$

Kolaylık açısından bu ilişki denklem (2.71)'de belirtilen şekilde verebiliriz.

$$T_c \approx \frac{1}{B_D} \quad (2.71)$$

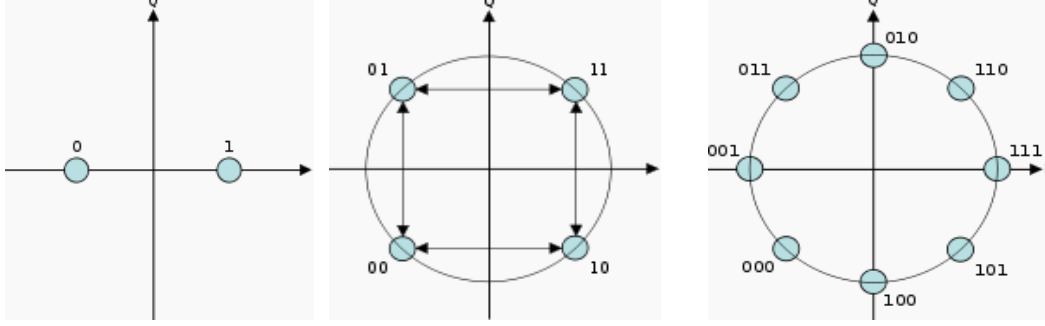
Sayısal iletimde denklem (2.72)'de görüldüğü gibi, genelde bu iki değer geometrik ortalaması kullanılır:

$$T_c = \sqrt{\frac{9}{16\pi B_D^2}} = \frac{0.423}{B_D} \quad (2.72)$$

2.3.1.7. PSK Yapıları

BPSK, QPSK (Quaternary Phase Shift Keying-4PSK) ve 8-PSK türü açılı modülasyonu, iki/dört/sekiz ayrı değer kullanıldığı faz modülasyon uygulamalarıdır. Semboller arasındaki uzaklıkları eşit yapmak için 2π faz değişim bölgesinin PSK seviyesi kadar eşit parçaya bölünmesi gerekmektedir. Dolayısıyla semboller arası açıklık açısal olarak ifade edilebilir. Örneğin QPSK 'da semboller arası açıklık $\pi/2$ olur. Sembollerin açıları bakımından, $k\pi/2, k=0,1,2,3$ ya da $(2k+1)\pi/4, k=0,1,2,3$ durumunda en uygun simetrik sembol konumlarıdır. Ayrı açılı ve genlik değeriyle sınırlı sayısal modülasyon yöntemlerinde,

bu ayrıık deęerlerin bir referans taşıyıcı fazörüne göre konumlarını gösteren şekillere işaret kümesi diyagramları (signal constellation diagram) denir. Şekil 2.22’de BPSK, QPSK ve 8PSK modülasyonlarına ait işaret kümesi diyagramları (sembol konumları) gösterilmiştir.



Şekil 2.22: BPSK, QPSK ve 8-PSK sembol konumları

Sembollerin üzerine yerleştirildikleri çember, sabit genlikli PM taşıyıcı fazörünün geometrik yeridir. Sayısal faz modülasyonunda, bu fazörün belli ayrıık deęerler almasına izin verilir.

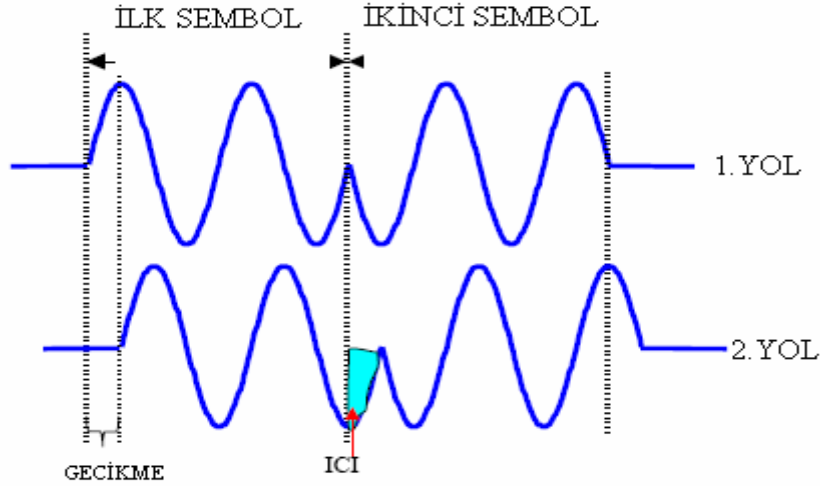
Genelde, n-PSK ile gösterebileceğimiz ve işaret kümesinde n sembol içeren bir ayrıık faz modülasyon yönteminde, her sembol ile $\log_2 n$ bit elde edilebilir. QPSK yönteminde, ayrıık konumlu dört sembol bulunduğu için 2 bit/sembol, 8-PSK ile 3 bit/sembol ve 16-PSK ile 4 bit/sembol iletilebilir. Kanal bant genişliği, saniyede iletilen sembol adedi ile ilgili olduğundan, verilen sabit bir kanal bant genişliği içerisinde daha fazla bilgi iletilemek için PSK seviyesi yükseltilebilir. Ancak sayısal haberleşmede bu mümkün değildir. Çünkü her ortamda bir gürültü mutlaka olacaktır. Sayısal yöntemlerde ise istenen koşul, ayrıık faz deęerlerinin mümkün olduğunca birbirinden uzak olmasıdır. Aksi halde gürültü nedeniyle hatalı verinin oluşmasına sebep olunacaktır.

Hangi seviyedeki PSK modülasyonunun kullanılacağına karar verebilmek için iki ayrıık sembol arasındaki uzaklığın yarısının gürültüden fazla olması güvenilir bir kriter olarak kabul edilebilir. Bu nedenle, belli bir kanal genişliğine sahip olunan durumlarda keyfi olarak bit/sembol oranları yani PSK seviyesi yükseltilerek veri iletim hızının artırılması mümkün değildir. Elde edilebilecek veri iletim hızı SNR deęeri tarafından sınırlanır. Belli bir gürültü seviyesi için yüksek veri iletim hızı sağlanabilmesi için verici gücü artırılarak, taşıyıcı

fazörünün çapının büyümesi nedeniyle semboller arası mesafe açılıp gürültünü etkisi azaltılabilir. Ayrıca hatalı veri iletimini daha da azaltmak için Sayısal haberleşme tekniklerinde genellikle Semboller Arası Girişim etkisini giderecek yöntemler kullanılmalıdır[15].

2.3.1.8. Semboller ve Taşıyıcılar Arası Girişim

Sayısal haberleşmede en ciddi iletim problemi Şekil 2.23'te görülebileceği gibi, sembollerin sinüzoidal olmaması durumunda oluşan semboller arası girişimdir (*Inter Symbol Interference- ISI*). Semboller arası girişimi en az değere indirebilecek süzgeç, Nyquist süzgecidir. Bu süzgecin en önemli özelliği frekans cevabının %50 zayıflama noktasına (f_0) göre tek simetrik bir karakter göstermesidir. Bu tür süzgeçlerin zaman domeninde impuls cevabı $T_s = 1/(2f_0)$ periyodunun katlarında sıfırdan geçer eğer bu süzgeç sembol süresi T_s olan bir darbe dizisine uygulanırsa, semboller alıcıda sembol merkezinde ve T_s aralıklarla örnekleme yapıldığında, bir örnek değerine daha önceki sembollerin bu t anında katkısı olmayacaktır. Yani semboller arası girişim olmayacaktır. Sayısal bir haberleşme sisteminde, semboller arası süre T_s kabul edilirse kullanılması gereken süzgecin $f_0 = 1/(2T_s)$ olacak şekilde tasarlanması halinde ISI etkisi önlenmiş olacaktır. [14]



Şekil 2.23: Semboller arası girişim (ISI)

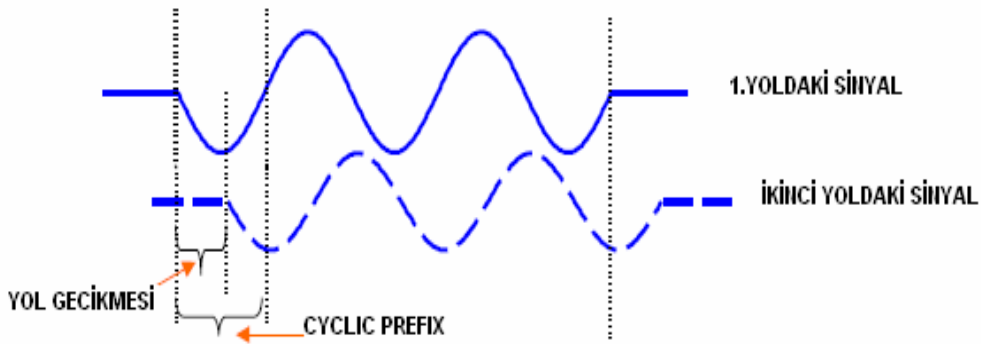
Bir n-PSK yönteminde n sayısı belirlendikten sonra semboller arasındaki açısal uzaklık ve konumları (Constellation diagram), sabit bir rotasyon açısı dışında belli olur. Belli bir işaret kümesinin herhangi bir açıda rotasyonu teorik olarak gürültü performansını etkilemez. Fakat demodülasyon uygulamasının kolaylığı bakımından I ve Q eksenlerine göre ayrı ayrı simetri

gösteren küme yapıları her zaman tercih edilir. Belli bir işaret kümesinin sembollerine, iletilecek sayısal bilginin sembollere nasıl atanacağı farklı bir konudur. Ancak PSK modülasyonlarında bu atama belirlenmiştir ve komşu semboller arasında yalnızca bir bitin değişik olması sağlanmıştır. Farklı modülasyon yöntemlerinde oluşan hatanın giderilebilmesi için demodülasyon sonrasında hata düzeltme kodlama yöntemlerine ek olarak farklı teknikler kullanılmaktadır. Şekil 2.24’de görüldüğü gibi, “cyclic prefix” parametresinin modülasyon esnasında kullanılması semboller arası girişimi önleyecek bir tekniktir.



Şekil 2.24: “Cyclic Prefix” ile bir sembolün periyodik genişlemesi

Şekil 2.25’te görüleceği gibi çoklu yol gecikme süresinin “Cyclic Prefix”ten kısa olması durumunda kablosuz iletimdeki ISI ve ICI problemleri bertaraf edilebilmektedir.



Şekil 2.25: Çoklu yol gecikme süresinin ‘CP’ den kısa olması

2.3.2. OFDM

OFDM sistemler; çoklu yol bozulmasına karşı dayanıklı, iletim kanalının frekans spektrumunun ufak parçalara ayrılarak frekansla değişmez hale getirilmesi, semboller arası girişimi engellenmesi, daha basit kanal eşitleyicilere ihtiyaç duyulması, farklı hata düzeltme kodları kullanarak daha güvenli iletim sağlanabilmesi amacıyla tercih edilen bir yapıdır.

2.3.2.1. OFDM'in Tercih Nedenleri

Sayısal haberleşme teknikleri, değişken koşullara sahip bir ortamda sınırlı sembolleri ilettikleri için sorun yaratan etkiler (iç/dış gürültüler, iletim kanalı yapısı,yansımalar v.b.) nedeniyle oluşan bozulmaların hata düzeltmeli kodlama yöntemleri (error correction) ile düzeltilebilmesine, düzeltilemeyenlerin olumsuz etkilerinin mümkün olduğunca azaltılmasına (error concealment) imkan verirler. Bu koşullar altında, sayısal kodlanmış veriyi iletmek için tamamen yeni teknikler geliştirmek zorunlu hale gelmiş ve tüm veri iletim ihtiyaçlarının karşılanması için kuvvetli bir sayısal modülasyon yönteminin verimli bir temel band kodlamasıyla birlikte kullanılması ile çözüm sağlanabilmiştir. [15]

Analog iletim yöntemlerinde, vericiden uzaklaştıkça işaret zayıflamakta, dolayısıyla işaret gürültü oranı bozulmakta, alıcıda ise bu nedenle aynı oranda kalitesiz sinyal gözlenebilmektedir. Sinyal iletim ortamının zayıflama karakteristiği sürekli olduğundan, vericiden uzaklaşıldıkça alıcıdaki sinyal kalitesinde de zayıflama, sürekli ve uzaklık ile orantılı olarak değişir. Sayısal iletimde de aynı sınırlama sözkonusu olmakla beraber, farklı teknikler kullanılarak gerçekleştirilebilen, sinyal-gürültü oranındaki iyileştirmeler nedeniyle daha iyi sonuçlar elde edilebilmektedir. Verici ve alıcı taraflarındaki sinyal/gürültü değerleri neredeyse aynı seviyelere getirilebilmekte ancak belli bir eşik değerinden sonra iletim tamamen kaybolabilmektedir (abrupt degradation). Sayısal sistemlerde fiziksel engeller (yağmur, sis,rüzgar v.b.) nedeniyle ortaya çıkan bu sorunun en az etkiye indirgenebilmesi için katmanlı (multilayer) modülasyon yapıları kullanılmaktadır. Ancak verinin katmanlara ayrılması karmaşık yapılar gerektirmekte ve sabit bant genişliği için veri hızını yavaşlatmaktadır.

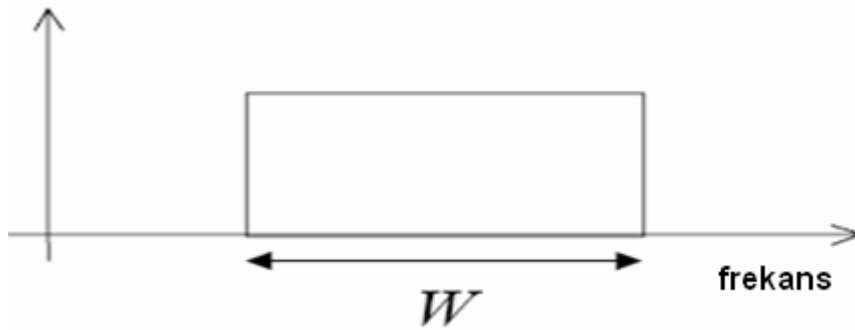
Sayısal yöntemlerde, kaynak işaret ile kodlanmış sayısal işaret arasında zaman domeninde sabit bir ilişki olmasına gerek yoktur. Bu nedenle kullanılan sayısal yöntemlere göre aynı işaretin iletiminde çok farklı parametreler kullanılabilir. Sayısal iletim için tek taşıyıcılı (klasik) ya da çok taşıyıcılı bir modülasyon kullanan sistemler ya da her taşıyıcısı

klasik bir şekilde modüle edilmiş fakat çok taşıyıcılı olan sistemler kullanılabilir. Tek taşıyıcı sistemlerde taşıyıcı başına düşen veri hızı çok yüksek olmaktadır. Bu nedenle kompleks yöntemler kullanılmaktadır. Çok taşıyıcılı sistemlerde ise, toplam veri hızı, taşıyıcı adedine bölüldüğü için, taşıyıcı bazında veri hızı düşmekte dolayısıyla paralel bilgisayar veri iletimine benzetilebileceği gibi daha basit ve kompleks olmayan modülasyon yöntemleri kullanılabilir. Günümüzde, çok taşıyıcılı yöntemlerin başında OFDM gelmektedir.[15]

Teknolojide sağlanan hızlı gelişmeler sonucunda geniş kitlelere sayısal yayın mümkün ve ekonomik hale gelmiş sayısal iletim yöntemleri yaygın şekilde uygulanmaya başlanmıştır. Farklı sayısal iletim kodlama ve modülasyon yöntemleri olmakla beraber, son yıllarda dikey frekans çoğullamalı modülasyon yöntemi (OFDM: Orthogonal Frequency Division Multiplex) [16] önem kazanmış ve tek frekans ağlarına imkan tanınması nedeniyle gün geçtikçe tercih edilen bir teknik durumuna gelmiştir. Bu teknik ile birlikte genelde Faz Kaymalı Anahtarlama (PSK-Phase Shift Keying) modülasyon kullanılmaktadır.

2.3.2.2. Seri/Paralel Veri İletim Sistemleri

Geleneksel bir seri veri iletim sisteminde her bir işaret frekans spektrumu, kullanılabilir band genişliğinin tümünü kaplamasına izin verilecek şekilde, sıralı olarak arka arkaya gönderilir. Şekil 2.26'de görülen tek taşıyıcılı sistemlerde, bir sönümleme ya da bozucu etki tüm iletim hattının etkileyerek iletimi bozmaktadır.



Şekil 2.26: Tek taşıyıcılı sistem

Paralel bir veri iletim sistemi ise, seri sistemlerde görülen bir çok sorunun negatif etkisini azaltmaktadır. Paralel bir sistem, bir anda birden fazla sıralı veri dizilerinin aynı anda iletildiği ve buna bağlı olarak, seçilen herhangi bir anda, birden fazla verinin iletildiği bir sistemdir. Böyle bir sistemde her bir veri sembolünün frekans spektrumu kullanılabilir band

genişliğinin sadece bir bölümünü kaplar. Şekil 2.27’de görülen çok-taşıyıcı sistem yapısındanda görüleceği gibi sadece alt-taşıyıcıların çok az bir bölümü etkilenmekte ve daha sonra hata düzeltici kodlama yöntemleri ile bu çok az hata da kolaylıkla ortadan kaldırılabilmektedir.



Şekil 2.27: Çok taşıyıcı sistem

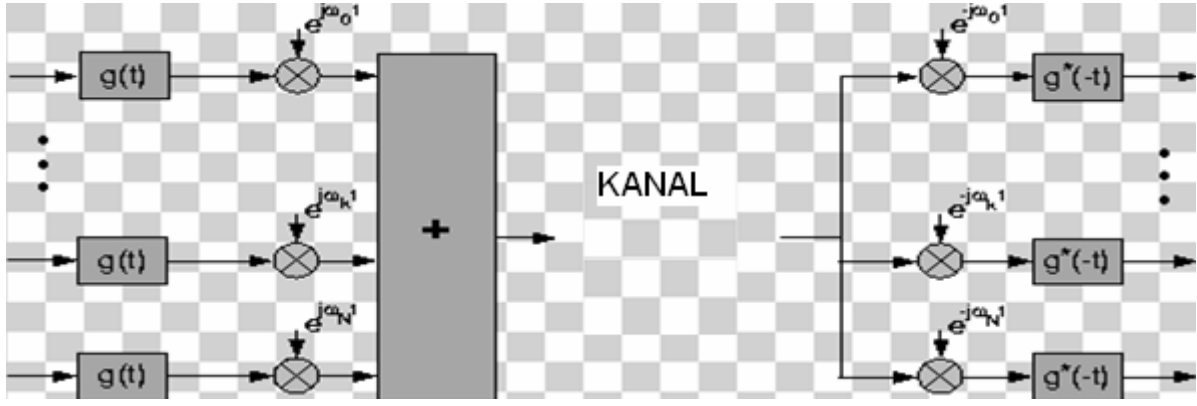
Paralel sistem yaklaşımı, frekansa bağlı zayıflamanın simgeler üzerinde dağıtılması ile avantajlı duruma gelmektedir. Bu durum, zayıflama nedeniyle ortaya çıkan hataların etkili bir biçimde dağılmasını sağlar. Böylece sıralı verilerin tamamen zarar görmesi yerine çoğu sembol daha az zarar görür. Bu durum ileri yönde hata düzeltme (FEC:Forward Error Correction) olmaksızın bile, sembollerin büyük bir kısmının başarılı bir şekilde geri elde edilebilmesini sağlar.

2.3.2.3. OFDM Tekniği

OFDM, teorik temelleri 1950'lere dayanan çok taşıyıcı bir modülasyon tekniğidir [17]. Sayısal işaret işleme (DSP:Digital Signal Processing) ve geniş çaplı tümdevre tasarımı (VLSI:Very Large Scale Integration) tekniklerindeki son gelişmeler, OFDM'in elektronik pazarında geniş kullanım alanları bulmasına imkan tanımıştır.

1960'larda OFDM birçok askeri sistemlerde, yüksek hızlı modemlerde, dijital mobil haberleşmede ve yüksek yoğunluklu kayıtlarda kullanılmıştır. 1990'larda ise mobil radyo FM kanalları üzerinden geniş bantlı data haberleşmesinde, yüksek bit oranlı dijital alt-taşıyıcı hatlarda (HDSL) asimetrik dijital alt-taşıyıcı hatlarda (ADSL), çok-yüksek-hızlı dijital alt-taşıyıcı hatlarda (VLSI), dijital ses yayıncılığında (DAB) ve yüksek tanımlamalı televizyon (HDT) yayıncılığında kullanılmıştır. [18]

Şekil 2.28'da genel bir yapısı görülen OFDM tekniğindeki diğer bir uygulama ise ANSI (American National Standards Institute) tarafından sayısal sıkıştırılmış video işaretlerini telefon hatları üzerinden göndermek amacıyla seçilmiş olan asimetrik sayısal abone hattı ADSL (Asymmetric Digital Subscriber Line) teknolojisidir[19].



Şekil 2.28 Genel OFDM yapısı

OFDM tek bir data dizisinin belli bir sayıdaki alt-taşıyıcılarla iletildiği çok-taşıyıcılı iletimin özel bir halidir. Burada OFDM bir modülasyon çeşidi gibi görülebildiği gibi aynı zamanda çoklayıcı tekniğidir. OFDM'in en önemli kullanıma sebeplerinden birisi, frekans seçici sönmülemeli kanallara ve dar-band bozulmasına karşı sağlamlığıdır. Ayrıca Pilot Sembol eklenmesi ile hata düzeltme olanağı artmaktadır.

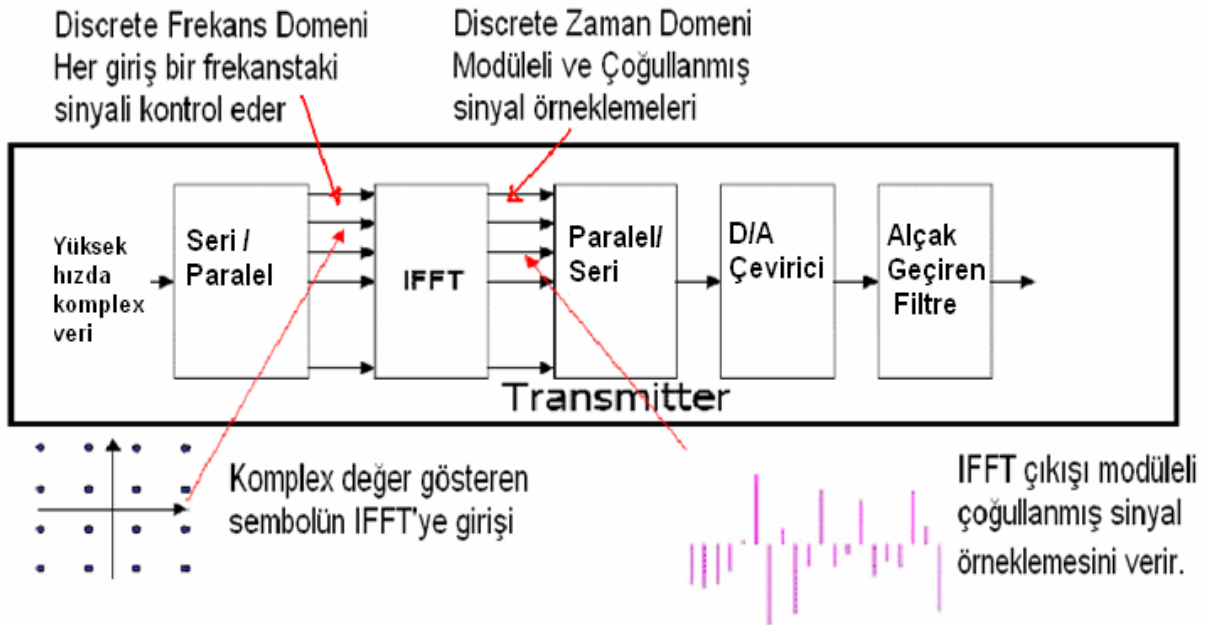
Geçmiş 1940'lı yıllara dayanan uçan araçlardaki VHF tabanlı haberleşme sistemi artan yoğun hava trafiği ile birlikte, dar-bant haberleşme sisteminin günümüzde doyum noktasına ulaşacağı değerlendirilmektedir. Dolayısıyla, OFDM gibi çok taşıyıcılı haberleşme sistemleri, yeni nesil haberleşme teknolojileri için büyük aday konumuna gelmektedir[20][21].

Kanal band genişliğinin tamamının alt bandlara bölünmesinden dolayı, her bir alt bandın frekans yanıtı göreceli olarak düzdür. Alt kanallar, asıl kanal genişliğinin belirli bir bölümünü kapladığı için, kanal eşitleme işlemi seri sistemlere oranla daha kolaydır. Basit bir dengeleme algoritması, her alt kanaldaki karesel ortalama bozulmasını minimum hale getirebilir, farksal kodlamanın kullanılması ise eşitlemenin tamamen ortadan kalkmasını sağlayabilir.

Paralel bir veri sisteminde, tüm işaret frekans bandı, frekansları birbiriyle çakışmayan, N tane alt kanala ayrılır, bu işlemden sonra N alt kanal frekans çoğullamalı olarak modüle edilir. Toplam frekans bandı N adet üst üste gelmeyen frekans alt-kanalına bölünür. Her alt-kanal ayrı sembol için modüle edilir ve daha sonra N alt-kanalın frekansları çoğullanır. Kanallar arası bozulmayı yok etmek için frekans spektrumunun üst üste gelmesini önlemek iyi gibi görünmektedir. Fakat bu yöntem, kanalın frekans spektrumunun verimli bir biçimde

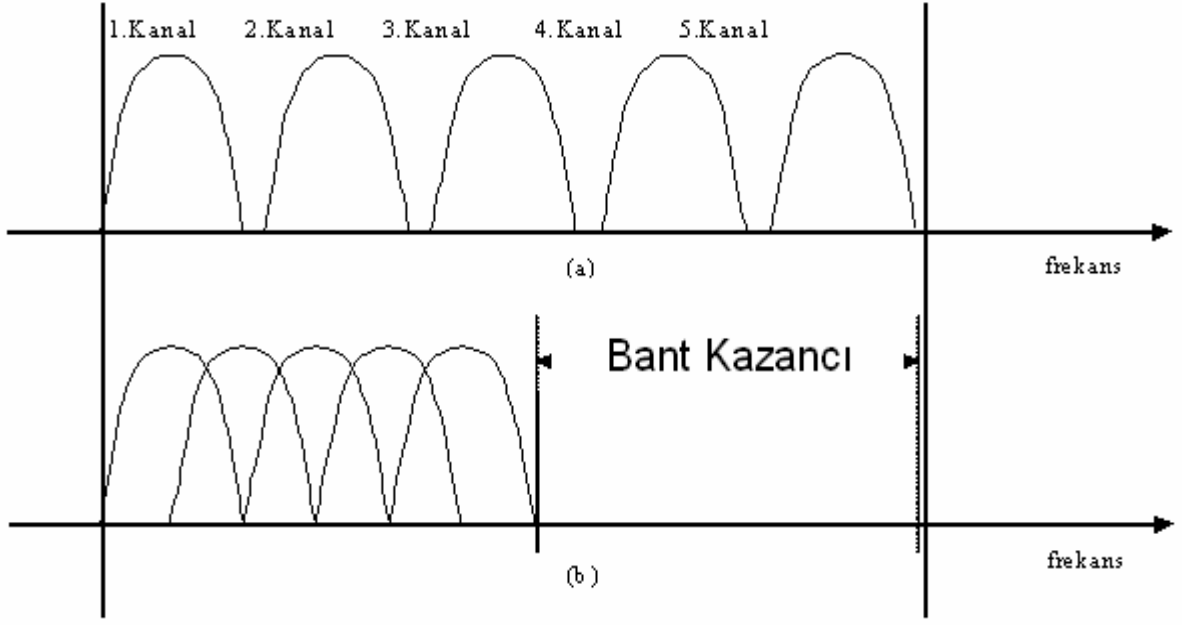
kullanılmamasına neden olmaktadır. Bu verimsizliği ortadan kaldırmak için önerilen yöntem, paralel data ve alt-kanalların üst üste bindiği, ancak kesişmelerin yüksek olduğu frekans bölümlenmeli çoğullamadır (FDM).

OFDM tekniğinde, kompleks yapı değiştirilerek işlemlerin daha kolay/çabuk yapılabilmesi için, Şekil 2.29'da görüleceği gibi matematiksel olarak IFFT uygulanıp frekans domeninden zaman domenine geçilmektedir. Burada, paralel veri demetleri IFFT girişleri olarak kullanılmakta, IFFT modülünün çıkışı sinyal örneklerinin toplamı olmakta ve modülasyon ile çoğullama tek adımda gerçekleştirilmektedir. Filtreleme ve A/D çevrimi de temel bant sinyalinin oluşması ile sonuçlanmaktadır[22].



Şekil 2.29: Modüleli sinyale IFFT uygulanması

Geleneksel üst üste gelmeyen çok-taşıyıcılı sistemle frekansların üst üste geldiği modülasyon tekniği arasındaki fark Şekil 2.30'da görülmektedir. Frekansları üst üste gelen yöntemin kullanılması band genişliğinde %50'lik bir kazançta neden olmaktadır. Üst üste gelme yöntemini gerçekleştirmek için modüle edilmiş farklı taşıyıcılar arasında bir dikliğe gerek vardır, bu da kanallar arasındaki kesişmenin (crosstalk) ihmal edilmesi anlamına gelir.



Şekil 2.30: (a) geleneksel çok-taşıyıcı (b) dik çok-taşıyıcı yapısı

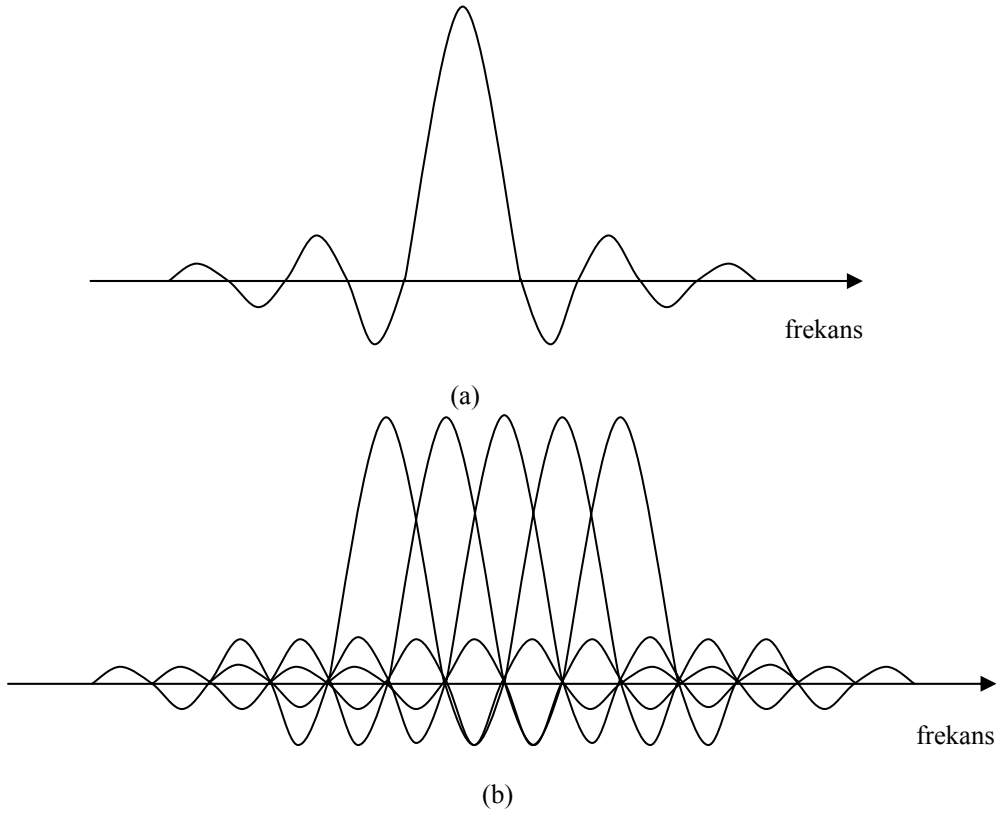
FDM tekniğinin diklik prensibine göre çalışma halini OFDM olarak adlandırılabilir. Bir sinyalin matematiksel olarak diklik (orthogonal) prensibiyle kullanılması denklem (2.73)'de olduğu şekilde gerçekleştirilebilir.

$$\int_0^T \sin \frac{2\pi kt}{T} \sin \frac{-2\pi kt}{T} dt = 0, \quad k \neq 1 \quad (2.73)$$

Diklik kelimesi sistemdeki taşıyıcılar arasında tam bir matematiksel ilişki olduğunu gösterir. Normal bir FDM sisteminde, bir çok taşıyıcı, sinyallerin geleneksel filtre ve demodülatör yoluyla alınmasından dolayı boşa ayrılırlar. Bu tür alıcılarda taşıyıcılar arasına gard süresi eklenmesi gerektiği için frekans domeninde verimliliğin düşmesine sebep olunur. Bir OFDM sinyalinde, her taşıyıcının kenar bandlarını üst üste getirecek ve sinyali hala bir bozulma olmadan alabilecek şekilde taşıyıcıları düzenlemek mümkündür. Bunu yapabilmek için taşıyıcıların bir birine dik olması gerekir. Alıcı bir demodülatörler kümesidir, her taşıyıcıyı taşıyıcı frekansları ile çarpıp bir sembol periyodu süresince entegre eder. Farklı taşıyıcı frekansları ile çarpılanlar diklikten dolayı integralleri alındıktan sonra sıfır olurlar.

Bir çok arařtırmacı yüksek verimlilikli dik frekanslı çok taşıyıcı sistemler üzerine çalışmışlardır. 1971 yılında Weinstein ve Ebert ayrık Fourier dönüşümünü (DFT) paralel data iletiminde modülasyon ve demodülasyonun bir parçası olarak kullanmışlardır.

Şekil 2.31(a)'da tek bir alt-kanalın frekans spektrumunu göstermektedir. Frekansları çoğullanmış OFDM sinyallerinin frekans spektrumu ise Şekil 2.31 (b)'de gösterilmektedir. Görüldüğü gibi, tam orta frekanslarda herhangi bir kesişme (crosstalk) olmadığı görülür. Bu yüzden eğer alıcıda DFT kullanılır ve her alt-taşıyıcının orta frekansı ile korelasyonu hesaplanırsa, iletilen datayı crosstalk olmadan elde edebiliriz. Frekans bölümleneli çoğullamanın içerdiği alt-taşıyıcı osilatörlerin kümelerini ve coherent demodülatörlerini ortadan kaldırmak için DFT'nin iyi bir uygulaması olan hızlı Fourier dönüşümü (FFT) tamamen dijital bir donanım olarak kullanılabilir.[14]



Şekil 2.31: (a) bir OFDM alt-kanalının spektrumu (b) bir OFDM sinyalinin spektrumu

Günümüzde çok-geniş-ölçülü bütünleşme (VLSI) teknolojisindeki gelişmeler yüksek hızlı ve geniş boyutlu FFT çiplerinin ticari olarak alınmasını sağlamıştır. Hem alıcı hem de vericide FFT tekniğini kullanmak, işlem sayısını N^2 den $M \log N$ 'e indirmiştir.

2.3.2.4. Alt Taşıyıcıların Üretilmesi

Bir OFDM sinyali PSK ya da QAM kullanılarak modüle edilmiş alt-taşıyıcıların toplamından oluşur. d_i kompleks bir sembol, N_s alt-taşıyıcı sayısı, T sembol süresi, ve f_c taşıyıcı frekansı olmak üzere $t = t_s$ anında başlayan bir OFDM sinyali denklem (2.74)'te olduğu gibi yazılabilir.

$$s(t) = \text{Re} \left\{ \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s/2} \exp(j2\pi(f_c - \frac{i+0.5}{T})(t-t_s)) \right\}, \quad t_s \leq t \leq t_s + T \quad (2.74)$$

$$s(t) = 0, \quad t < t_s \wedge t > t_s + T$$

Bu formülün yerine temel band haberleşmesinde genellikle denklem (2.75) kullanılır. Bu gösterimde gerçek ve sanal kısımlar yatay ve dikey bileşenleri gösterir. Yatay ve dikey bileşenler OFDM sinyalini elde etmek için belirlenen taşıyıcı frekansın cosinüs ve sinüs'leridir.

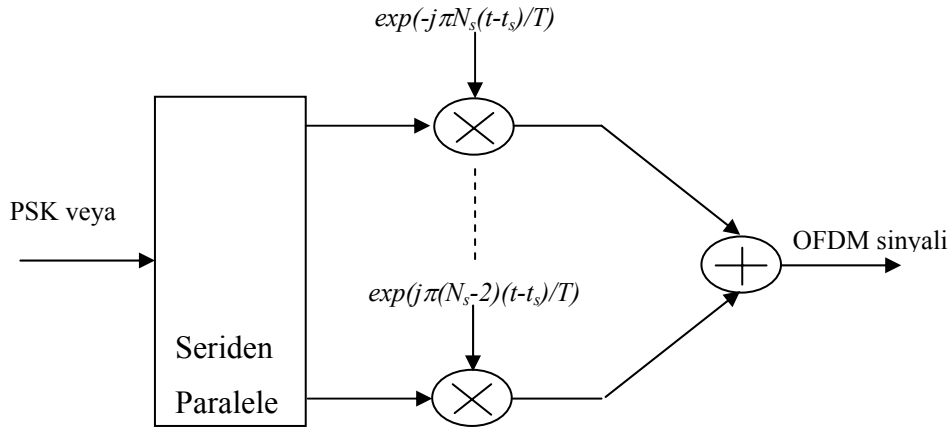
$$s(t) = \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s/2} \exp(j2\pi \frac{i}{T}(t-t_s)), \quad t_s \leq t \leq t_s + T \quad (2.75)$$

$$s(t) = 0, \quad t < t_s \wedge t > t_s + T$$

Şekil 2.32'de OFDM modülatörünün işlevi blok diyagram olarak görülmektedir. Şekil 2.33 ise üç alt-taşıyıcılı bir OFDM sinyalini göstermektedir. Burada tüm alt-taşıyıcılar aynı faz ve genliğe sahipler, fakat pratikte genlik ve faz farklı olacak şekilde modüle edilebilir. Her alt-taşıyıcının T aralığında tam sayıda periyodu vardır, ve en yakınındaki alt-taşıyıcıdan farklıdır. Bu özellik alt-taşıyıcılar arasındaki diklik için önemlidir. Eğer, denklem (2.75), j . alt-taşıyıcı j/T frekansı ile demodüle edilir ve sonra T süresi boyunca entegre edilirse sonuç denklem (2.76)'da olduğu gibi olur.

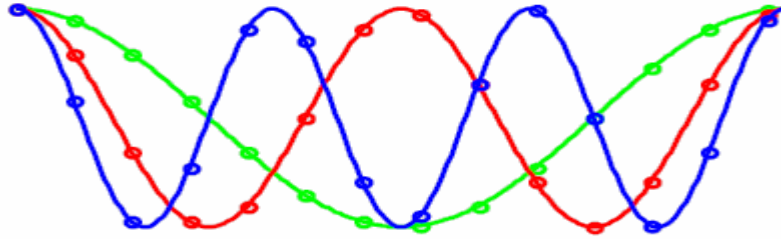
$$\begin{aligned}
& \int_{t_s}^{t_s+T} \exp(-j2\pi \frac{j}{T}(t-t_s)) \sum_{i=-\frac{N_s}{2}}^{\frac{N_s-1}{2}} d_{i+N_s/2} \exp(j2\pi \frac{i}{T}(t-t_s)) dt \\
&= \sum_{i=-\frac{N_s}{2}}^{\frac{N_s-1}{2}} d_{i+N_s/2} \int_{t_s}^{t_s+T} \exp(j2\pi \frac{i-j}{T}(t-t_s)) dt = d_{j+N_s/2} T
\end{aligned} \tag{2.76}$$

Denkleme bakıldığında karmaşık taşıyıcı T süresi boyunca entegre edilir. Demodüle edilmiş j . alt-taşıyıcı için bu integrasyonda istenen sonucu $d_{j+N_s/2}$ verir. Diğer tüm alt-taşıyıcılar için entegrasyonun sonucu sıfırdır. Çünkü frekans farkı $(i-j)/T$, T integral süresi içinde tam sayıda periyod yaratır, bu nedenle integral sonucu her zaman sıfırdır.



Şekil 2.32: OFDM modülatörü

Şekil 2.33'de verilen üç alt taşıyıcılı bir sistemde bütün alt taşıyıcılar ve aynı faz ve genliğe sahiptir. Ancak gerçek uygulamada bütün alt taşıyıcılar için genlikler ve fazlar farklı modüle edilirler.



Şekil 2.33: Üç alt-taşıyıcılı OFDM sinyali

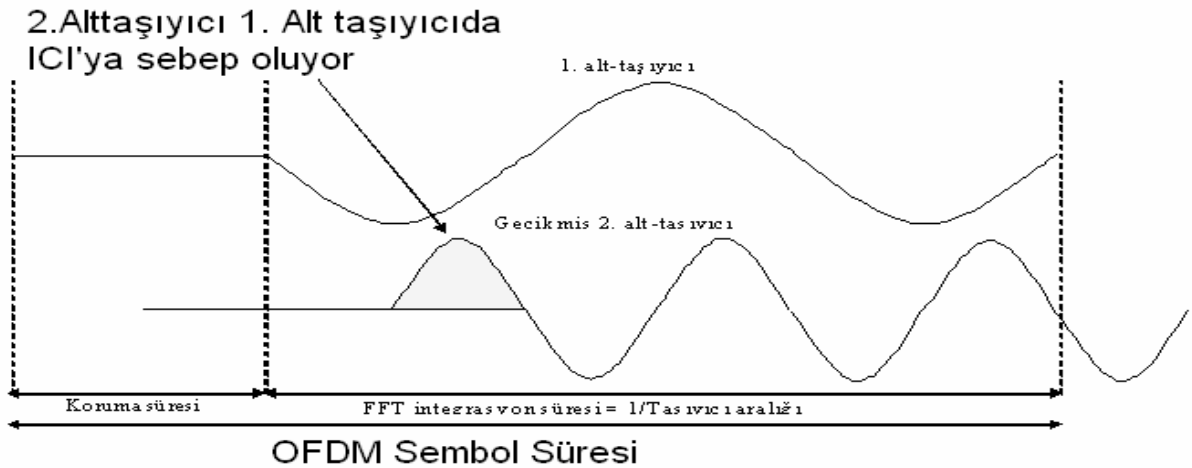
OFDM alt-taşıyıcıları arasındaki diklik başka bir şekilde de gösterilebilir. Her OFDM sembolü T -saniye süresinde sıfır olmayan alt-taşıyıcılardan oluşur. Bundan dolayı, bir sembolün spektrumu, alt-taşıyıcı frekansında bulunan Dirac sinyaller grubu ile T -saniye boyunca bir, diğer zamanlarda sıfır olan kare dalgaının spektrumu ile konvolüsyonudur. Kare dalgaının genlik spektrumu, $1/T$ ile çarpımını tam yapan tüm f ler için sıfır veren $\text{sinc}(\pi fT)$ dir. Her alt-taşıyıcı spektrumunun maksimumunda, diğer alt-taşıyıcıların spektrumları sıfır olur. Temel olarak semboller arası bozucu olmayan bir dalga şekli için Nyquist kriteri yerine getirilmektedir. [23]

Denklem (2.76)'de verilen karmaşık temel band OFDM sinyali, N_s giriş sembolünün ters Fourier dönüşümünden başka bir şey değildir. t zamanını örnek sayısı n ile değiştirerek ayrık zamandaki eşdeğer ifade denklem (2.77)'daki gibi elde edilir bu da ters ayrık Fourier dönüşümüdür (IDFT). Bu dönüşüm, pratikte ters hızlı Fourier dönüşümü (IFFT) olarak kullanılır. IFFT, IDFT'den oldukça daha az işlem içerir.

$$s(n) = \sum_{i=0}^{N_s-1} d_i \exp(j2\pi \frac{in}{N}) \quad (2.77)$$

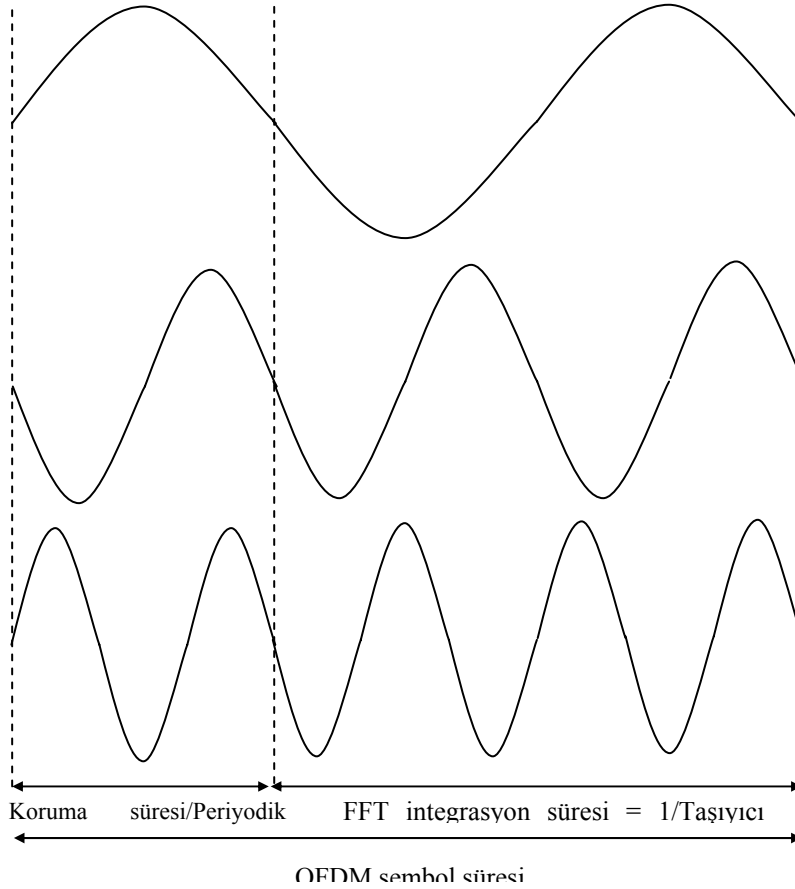
2.3.2.5. Koruma Süresi (Guard Time) ve Periyodik İlaveler

OFDM yönteminin kullanılmasının en önemli nedeni çok-yollu gecikme yayılımıdır. Giriş dizisini N_s adet alt-taşıyıcıya bölerek sembol süresi N_s defa küçültülmüş olur bu da sembol süresine göre çok-yollu gecikme yayılımını azaltır. Semboller-arası bozulmayı (ISI) tamamen ortadan kaldırmak için OFDM sembolüne gard süresi eklenir.



Şekil 2.34: Koruma zamanı olmayan bir sinyalin çok-yollu ortamlardaki etkisi

Gard süresi beklenen en uzun gecikme yayılımından daha büyük seçilir. Böylece bir önceki sembolün bir sonraki sembole karışması önlenir. Klasik yöntemlerde gard süresinde hiçbir sinyal eklenmemekteydi. Bu durumda taşıyıcılar-arası bozulma (ICI) ortaya çıkabilmekteydi. ICI alt-taşıyıcılar arasında oluşan ve hiç bir şekilde dik olmayan çakışmadır (crosstalk). Bu etki Şekil 2.34'de gösterilmiştir. Burada 1. alt-taşıyıcı ve gecikmiş 2. alt-taşıyıcı gösterilmektedir. OFDM alıcısı 1. alt-taşıyıcıyı demodüle etmeye çalıştığında, 2. alt-taşıyıcının gecikmiş olmasından dolayı bir bozulmayla karşılaşacaktır. Aynı sebepten dolayı 1. alt-taşıyıcı 2. aly-taşıyıcıda bir çakışmaya sebebiyet verecektir.

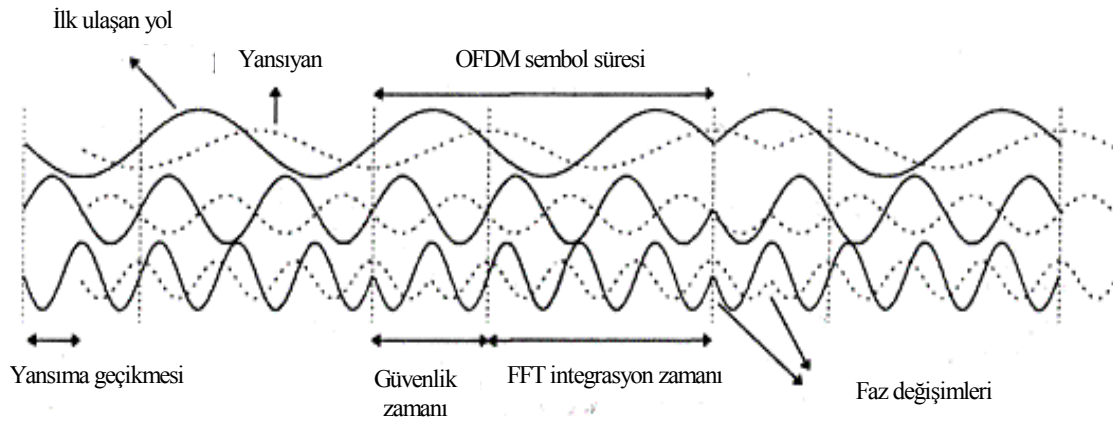


Şekil 2.35: Periyodik örnekli OFDM sembolü

Taşıyıcılar arası bozulmayı ortadan kaldırmak için gard süresi boyunca OFDM sembolü Şekil 2.35'te gösterildiği gibi periyodik olarak uzatılır. Gecikme süresi gard süresinden küçük olduğu sürece, OFDM sembolünün gecikmiş benzerinin FFT zaman aralığındaki tekrarları her zaman tam sayılı olacaktır. Sonuç olarak, çok-yollu sinyallerde gard süresinden küçük olan gecikmeler, taşıyıcılar arası bozulmaya neden olmazlar.

Çok yollu ortamlar, OFDM sistemlerini etkiler . İki yollu kanalda alınan işaretlerin görüldüğü Şekil 2.36'da, üç sembol süresinde üç ayrı alt taşıyıcı görülmektedir. Pratikte bir OFDM alıcısı sadece bunların toplamını görür fakat bu şekilde ayrı göstererek çok yolun etkisi daha açık olarak görülmektedir. Burada, çoklu yol gecikmesi koruma zamanından daha küçüktür. Bu durumda FFT aralığında faz değişimi olmayacaktır. Böylece bir OFDM alıcısı fazı biraz kaymış saf bir sinüs dalga göreceğinden alt taşıyıcılar arasındaki diklik bozulmayacaktır.

Çoklu yol gecikmesinin koruma zamanından daha büyük olduğu durumlarda taşıyıcılar arasında diklik kaybolur. Bu durumda, faz geçişleri taşıyıcıdaki FFT aralıklarına yayılacaktır. İlk yoldaki sinüs dalgalarıyla diğer gecikmeli yoldaki faz modüleli dalganın toplamı artık dik olan saf sinüs dalgası vermeyecek ve belirli seviyede bir girişimle oluşacaktır.



Şekil 2.36: İki yollu üç alt taşıyıcılı OFDM sinyal örneği.

2.3.2.6. Pencereleme

Alt taşıyıcıların diğer bir sinyalle birleştiği yerlerde modülasyondan doğan keskin geçişler oluşacağından band dışı spektrum, *sinc* fonksiyonuna göre daha yavaş bir şekilde azalır. Fakat, yüksek sayılı alt-taşıyıcılar için frekans spektrumu ise daha hızlı azalır. Spektrumu daha hızlı aşağı çekebilmek için her OFDM sembole uygulanan Periyodik ek ve pencereleme yapısı Şekil 2.37'de görülmektedir. Pencereleme OFDM sembolünün başlangıç ve bitiş noktalarında yumuşak bir şekilde sifira gitmesini sağlayan bir avantajdır. En çok kullanılan pencere tipi yükseltilmiş cosinüs (*raised cosine*) penceresidir ve denklem (2.78)'de olduğu şekliyle tanımlanır.

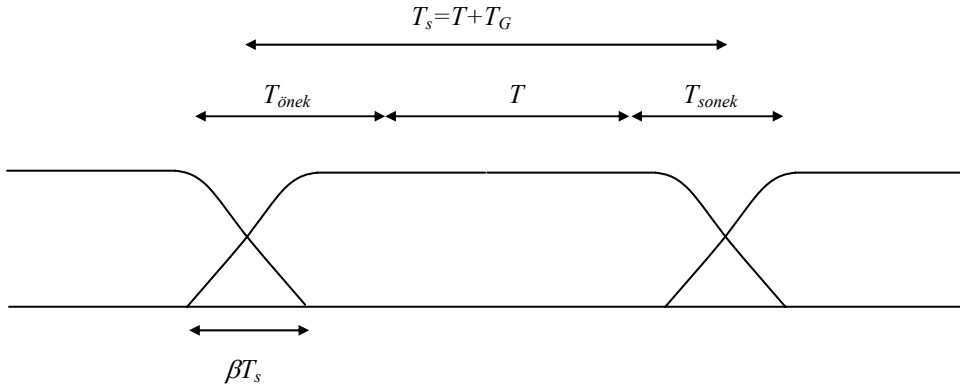
$$w(t) = \begin{cases} 0,5 + 0,5 \cos(\pi + t\pi / (\beta T_s)) & 0 \leq t \leq \beta T_s \\ 1,0 & \beta T_s \leq t \leq T_s \\ 0,5 + 0,5 \cos((t - t_s)\pi / (\beta T_s)) & T_s \leq t \leq (1 + \beta)T_s \end{cases} \quad (2.78)$$

Burada T_s sembol süresidir ve bu süre toplam sembol süresinden kısadır. OFDM sinyalinin zamanlama yapısı Şekil 2.37'deki gibidir. $t = t_s = kT_s$ zamanında başlayan OFDM sembolünün denklemi aşağıdaki gibi tanımlanır.

$$s_k(t) = \text{Re} \left\{ w(t - t_s) \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s(k+1/2)} \exp(j2\pi(f_c - \frac{i+0,5}{T})(t - t_s - T_{\text{önek}})) \right\}, \quad (2.79)$$

$$t_s \leq t \leq t_s + T_s(1 + \beta)$$

$$s_k(t) = 0, \quad t < t_s \wedge t > t_s + T_s(1 + \beta)$$



Şekil 2.37: OFDM periyodik eki ve pencereleme

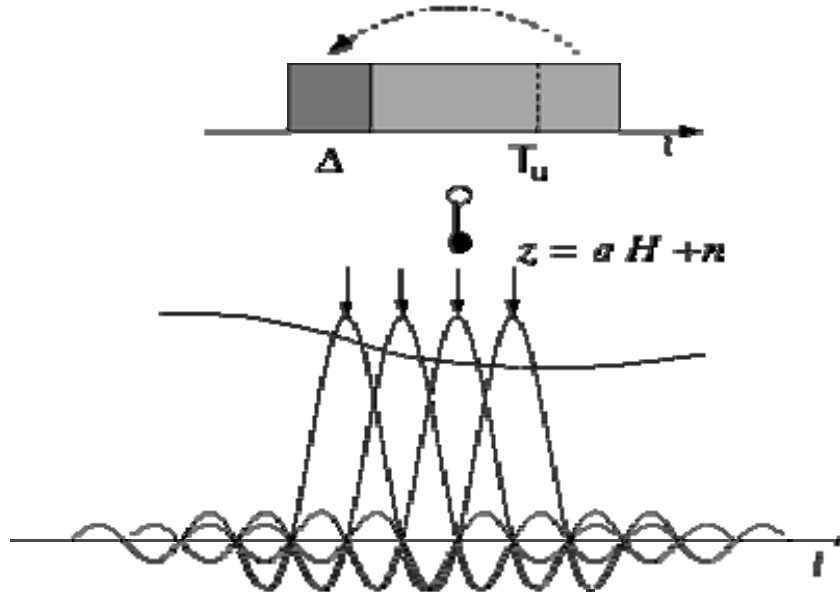
Pratikte OFDM sinyali üretilirken; önce N_s adet giriş değerine sıfırlar eklenerek IFFT'si alınacak N adet örnek elde edilir. Daha sonra IFFT çıktısının $T_{\text{önek}}$ örneği OFDM sembolünün başına, T_{sonek} örneği de sonuna eklenir. Elde edilen OFDM sembolü band dışı gücü çok daha hızlı bir şekilde düşürebilmek için raised cosine penceresi $w(t)$ ile çarpılır. β raised cosine penceresinin düşme (*roll-off*) faktörü olmak üzere, βT_s lik bir örtüşme bölgesi ile bir önceki OFDM sembolünün arkasına eklenir.

Bant dışı spektrumu azaltmak için, gerektiği takdirde pencereleme yerine geleneksel filtreleme teknikleri de kullanılabilir. Bir OFDM sinyalini pencere ile çarpılması pencere fonksiyonunun spektrumu ile alt taşıyıcı frekanslardaki dürtü kümelerini birbiriyle konvolüsyonu sonucunu

doğurur. Filtreleme uygulandığında zaman bölgesinde konvolüsyon alınır ve OFDM spektrumu filtrenin frekans yanıtı ile çarpılır. Dijital filtreleme tekniklerinin çerçeveleme yapmaktan daha karmaşık olduğu göz ardı edilmemelidir. Çünkü bir dijital filtreleme tekniği ile her bir örnekte birkaç çarpım gerekecekken pencereleme ile her sembolde birkaç çarpım gerektirecektir.

2.3.2.7. OFDM ve Diklik (Orthogonality)

OFDM’de iletilmek üzere hazırlanan alt band darbeleri dikdörtgen olarak seçilir. Bu da darbe şekillendirme ve modülasyonun basit IFFT (Inverse Fast Fourier Transform) uygulaması ile gerçekleşmesi için avantaj sağlamaktadır. Fourier Transform teorisine göre Şekil 2.38’de dikdörtgen darbe şekli $\sin(x)/x$ tipinde olan alt taşıyıcının spektrumuna karşılık gelecektir.



Şekil 2.38: OFDM ve diklik (orthogonality) prensibi

Matematiksel olarak diklik prensibi denklem (2.80)’da belirtildiği şekilde ifade edilir.

$$\int_0^T \sin \frac{2\pi kt}{T} \sin \frac{-2\pi kt}{T} dt = 0, \quad k \neq 1 \quad (2.80)$$

OFDM sistemlerde alt taşıyıcıların spektrumları ayrılmamakta ve üstüste gelmektedir. Buna rağmen taşıyıcıların birbirinden ayrıştırılabilmesine diklik (orthogonality) özelliği olanak sağlamaktadır. Modülasyon için IFFT kullanılması ile alt taşıyıcılar arasındaki boşluk seçilebilmektedir. Şekil 2.38’de görüldüğü gibi sinyal (ok ile gösterilmiştir) olarak kabul

edilen noktalar dışındaki işaretler yok sayılmaktadır. Dikliğın korunabilmesi için: Alıcı ve verici senkronize (aynı modülasyon frekansı ve aynı iletim zamanlaması) olması, kullanılan malzemelerin yüksek kalitede olması ve çok yollu kanal olmaması gerekmektedir. Çok yollu kanal ortamının mevcut olması durumunda OFDM tekniđi ile basit bir şekilde elimine edilmesi mümkündür.

2.3.2.8. OFDM Parametrelerinin Seçimi

Farklı OFDM parametrelerinin seçimi, bazen birbirine ters düşen parametreler arasında bir etkileşimdir. Parametre seçiminde üç ana faktör bulunmaktadır. Bant genişliđi, bit hızı ve gecikme yayılımıdır. Gecikme yayılımı doğrudan koruma zamanını gösterir. Kural olarak koruma zamanı yaklaşık gecikme yayılımından bir kaç kat fazla olmalıdır.

Geçikme yayılmasına bađlı olarak, koruma zamanı belirlendikten, sonra sembol süresi ayarlanabilir. Koruma zamanı tarafından meydana getirilen güç kaybını en aza indirmek için koruma zamanından çok daha büyük sembol zamanına sahip olunması gerekir. Bu deđer rastgele seçilemez. Daha büyük bir sembol zamanı daha küçük alt taşıyıcı aralıklı, daha fazla alt taşıyıcı, daha büyük gerçekleştirme karmaşıklığı faz gürültüsüne karşı daha fazla duyarlılık ve frekans kaydırması ve artan ortalama güç oranıyla birlikte deđerlendirilmelidir. Pratik bir tasarımda sembol zamanının koruma zamanının en az beş katı olması gerekebilir. Bu koşulun sağlanması da yaklaşık olarak 1 dB'lik güç kaybına sebep olabilir.

Koruma zamanı ve sembol zamanı belirlendikten sonra oluşturulacak olan alt taşıyıcıların sayısı, ihtiyaç duyulan bit hızının alt taşıyıcı başına bit hızına bölünmesiyle belirlenebilir. Alt taşıyıcı başına bit hızı, modülasyon tipi kodlama hızı ve sembol hızıyla tanımlanır.

Örnek olarak, Bit hızı:20 Mbps, Tolere edilebilen gerileme yayılımı :200 ns ve Bant genişliđi:< 16 MHz olarak belirlenen yapıya uygun bir sistem tasarlamak istediđimizde;

200 nsn'lik gecikme yayılımına karşılık koruma zamanı için 800 ns'lik bir deđer güvenilir olacaktır. OFDM sembol zamanını koruma zamanının 6 katı (4,8 μ s) seçerek koruma zaman kaybı 1dB'den daha küçük yapılır. Alt taşıyıcı aralıđı bu durumda 4,8-0,8=4 μ s'nin tersidir.bu da 250 kHz'i verir. İhtiyaç duyulan alt taşıyıcıların sayısını belirlemek için ihtiyaç duyulan bit hızının oranına ve OFDM sembol hızına bakabiliriz. 20 Mbps elde edebilmek için, her bir OFDM sembolü 96 bitlik bilgi (96/4,8 μ s = 20 Mbps) taşımak zorundadır. Alt taşıyıcı başına,

sembol başına 1,5 bit veren QPSK'yı hız $\frac{3}{4}$ 'te kodlama yapıldığı kabul edilirse. Bu durumda, sembol başına 96 bite ulaşmak için 64 alt taşıyıcıya gerek vardır. Böylece, 64 alt taşıyıcı ile $64 \times 250 \text{ kHz} = 16 \text{ MHz}$ 'lik bir bant genişliği elde edilir.

Seçilen parametleri etkileyebilecek diğer bir faktör FFT/IFFT aralığı içinde ve sembol aralığı içinde örneklerin tam sayısına ihtiyaç duyulmasıdır. Verilen örnekte alt taşıyıcılar arasında dikliği korumak için FFT/IFFT aralığında tam olarak 64 örneğe ihtiyaç duyulacağından örnekleme hızı $64/4 \mu\text{s} = 16 \text{ MHz}$ yapılarak elde edilebilir. Buna rağmen bu özel örnekleme hızı için $4,8 \mu\text{s}$ 'lik sembol aralıklı tam sayılı örnekler yoktur. Bu problemin tek çözümü tam sayı sınırlamasını karşılamak için parametrelerden bir tanesinin az bir değerde değiştirilmesidir. Örneğin, sembol başına düşen örnek sayısı 78 yapılabilir, bu da $78/4,8 \mu\text{s} = 16,25 \text{ MHz}$ 'lik bir örnekleme hızı verir. Bu durumda FFT aralığı $64/16,25 \text{ MHz} = 3,9385 \mu\text{s}$ olur. Bu durumda koruma zamanı ve alt taşıyıcı aralığı ile birlikte FFT aralığı $4 \mu\text{s}$ durumundan biraz büyük olacaktır.

2.3.2.9. OFDM Standartları

Kablosuz ağların sahip oldukları temel karakteristikler nedeniyle bilinen kablolu ağlara göre farklılıklar göstermektedir. Bazı kullanıcı ülkeler kendilerine has özel gereksinimler belirlemişlerdir. Kablosuz haberleşmede yaygın olarak IEEE 802.11 serisi standartlar referans olarak kabul edilmektedir. Bu standartların genel amacı, hareket eden bir araca monte edilmiş ya da elde taşınan otomatik bir makine gibi ekipmanlar arasındaki kablosuz bağlantının kurulabilmesidir. Bu standartlar ayrıca yerel ağ haberleşmesinin sağlanabilmesi için bir ya da birden fazla frekans bandına erişim kurallarını da belirlemektedir. Kablosuz bir haberleşme türü olan OFDM sistemleri ile ilgili standartlar;

- IEEE 802.11 standardı, uyumlu cihazların gezgin (mobil) ağlarda çalışması için gereken servis ve fonksiyonlarını ve Asenkron servislerin desteklenebilmesi için, Orta Düzeyde Erişim Kontrolü (MAC-Medium Access Control) prosedürlerinin tanımlanması,
- IEEE 802.11 MAC tarafından kontrol edilen çeşitli fiziksel (PHY) sinyal teknikleri ve arabirim fonksiyonlarının tanımlanması,
- Kablosuz gezgin ağlardaki IEEE 802.11 uyumlu cihazların çalıştırılmasına olanak sağlanması,

- Gezgin ağlarda IEEE 802.11 uyumlu cihazlar arasında gerçekleşen veri akışının ve transfer edilen kullanıcı bilgilerinin korunmasını sağlayacak gereksinim ve prosedürlerin tanımlanması,
- 802.11a/g: Temel OFDM standardlarının tanımlanması,
- 802.11n: MIMO,STC, OFDM ile ilgili standartlar olmakla beraber. Diğer uluslararası standartlar;
- 802.16 : Tek taşıyıcı, direkt görme (LOS),Lisanslı frekans bantlarının tanımlanması,
- 802.16-2004 : Çoklu Taşıyıcı, direkt görmeme (NLOS) , lisanslı/Lisanssız bantlar,
- 802.16e: Mobil kullanıcılar, hücreler arasında iletimin tanımlanması,
- 802.15.3a: Çok bantlı OFDM fonksiyonlarını tanımlanması amacıyla kullanılmaktadır.

2.4. OFDM HABERLEŞME KANALLARINDA ÇOK SEVİYELİ LDPC KODLAMA

Tez çalışmasında öncelikle LDPC Kodlama öğrenilmiş ve benzetimi yapılmıştır. Daha sonra OFDM haberleşme sistemlerine ait benzetim gerçekleştirilmiştir. Öncelikle LDPC kodlamalı OFDM yapısı incelenmiş, daha sonra Çok Seviyeli LDPC Kodlamalı OFDM sisteminin benzetimi yapılmıştır.

2.4.1. LDPC Kodlamalı OFDM

Değişken koşullara sahip bir ortamda sınırlı sembolleri ilettikleri için sorun yaratan etkiler nedeniyle, hata düzeltmeli kodlama yöntemleri ile olumsuz etkilerinin mümkün olduğunca azaltılmasına imkan veren sayısal haberleşme teknikleri ile iletim kalitesi gün geçtikçe artmaktadır.

Sinyal iletim ortamının zayıflama karakteristiği sürekli olduğundan, Sayısal iletimde vericiden uzaklaştıkça alıcıdaki sinyal kalitesinde de zayıflama, sürekli ve uzaklık ile orantılı olarak değişme sözkonusu olmakla beraber, farklı teknikler kullanılarak gerçekleştirilebilen, sinyal-gürültü oranındaki iyileştirmeler nedeniyle daha iyi sonuçlar elde edilebilmektedir. Kullanılan sayısal yöntemlere göre aynı işaretin iletiminde çok farklı parametreler kullanılabilir.

Sayısal iletim için tek taşıyıcılı (klasik) ya da çok taşıyıcılı bir modülasyon kullanan sistemler kullanılabilir. Tek taşıyıcı sistemlerde taşıyıcı başına düşen veri hızı çok yüksek olmaktadır.

Bu nedenle kompleks yöntemler kullanılmaktadır. Çok taşıyıcılı sistemlerde ise, toplam veri hızı, taşıyıcı adedine bölüldüğü için, taşıyıcı bazında veri hızı düşmekte dolayısıyla paralel bilgisayar veri iletimine benzetilebileceği gibi daha basit ve kompleks olmayan modülasyon yöntemleri kullanılabilir.

Günümüzde, çok taşıyıcılı yöntemlerin başında OFDM gelmektedir. OFDM ile kodlanmış ortamlarda, hata düzeltme algoritmaları kullanılarak daha başarılı sonuçlar alınabileceği varsayıldığı için buna yönelik çalışma yapılmıştır. Bu tez çalışmasında nihai olarak, OFDM (Orthogonal Frequency Division Multiplex) [16] haberleşme ortamlarında, çok seviyeli (Multi Level) Düşük Yoğunluklu Parite Kontrol (LDPC: Low Density Parity Check) [3],[4] kodlama yapılan bir haberleşme sisteminin benzetimi yapılarak sistem performansı ölçülmeye çalışılmıştır.

Semboller arası girişimin (ISI) çok fazla olduğu, çok yollu sönmülemeli ortamlarda ve yüksek veri hızında OFDM cazip bir tekniktir. Bu teknik, Reed-Solomon kodlama, Turbo kodlama ve yakın zamanda uygulanan LDPC gibi bir çok hata düzeltme algoritmaları ile birlikte kullanılabilir. Yapılan çalışmalarda, AWGN kanallarında yapılan LDPC-COFDM performansının Turbo kodlamalı OFDM'den daha kötü olduğu ancak frekans seçmeli sönmülemeli kanallarda ise tam tersi bir durum olduğu ispatlanmıştır [24].

IEEE 802.16a standartlarındaki bir sistemde, LDPC Kodlamayla birlikte OFDM modülasyon kullanılması durumunda sonucun, Turbo kodlamalı OFDM sisteme göre daha iyi sonuçlar verdiği gözlenmiştir [25]. Ayrıca, LDPC Kodlamalı OFDM tekniğinin, fiber optik sistemlerde de çok verimli sonuçlar verdiği ispatlanmıştır [26].

Güçlü hata düzeltme kodlama algoritmaları kullanıldığında, sönmülenmiş alt bantlar yerine diğer alt bantlar ile birlikte karşı tarafa aktarılan ortalama güç oranının referans olarak alınmasıyla, sönmülemeye uğramış alt bantların yanılıcı etkisi yok edilebilmektedir[27].

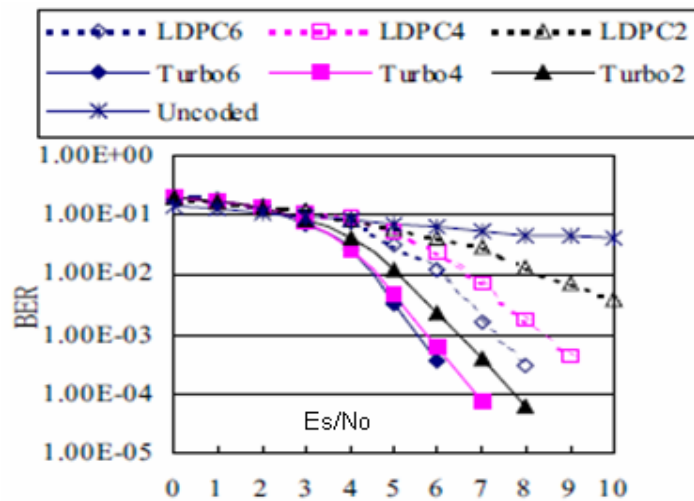
Yakın zamanda, LDPC Kodlama tekniği kodlama teorisi alanında büyük dikkat çekmiştir. Bu çalışmada 'BER' oranını OFDM kanallarda daha iyi düzeye getirebilmek için Çok Seviyeli LDPC kodlamanın etkisi incelenecektir. OFDM tabanlı LDPC kodlamalı haberleşmede sönmülemeli kanallarda iletim kalitesinde daha iyi sonuçlar alınabilecektir. OFDM sistemleri örneğin yerel ağlar (WLAN) gibi gelişen kablosuz yeni nesil (4G) haberleşme sistemlerinde

anahtar rol almaktadır. Hata düzeltme amacıyla kullanılan; katlamalı kodlar, turbo kodlama ya da LDPC kodlama teknikleri OFDM'in kalitesini etkileyecek tekniklerdir [28].

Çok yollu sönümlenmeli kanallarda OFDM'in bazı alt taşıyıcıları, derin sönümlemeden dolayı kaybolabilmektedir. Çoğunluktaki diğer alt taşıyıcıların hatasız olarak iletimi sağlanmasına rağmen, düşük seviyeli alt taşıyıcıların etkisiyle Bit-Hata Oranı (Bit Error Rate-BER) yüksek çıkabilmektedir. Bu etkinin azaltılabilmesi için hata düzeltme kodlaması gerekmektedir. Bu kodlama teknikleri, OFDM tabanlı haberleşme sistemlerinde (Wimax, DAB,DVB ADSL, Hiper LAN) Shannon Limitlerine yakın kazanç sağlayabilir. Güçlü hata düzeltme kodlama algoritmaları kullanıldığında, sönümlenmiş alt bantlar yerine diğer alt bantlar ile birlikte karşı tarafa aktarılan ortalama güç oranının referans olarak alınmasıyla, sönümlemeye uğramış alt bantların yanıtıcı etkisi yok edilebilmektedir [27].

Yakın zamanda hata düzeltme algoritmalarında sağlanan gelişmeler ile tekrarlamalı (iteratif) kodlayıcıların, sistem performansını "Shannon" limitlerine yaklaştırdığı belirlenmiştir. "Shannon" limitlerine yakın sistem performansı elde edilebilmesi için geniş blok boyutlarında (10.000-40.000 bit) kodlama yapılması gerekmektedir. Bu ise kablosuz haberleşmede büyük gecikmelerden dolayı pratik bir uygulama değildir. Kablosuz sistemlerde iyi kalitede iletimin sağlanabilmesi için küçük blok uzunluklarında iyi kodlama tekniği kullanılmalıdır [29][30].

Şekil 2.39'da görüldüğü gibi, LDPC kodlamalı ya da Turbo kodlamalı OFDM yapısına ilişkin yapılan çalışmalarda, sistem performanslarına yönelik bulgular bulunmaktadır [31].

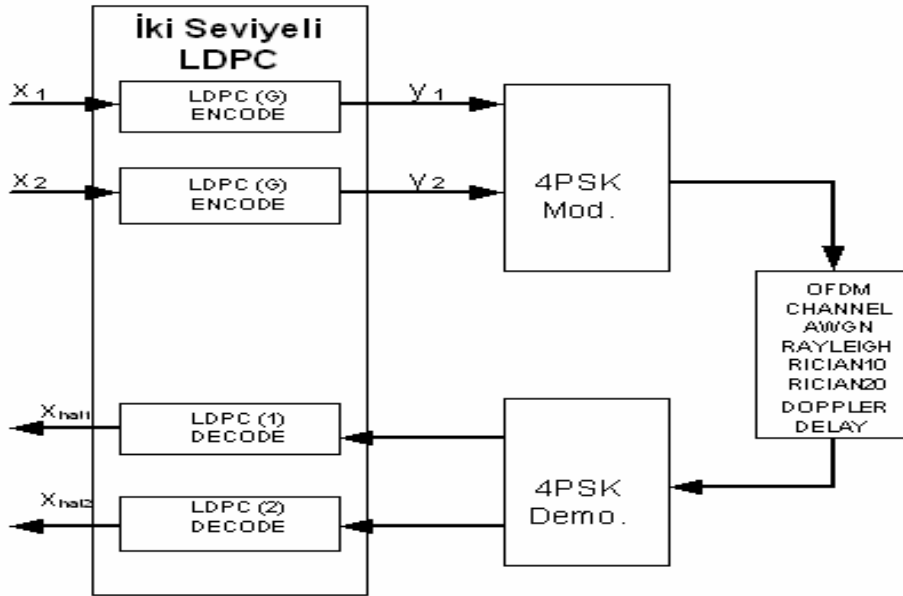


Şekil 2.39: Farklı kodlamalı OFDM sistem performansları

Farklı kodlamalı OFDM sistemlerinin performanslarına yönelik yapılan çalışmada, Turbo kodlama ve LDPC Kodlama ile entegre edilmiş OFDM sistemlerin performansları karşılaştırılmıştır. İdeal bit hata oranlarına (BER) yaklaşık 7-9 desibel aralığında ulaşılmaktadır. Kodlanmamış bir OFDM yapısında ise istenen seviyede başarımlar sağlanamamaktadır.

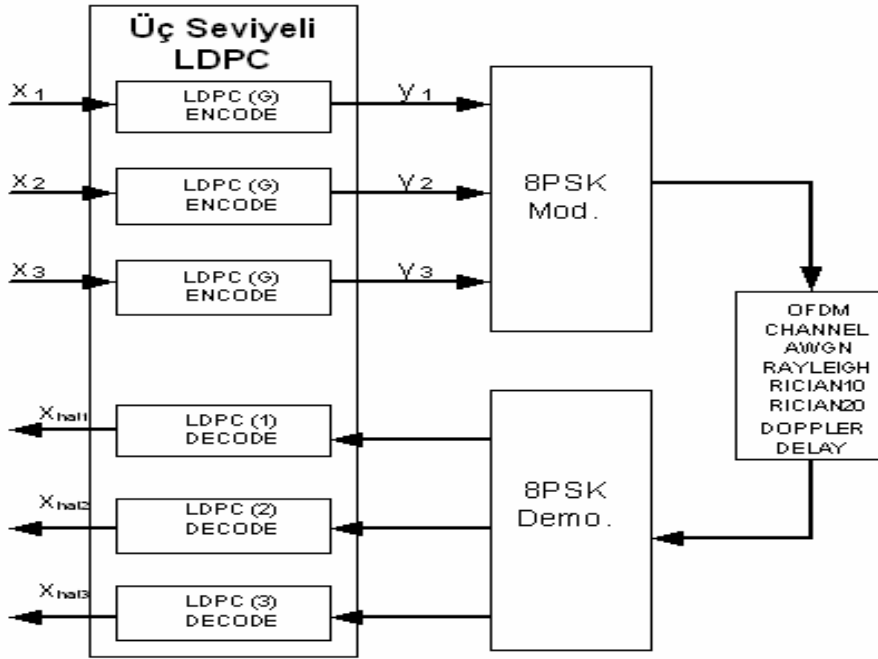
2.4.2. Çok Seviyeli LDPC Kodlamalı OFDM

Transfer edilmesi gereken verinin artması ve kısıtlı bant genişliğinin haberleşme sistemlerinde yeni gelişmeleri beraberinde getirmektedir. Farklı modülasyon tekniklerinin yanısıra, daha fazla verinin kısıtlı bant aralığında en az hata ile gönderilmesi için verinin paralel kollarda kodlanarak gönderilmesi ile zaman tasarrufu sağlanabilecektir. Bu çalışmada, LDPC kodlama çok seviyeli yapılarak, elde edilen kazanımlar ortaya çıkarılmaya çalışılmıştır. Şekil 2.40' da görülen iki seviyeli LDPC kodlama blok yapısından anlaşılacağı gibi, iki adet LDPC kodlayıcı bloğu kullanılarak elde edilen kodlanmış veri sembole dönüştürülmek amacıyla 4 PSK modülatörden geçirilmektedir.



Şekil 2.40: İki seviyeli LDPC kodlamalı OFDM

Şekil 2.41' de görülen üç seviyeli LDPC kodlama blok yapısından da anlaşılacağı gibi üç adet LDPC kodlayıcı bloğu kullanılarak elde edilen kodlanmış veri sembole dönüştürülmek amacıyla 8 PSK modülatörden geçirilmektedir.



Şekil 2.41: Üç Seviyeli LDPC Kodlamalı OFDM

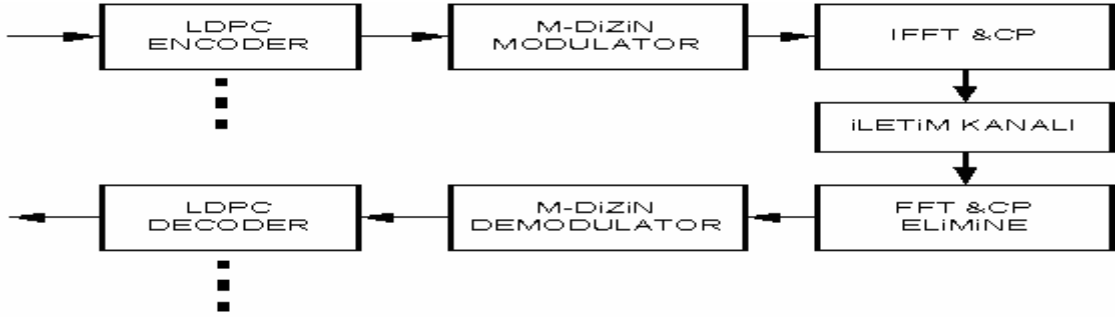
Blok şemalardan da anlaşılacağı gibi Çok seviyeli kodlama, seviyesine göre neredeyse aynı zaman aralığında gönderilebilecek veri miktarı iki ya da üç katına çıkarılabilmektedir.

Hızın arttırılabilmesi büyük bir avantaj sağlamaktadır. Ancak seviye derecesi rastgele olarak arttırılamamaktadır. Çünkü arttırılan her seviye kodlama sonuçlarında eşde edilen hata başarımlarına ciddi etkilere neden olabilecektir.

Hata başarımının istenen seviyede ve veri iletiminin istenen hızda elde edilebilmesi amacıyla kodlama seviyesi belirlenirken, sahip olunan sistemin alt yapısından kaynaklanan gürültü seviyesi büyük öneme sahiptir. Sistemin sahip olduğu gürültü seviyesi, kodlama seviyesini belirlemede en önemli belirleyici parameter olarak kabul edilmesi gerekmektedir.

3. MALZEME VE YÖNTEM

OFDM tabanlı haberleşme sistemlerinde veri transferine ait performansların görülebilmesi için öncelikle Matlab ortamında Çok Seviyeli LDPC kodlama yapılmış akabinde OFDM kodlama ve son olarak ta ikisinin birlikte kullanılmasını amaçlayan benzetim ortamları oluşturulmuş, her bir yöntem için ayrı ayrı kodlamalar yapılmıştır. Benzetim ortamına ait genel yapı ise Şekil 3.1’ de görülmektedir.



Şekil 3.1: Benzetim sistemi blok diyagramı

Yapılan benzetim (Simülasyon) ile sağlıklı sonuçlar alınabilmesi için izlenen aşamalar aşağıda verilmiştir;

- Benzetim ortamının oluşturulmasında kullanılacak parametrelerin belirlenmesi,
- LDPC Encoder/Decoder alt ünitelerinin benzetim programlarının yazılarak, LDPC kodlamanın öğrenilmesi,
- M-Dizin Demodulator/Modulator alt ünitelerinin benzetim programlarının yazılması,
- OFDM haberleşme sisteminin benzetim programının hazırlanması,
- Haberleşme simülasyonunun gerçekleştirileceği farklı iletim ortamlarının benzetim programının hazırlanması,
- LDPC/OFDM benzetim programının iletim ortamlarında farklı ölçüde veri paketleri ile test edilmesi,
- Benzetim programları ile yapılan testler sonucunda elde edilen Bit Hata Oranlarının (BER) değerlendirilmesini kapsamaktadır.

4. BULGULAR

4.1 ÇOK SEVİYELİ LDPC KODLAMA BENZETİMİ

4.1.1. Benzetim Ortamı

Öncelikle benzetim ortamının oluşturulmasında kullanılacak parametreler belirlenmiş ve Matlab programında kodlanmıştır. Benzetim çalışması boyunca, çok seviyeli LDPC (Multilevel-LDPC) uygulaması konusu yoğunluk kazanmıştır. Çok seviyeli LDPC benzetimi gerçekleştirilmiş ve 4PSK/8PSK ile kodlama yapılarak rastgele üretilmiş LDPC verisinin AWGN, Rayleigh, Rician(K=10) ve Rician (K=20) gibi farklı gürültü ortamlarında, Ga (52x200), Gb(150x200) ve SED (1024x1013) üreteç matrisleri ile birlikte ayrı ayrı kullanılarak performans testi gerçekleştirilmiştir.

4.1.2. Programlama Akışı

Aşağıda verilen program akışı yalnızca önceden tanımlanmış bir üreteç matrisi, herhangi bir gürültü ortamının kullanılması ve tanımlanmış PSK yapısı ile elde edilen bir yapıdır. Ga, Gb, SED Üreteç matrislerinin her biri ile AWGN, Rayleigh, Rician 10 ve Rician20 gürültü ortamlarında farklı PSK yapılarında testler yapılmıştır.

Benzetimde ilk adım olarak global değişkenler tanımlanmıştır. Ayrıca bu kısımda hangi üreteç matrisinin kullanılacağı da belirlenmektedir.

```

1 - clear all
2 - global Gb Hb
3 - %load('SED_1024_1013_G.mat');
4 - %load('SED_1024_1013_4_H.mat');
5 - %load('gb.mat');
6 - %load('hb.mat');
7 - load ga;
8 - load ha;
9 - Gb=Ga;
10 - Hb=Ha;
11 - [in,out]=size(Gb);

```


Daha sonra rastgele bitler üretilmiş çıkış bitlerinin nasıl kodlanacağı belirlenmiştir.

```

13 - x1 = (round(rand(1,size(Gb,1)))); % random bits
14 - x2 = (round(rand(1,size(Gb,1)))); % random bits
15 - x3 = (round(rand(1,size(Gb,1)))); % random bits
16
17 - y1 = mod(x1*Gb,2); % coding
18 - y2 = mod(x2*Gb,2); % coding
19 - y3 = mod(x3*Gb,2); % coding
20
21 - bellek_e=0;

```

Hangi seviyede modülasyon yapılacak ise (4/8PSK) ona göre sembol tanımlamaları yapılmıştır.

```

23 - y=[y1;y2;y3];
24 - s0=1;
25 - s1=sqrt(2)/2+i*sqrt(2)/2;
26 - s2=i;
27 - s3=-sqrt(2)/2+i*sqrt(2)/2;
28 - s4=-1;
29 - s5=-sqrt(2)/2-i*sqrt(2)/2;
30 - s6=-i;
31 - s7=sqrt(2)/2-i*sqrt(2)/2;
32
33
34 - yd=bi2de(y,'left-msb');
35
36 - for dongu1=1:length(y)
37 -     if yd(dongu1)==0 r2(dongu1)=s0;
38 -     elseif yd(dongu1)==1 r2(dongu1)=s1;
39 -     elseif yd(dongu1)==2 r2(dongu1)=s2;
40 -     elseif yd(dongu1)==3 r2(dongu1)=s3;
41 -     elseif yd(dongu1)==4 r2(dongu1)=s4;
42 -     elseif yd(dongu1)==5 r2(dongu1)=s5;
43 -     elseif yd(dongu1)==6 r2(dongu1)=s6;
44 -     elseif yd(dongu1)==7 r2(dongu1)=s7;
45 -     end
46 - end

```

Gürültü seviyesi desibel olarak manuel değiştirilebilecek şekilde kodlanmış, standart sapma elde edilmiş ve programdaki iterasyon sayısının manuel olarak belirlenmesi sağlanmıştır.

```

48 - EsN0db=12;
49 - SNRv=10^(EsN0db/10);
50 - rate=in/out;
51 - sigma=1/sqrt(2*SNRv*rate);
52
53
54 - for nframe=1:1000

```

AWGN, Rayleigh ya da Rician(10/20) gürültü ortamlarının benzetimi yapılmıştır. K değeri manuel olarak değiştirilerek istenen gürültü ortamı benzetimi gerçekleştirilmiştir.

```

56      %Rician Channel
57 -    K=1e10; %K=20, K=10, K=1(Rayleigh)
58 -    sdr=sqrt(0.5/(K+1));
59 -    M=sqrt(K/(K+1));
60 -    st1=rand(size(Hb,2),1);st2=rand(size(Hb,2),1);
61 -    A1=sqrt(-2*log(st1)).*cos(2*pi*st2)*sdr;
62 -    A2=M+sqrt(-2*log(st1)).*sin(2*pi*st2)*sdr;
63 -    roi=sqrt(A1.^2+A2.^2);
64
65 -    noise=sigma*randn(1,size(Gb,2))+i*sigma*randn(1,size(Gb,2));
66
67 -    r=r2.*roi' + noise;

```

Öklid uzaklıkları ve olasılıkları ile her bir çıkış bitin sembol konumlarının belirlenmesini sağlayan kodlama, yumuşak karar kod çözme ile gerçekleştirilmiştir.

```

71 -    for dongu2=1:size(Hb,2)
72 -        dis1(dongu2,:)=abs((r(dongu2)-s0).^2);
73 -        dis2(dongu2,:)=abs((r(dongu2)-s1).^2);
74 -        dis3(dongu2,:)=abs((r(dongu2)-s2).^2);
75 -        dis4(dongu2,:)=abs((r(dongu2)-s3).^2);
76 -        dis5(dongu2,:)=abs((r(dongu2)-s4).^2);
77 -        dis6(dongu2,:)=abs((r(dongu2)-s5).^2);
78 -        dis7(dongu2,:)=abs((r(dongu2)-s6).^2);
79 -        dis8(dongu2,:)=abs((r(dongu2)-s7).^2);
107 -    prob_sifir1(dongu2)=1/4*(1/dis1(dongu2)+1/dis2(dongu2)+1/dis3(dongu2)+1/dis4(dongu2));
108 -    prob_bir1(dongu2)=1/4*(1/dis5(dongu2)+1/dis6(dongu2)+1/dis7(dongu2)+1/dis8(dongu2));
109
110 -    prob_sifir2(dongu2)=1/2*(1/dis1(dongu2)+1/dis2(dongu2)+1/dis5(dongu2)+1/dis6(dongu2));
111 -    prob_bir2(dongu2)=1/2*(1/dis3(dongu2)+1/dis4(dongu2)+1/dis7(dongu2)+1/dis8(dongu2));
112
113 -    prob_sifir3(dongu2)=1/2*(1/dis1(dongu2)+1/dis3(dongu2)+1/dis5(dongu2)+1/dis7(dongu2));
114 -    prob_bir3(dongu2)=1/2*(1/dis2(dongu2)+1/dis4(dongu2)+1/dis6(dongu2)+1/dis8(dongu2));
115
116 -    rs1(dongu2)=1- 2*prob_bir1(dongu2)/ (prob_bir1(dongu2)+prob_sifir1(dongu2));
117 -    rs2(dongu2)=1- 2*prob_bir2(dongu2)/ (prob_bir2(dongu2)+prob_sifir2(dongu2));
118 -    rs3(dongu2)=1- 2*prob_bir3(dongu2)/ (prob_bir3(dongu2)+prob_sifir3(dongu2));
119 -    rs=[rs1 rs2 rs3];
120 -    end

```

Yakın komşuluklar (likelihoods) yardımıyla her bir çıkış bitinin değeri her iterasyonda yeniden elde edilmiştir. PSK işaretin her bitinin sıfır ve bir olma olasılıkları elde edildikten sonra bu olasılıklar LDPC kod çözücünde kullanılarak her seviye bir bite karşılık gelecek şekilde giriş bit dizileri kestirilir.

```

24 -     f01=1./(1+exp(-2*rs1/sigma^2));           % likelihoods
25 -     f11=1-f01;
26
27 -     f02=1./(1+exp(-2*rs2/sigma^2));           % likelihoods
28 -     f12=1-f02;
29
30 -     f03=1./(1+exp(-2*rs3/sigma^2));           % likelihoods
31 -     f13=1-f03;
32
33 -     [z_hat1, success, k] = ldpc_decode(rs1,f01,f11,Hb);
34 -     [z_hat2, success, k] = ldpc_decode(rs2,f02,f12,Hb);
35 -     [z_hat3, success, k] = ldpc_decode(rs3,f03,f13,Hb);
36
37 -     x_hat1 = z_hat1(size(Gb,2)+1-size(Gb,1):size(Gb,2));
38 -     x_hat1 = x_hat1';
39 -     x_hat2 = z_hat2(size(Gb,2)+1-size(Gb,1):size(Gb,2));
40 -     x_hat2 = x_hat2';
41 -     x_hat3 = z_hat3(size(Gb,2)+1-size(Gb,1):size(Gb,2));
42 -     x_hat3 = x_hat3';

```

Gerekli karşılaştırmalar yapılarak performans sonuçları ekrana yansıtılmıştır.

```

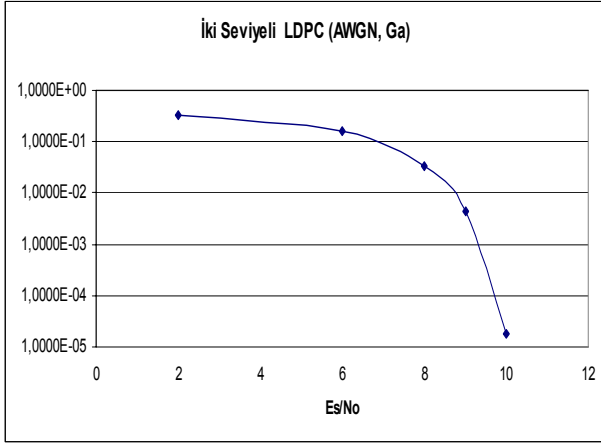
145 -     err(nframe)=length(find(x_hat1~=x1))+length(find(x_hat2~=x2))+length(find(x_hat3~=x3));
146 -     fprintf('+++++++ SD4 8PSK ++++++\n');
147 -     fprintf('%d frames are transmitted\n',3*nframe)
148 -     Pb=sum(err)/(nframe*length(x1)*3);
149 -     fprintf('Pb= %8.4e\n',Pb);
150 -     fprintf('+++++\n');
151 -     end

```

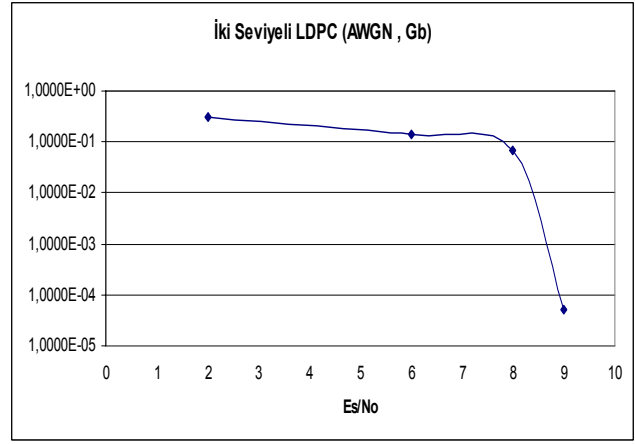
4.1.3. Benzetim Sonucunda Elde Edilen Grafikler

Farklı gürültü ortamlarında ve farklı üreteç matrisleri kullanılarak elde edilen Bit Hata Oranları (Bit Error Rate-BER) sonuçları aşağıda verilen grafiklerde görülmektedir.

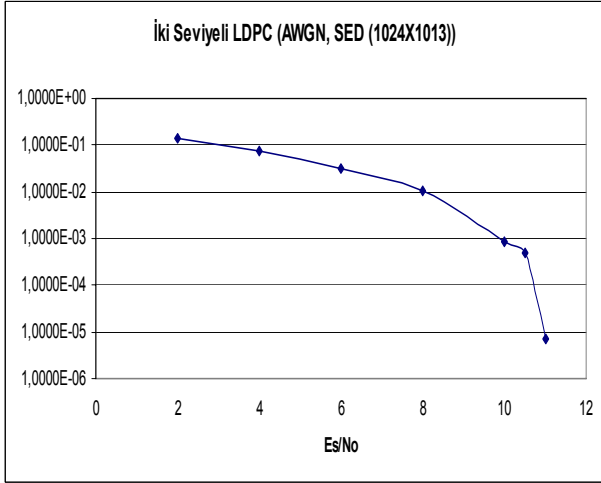
AWGN Ortamında İki ve Üç Seviyeli LDPC (Ga/Gb/SED Matrisleri)



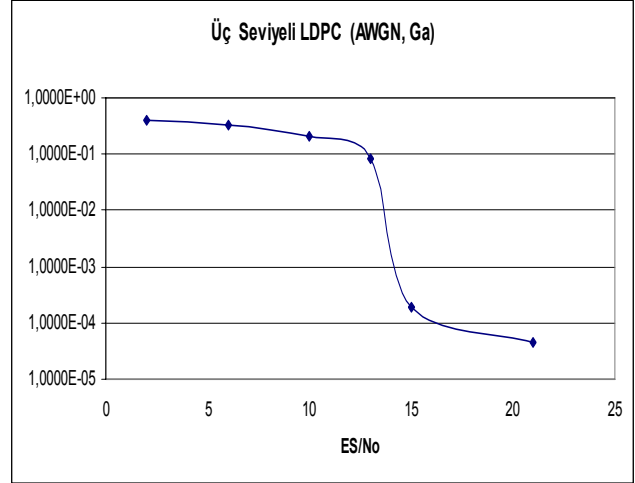
Şekil 3.2: İki seviyeli LDPC (AWGN, Ga)



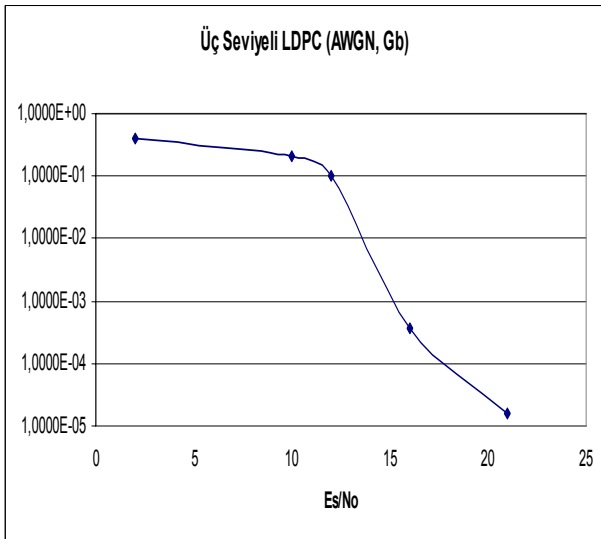
Şekil 3.3: İki seviyeli LDPC (AWGN, Gb)



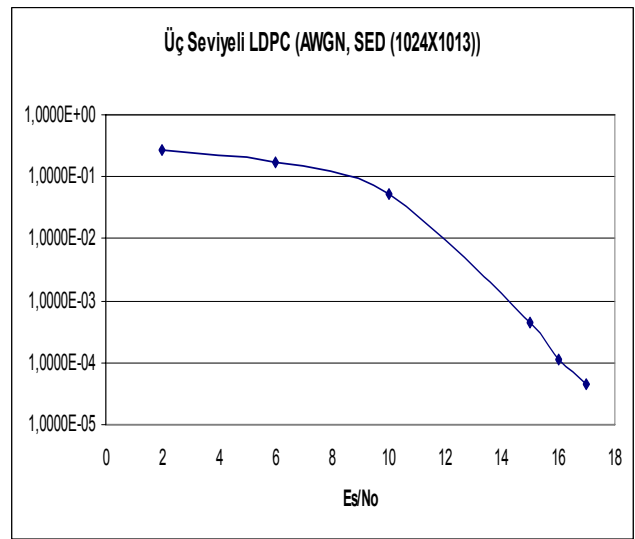
Şekil 3.4: İki seviyeli LDPC (AWGN, SED)



Şekil 3.5: Üç seviyeli LDPC (AWGN, Ga)

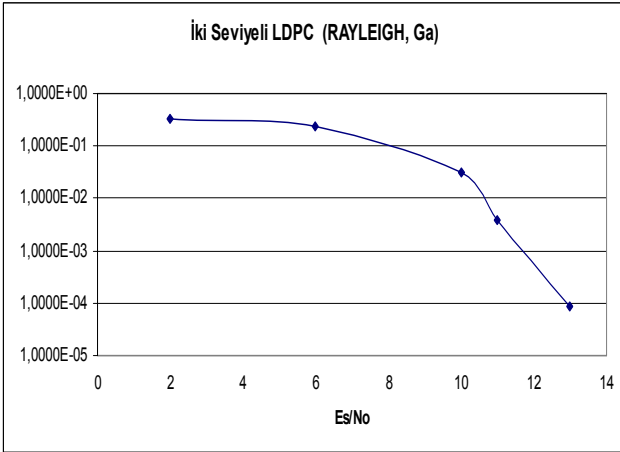


Şekil 3.6: Üç seviyeli LDPC (AWGN, Gb)

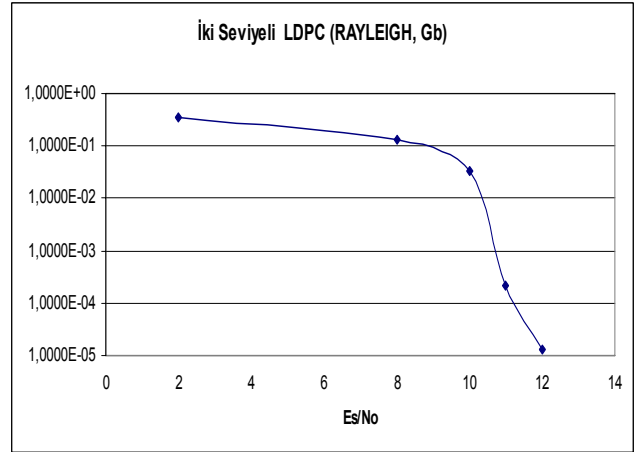


Şekil 3.7: Üç seviyeli LDPC (AWGN, SED)

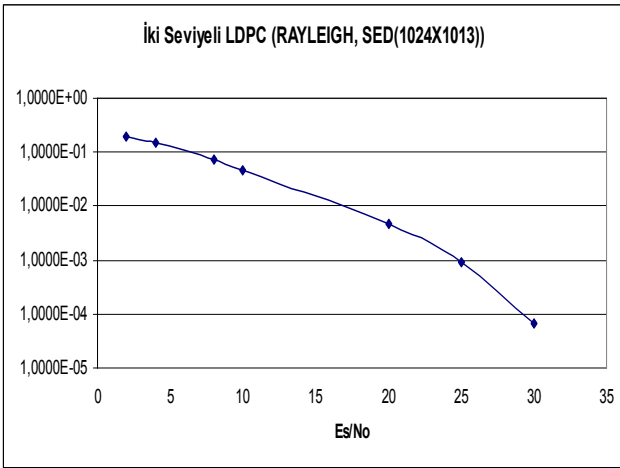
RAYLEIGH Ortamında İki ve Üç Seviyeli LDPC (Ga/Gb/SED Matrisleri)



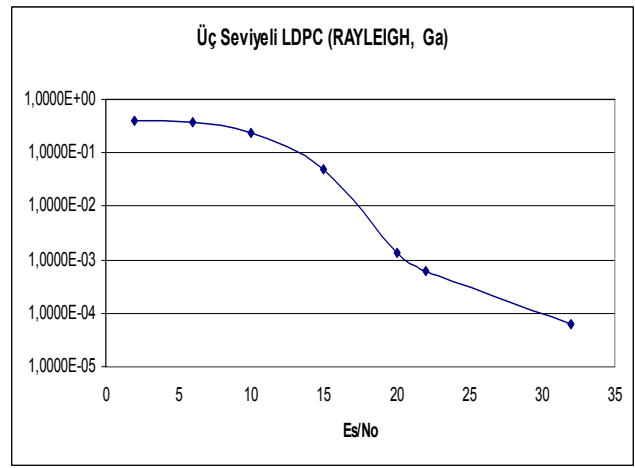
Şekil 3.8: İki seviyeli LDPC (Rayleigh, Ga)



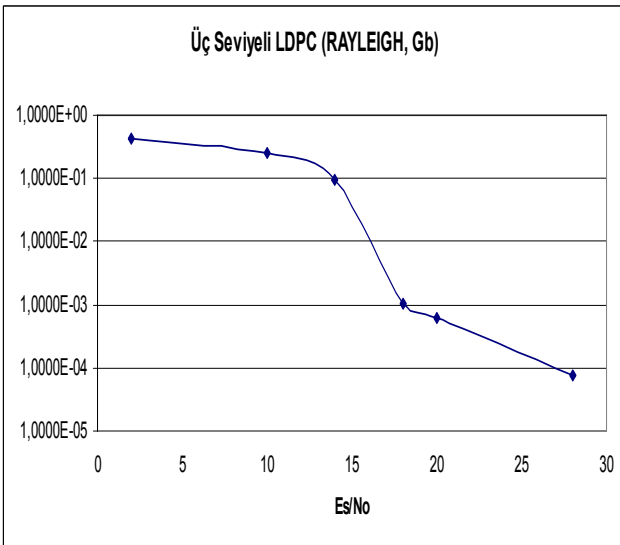
Şekil 3.9: İki seviyeli LDPC (Rayleigh, Gb)



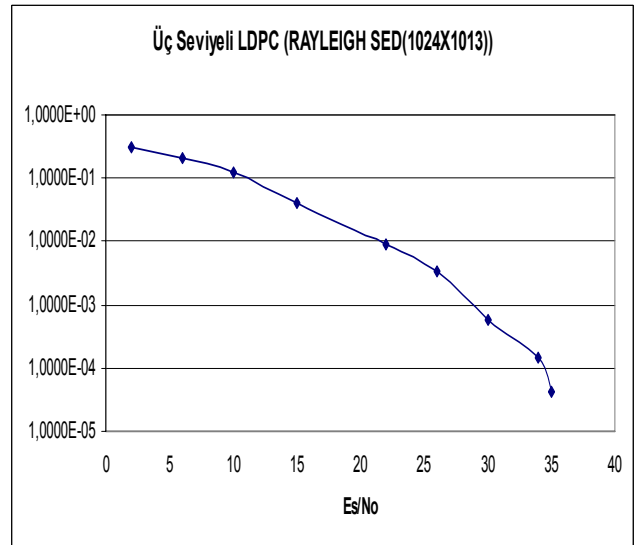
Şekil 3.10: İki seviyeli LDPC (Rayleigh, SED)



Şekil 3.11: Üç seviyeli LDPC (Rayleigh, Ga)

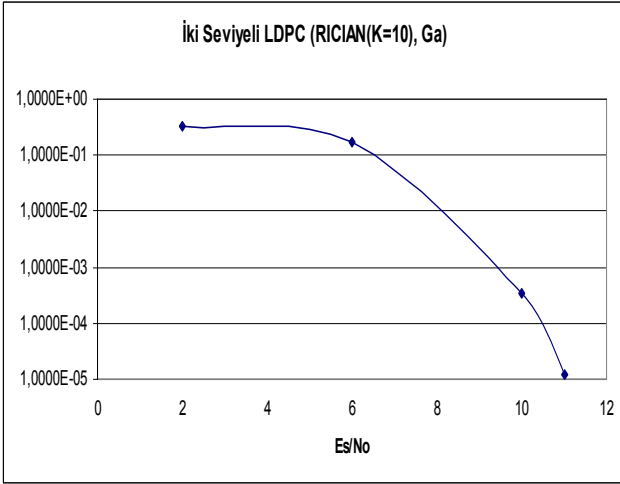


Şekil 3.12: Üç seviyeli LDPC (Rayleigh, Gb)

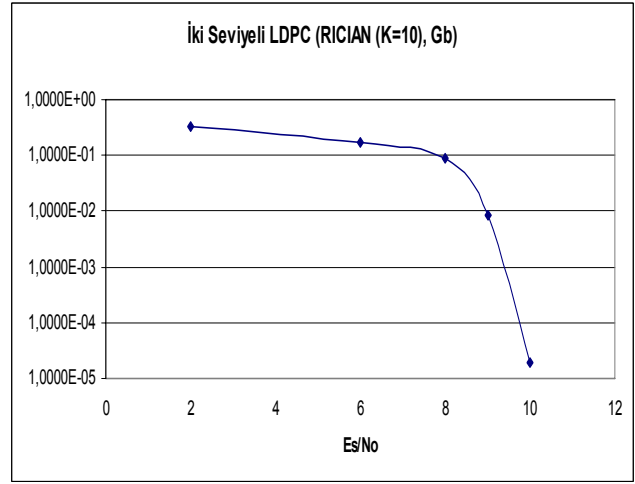


Şekil 3.13: Üç seviyeli LDPC (Rayleigh, SED)

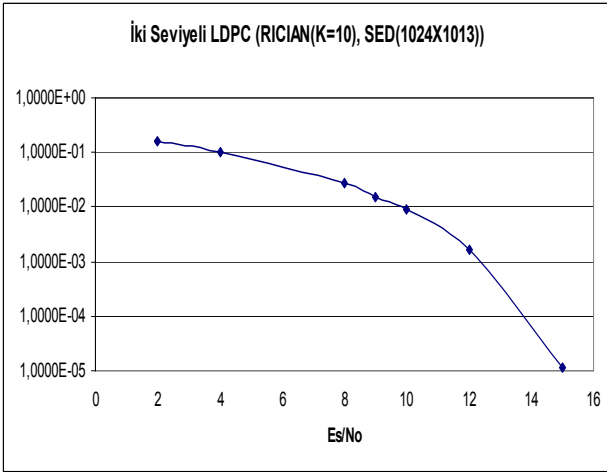
RICIAN(K=10) Ortamında İki ve Üç Seviyeli LDPC (Ga/Gb/SED Matrisleri)



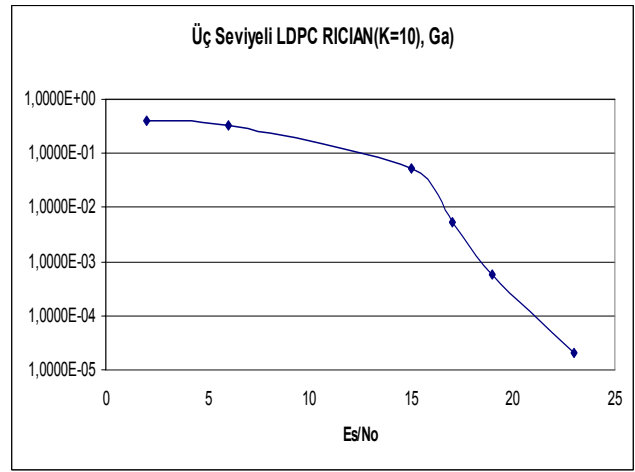
Şekil 3.14: İki seviyeli LDPC (Rician K=10, Ga)



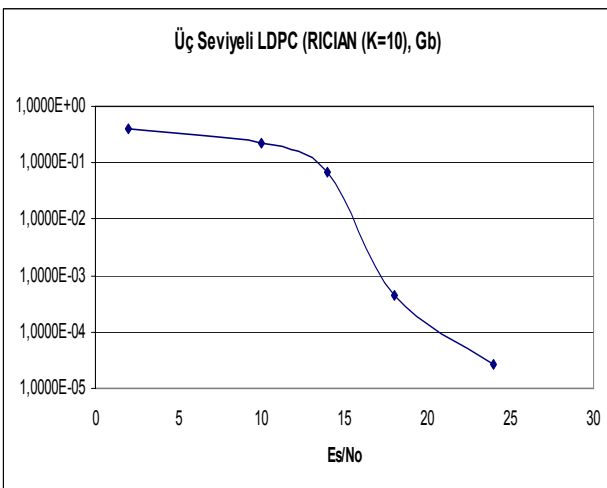
Şekil 3.15: İki seviyeli LDPC (Rician K=10, Gb)



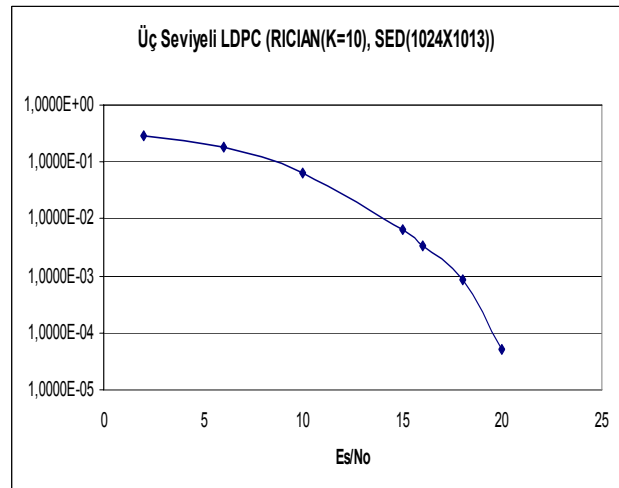
Şekil 3.16: İki seviyeli LDPC (Rician K=10, SED)



Şekil 3.17: Üç seviyeli LDPC (Rician K=10, Ga)

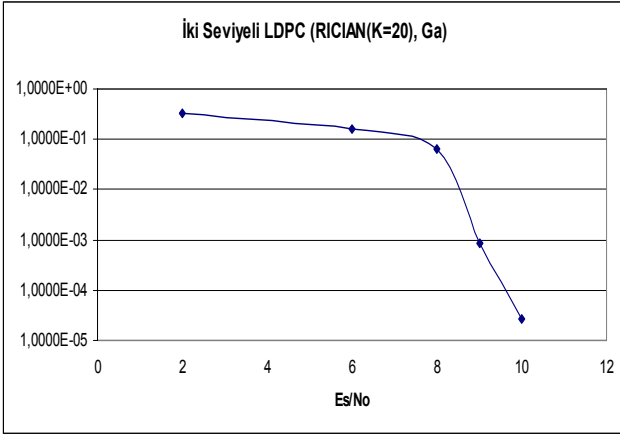


Şekil 3.18: Üç seviyeli LDPC (Rician K=10, Gb)

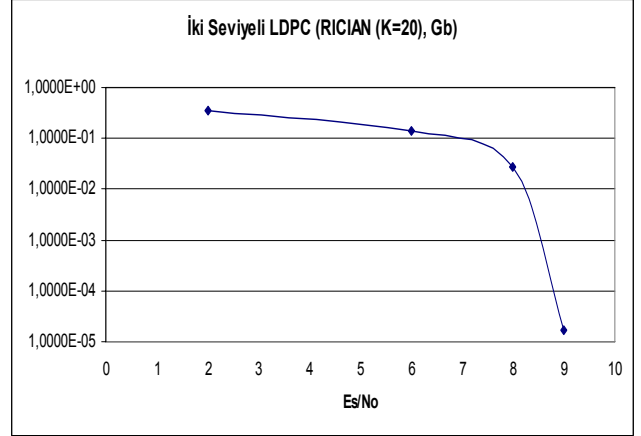


Şekil 3.19: Üç seviyeli LDPC (Rician K=10, SED)

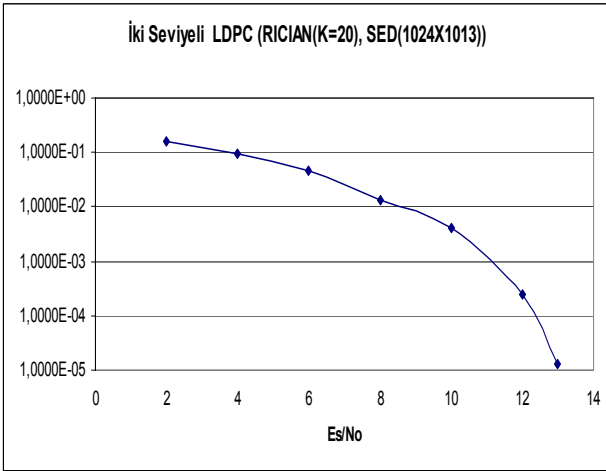
RICIAN (K=20) Ortamında İki ve Üç Seviyeli LDPC (Ga/Gb/SED Matrisleri)



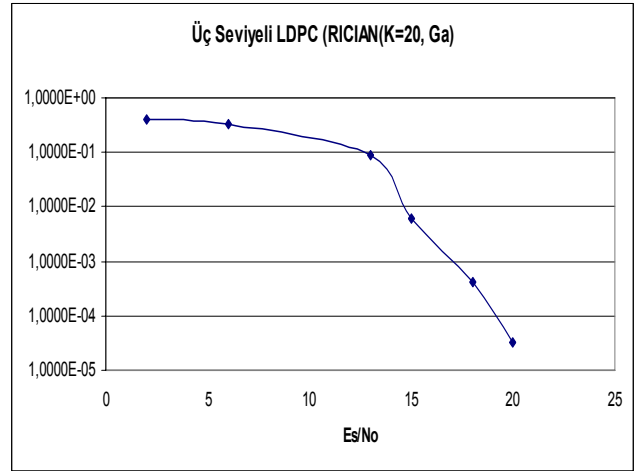
Şekil 3.20: İki seviyeli LDPC (Rician K=20, Ga)



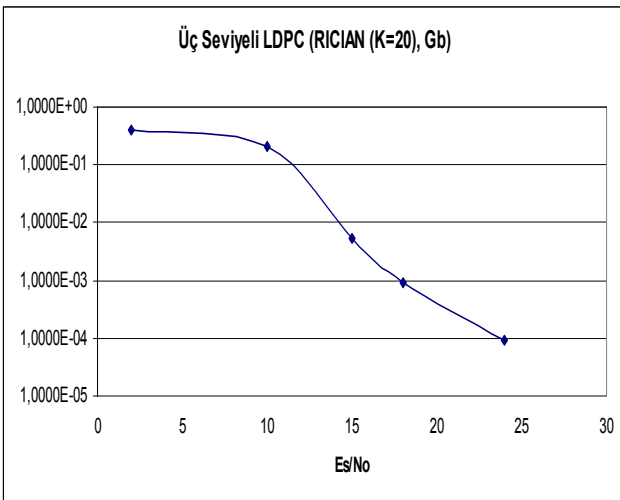
Şekil 3.21: İki seviyeli LDPC (Rician K=20, Gb)



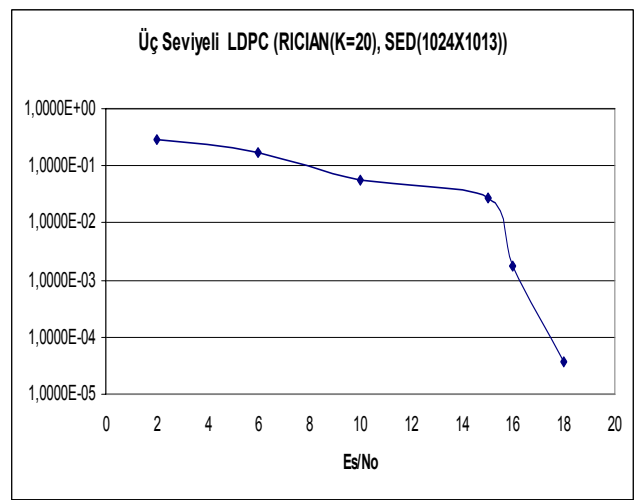
Şekil 3.22: İki seviyeli LDPC (Rician K=20, SED)



Şekil 3.23: Üç seviyeli LDPC (Rician K=20, Ga)



Şekil 3.24: Üç seviyeli LDPC (Rician K=20, Gb)



Şekil 3.25: Üç seviyeli LDPC (Rician K=20, SED)

4.1.4. Elde Edilen Bulgular

LDPC kodlama performansının istenen seviyede olduğu gözlemlenmiştir. Kullanılan üreteç matrisinin ve gürültü ortamının karmaşıklığına göre literatür taramalarında elde edilen performanslar ile karşılaştırıldığında beklenen sonuçlar elde edilmiş hatta daha iyi sonuçlar elde edilmiştir. Tez konusu olan, çok seviyeli LDPC kolamalı OFDM üzerine yoğunlaşılması gerektiğinden, elde edilen bu sonuçlardan göreceli olarak yararlanılarak değerlendirme yapılmıştır.

Benzetim ortamında dört farklı gürültü ortamı (AWGN, Rayleigh, Rician10, Rician20) ve üç farklı üreteç matrisi kullanılmıştır. Bu benzetim ile, bir LDPC programlama yapılması ile LDPC programlama mantığının öğrenilmesi ve daha sonra yapılacak çalışmalara hazırlıklı olunması amaçlanmıştır.

Şekil 3.2/3/4'te, iki seviyeli LDPC kodlama yapılarak, AWGN gürültü ortamında, üç farklı üreteç matrisi kullanılarak elde edilen sonuçlarda, en iyi değer için Gb üreteç matrisi kullanılması ile elde edildiği görülmektedir.

Şekil 3.5/6/7'de, üç seviyeli LDPC kodlama yapılarak, AWGN gürültü ortamında, üç farklı üreteç matrisi kullanılarak elde edilen sonuçlarda, en iyi değer için daha büyük olan SED üreteç matrisinin kullanılması ile elde edildiği görülmüştür.

Şekil 3.8/9/10'da, iki seviyeli LDPC kodlama yapılarak, Rayleigh gürültü ortamında, üç farklı üreteç matrisi kullanılarak elde edilen sonuçlarda, en iyi değer için Gb üreteç matrisi kullanılması ile elde edildiği görülmüştür.

Şekil 3.11/12/13'de, üç seviyeli LDPC kodlama yapılarak, Rayleigh gürültü ortamında, üç farklı üreteç matrisi kullanılarak elde edilen sonuçlarda, en iyi değer için Gb üreteç matrisi kullanılması ile elde edildiği görülmüştür.

Şekil 3.14/15/16'da, iki seviyeli LDPC kodlama yapılarak, Rician (10) gürültü ortamında, üç farklı üreteç matrisi kullanılarak elde edilen sonuçlarda, en iyi değer için Gb üreteç matrisi kullanılması ile elde edildiği görülmüştür.

Şekil 3.17/18/19'da, üç seviyeli LDPC kodlama yapılarak, Rician gürültü ortamında, üç farklı üreteç matrisi kullanılarak elde edilen sonuçlarda, en iyi değer büyük boyutta olan SED üreteç matrisinin kullanılması ile elde edildiği görülmüştür.

Şekil 3.20/21/22'de, iki seviyeli LDPC kodlama yapılarak, Rician (20) gürültü ortamında, üç farklı üreteç matrisi kullanılarak elde edilen sonuçlarda, en iyi değer Gb üreteç matrisi kullanılması ile elde edildiği görülmüştür.

Şekil 3.23/24/25'te, üç seviyeli LDPC kodlama yapılarak, Rician gürültü ortamında, üç farklı üreteç matrisi kullanılarak elde edilen sonuçlarda, en iyi değer SED üreteç matrisi kullanılması ile elde edildiği görülmüştür.

Hangi üreteç matrisi kullanılırsa kullanılsın, paralel veri transfer seviyesi (iki seviyeden üç seviyeye) arttırıldığı takdirde, sistem performansının göreceli olarak düştüğü gözlemlenmiştir. Transfer edilmek istenen verinin özelliğine göre, daha kısa zamanda daha çok veri gönderilmek istendiğinde çok seviyeli olarak kodlanması gerekecektir. Ancak bu durumda, gönderilen veri ile oluşacak hataların yüksekliği göz ardı edilmemelidir. Önemli olan, uygun bir hata olasılığı için, bulunulan sinyal gürültü oranına bağlı olarak seviye sayısını belirlemektir. Diğer bir şeyle hızı belirlemektir.

Test sonuçlarından görüleceği gibi, farklı gürültü ortamları birbirleriyle karşılaştırıldığında, en iyi sonucun AWGN gürültü ortamında alındığı görülmüştür. Bu beklenen bir durumdur. Çünkü diğer gürültü ortamlarına nazaran, bu gürültü ortamının en az karmaşıklığa sebep olduğu, dolayısıyla iletim ortamında gönderilen veriyi en az etkileyeceği açıktır.

Üç farklı üreteç matrisi kullanılarak elde edilen sonuçlar karşılaştırıldığında en iyi sonuçların Gb üreteç matrisi kullanıldığı durumlarda elde edildiği gözlemlenmiştir. Yaklaşık olarak dokuz desibel değerinde, ortalama Bit Hata Oranlarına ulaştığı gözlenmiştir. İkinci derecede verimli sonuçlar SED üreteç matrisinin kullanılması ile elde edilmiştir. Ancak SED matrisi çok büyük bir matris yapısına sahip olduğu için, işlem hızını yavaşlattığı gözlemlenmiştir. Ga üreteç matrisi daha küçük olmasına rağmen en kötü sonuçlar alınmıştır.

Sonuç olarak; Kodlama kalitesinin üreteç matrislerine bağlı olduğu, üreteç matrislerinin büyük olması iyi sonuçlar vermesine rağmen daha küçük üreteç matrislerinden daha iyi

sonular alınmadığı ve gürültü ortamının karmaşıklığına göre gönderilmek istenen veri kalitesinin deęiştığı gözlenmiştir. LDPC Kodlama yapıldığında, üreteç matrisinin uygunluğunun büyük önem gösterdiği gözlemlenmiştir. LDPC Seviyesinin artması ile veri iletim kalitesinde bozulmanın arttığı gözlenmiştir.

4.2. OFDM KODLAMA BENZETİMİ

4.2.1. Benzetim Ortamı

Matlab kullanılarak hazırlanan benzetim ortamında; SNR, doppler, Pilot Sembol ve PSK seviyelerinin etkilerinin görülmesi amaçlanmıştır. Benzetim ortamında taşıyıcı sayısı ve “Cyclic Prefix” sayısı sabit tutulmuştur.

4.2.2 Programlama Akışı

Aşağıda verilen program akışında yalnızca bir SNR, Doppler, Pilot Sembol ve PSK değeri kullanılmıştır. Söz konusu parametreler, her biri için manuel olarak deęiştirilerek tüm testler gerçekleştirilmiştir.

Benzetimde ilk adım olarak parametreler belirlenmiştir.

```

6 - K = 128;           % number of subcarriers
7 - Lcp = 32;         % length of the Cyclic Prefix
8 - N = K + Lcp;      % total duration of one OFDM symbol
9 - P = 16;           % number of pilots in one OFDM symbol
10 - S = K/P;         % space between adjacent pilot symbols
11 - SNR =16;

```

Her test başlangıcında etkisi incelenen parametreler manuel olarak deęiştirilerek sonular alınmıştır. Daha sonra istenen iterasyon değeri rastgele olarak veri bitlerinin oluşturulması sağlanmıştır.

```

23 - v=round(rand(K,3));
24 - data=bi2de(v);
25 - org_data=pskmod(data,8);

```

Rastgele veri bitleri elde edildikten sonra, Pilot Sembol eklenerek kod data elde edilmiştir.

```

27 - i=0:P-1;
28 - org_data(i*S+1)=1;      % inserting pilots

```

Kod data elde edildikten sonra OFDM Sembolü elde edilme işlemi gerçekleştirilmiştir.

```

55 - S1 = ifft(cod_data,K)./sqrt(K);
56 - S2 = [S1(K - Lcp + 1 : K) S1];

```

Daha sonra kanal kodlaması yapılmıştır. Bir kanalda beş ayrı yansıma eklenebilmiş ve her yansıyan sinyalin parametrelerinin manuel olarak değiştirilme imkanı sağlanmış, denemeler yapılarak optimum yansıma katsayıları belirlenmiştir. Ayrıca kanaldaki her yansımanın maruz kalacağı gecikme ve doppler etkisinin de değiştirilmesi olanağı sağlanmıştır.

```

75 - L=5;
76 - alpha = [1 0.55 0.25 0 0]';
77 - delay = [0 4 8 0 0]';
78 - %doppler = [0 0 0 0 0];
79 - doppler = [0.0002*2*pi 0.0002*2*pi 0.0002*2*pi 0 0]';
80
81 - m = [0 : N-1];
82 - m = [0 : N-1];
83 - y1 = alpha(1) * exp (j * doppler(1) .* m) .* [zeros(1,delay(1)) S2(1:N-delay(1))];
84 - y2 = alpha(2) * exp (j * doppler(2) .* m) .* [zeros(1,delay(2)) S2(1:N-delay(2))];
85 - y3 = alpha(3) * exp (j * doppler(3) .* m) .* [zeros(1,delay(3)) S2(1:N-delay(3))];
86 - y4 = alpha(4) * exp (j * doppler(4) .* m) .* [zeros(1,delay(4)) S2(1:N-delay(4))];
87 - y5 = alpha(5) * exp (j * doppler(5) .* m) .* [zeros(1,delay(5)) S2(1:N-delay(5))];
88 - Y = y1 + y2 + y3 + y4 + y5;

```

Daha sonra kanalda oluşacak gürültü benzetimi yapılmıştır.

```

112 - SNRW=10^(SNR/10);
113 - varS2=var(S2);
114 - varN=varS2/SNRW/2;
115 - noise=sqrt(varN)*randn(1,N)+j*sqrt(varN)*randn(1,N);
116 - Rec=Y+noise;

```

Alıcı tarafından demodülasyon işleminin başlaması için, veri içermeyen “Cyclic Prefix” bilgileri alınan data dan çıkartılmış ve demodülasyon işlemi gerçekleştirilmiştir.

```

130     % Demodulation
131
132 -    Rec2 = Rec(Lcp+1:N); %cyclic prefix is removed
133
134 -    Rec3 = fft(Rec2,K).*sqrt(K); % OFDM symbol is demodulated

```

Parametreleri bilinmeyen kanal benzetimi yapılmış ve Bit Hata Oranı (BER1) çıkarılmıştır.

```

148     % The worst case: no channel estimation
149 -    est_symb=pskdemod(Rec3,8)';
150 -    est_data0=de2bi(est_symb);
151
152     %error = sum(abs(est_data0-cod_data)/2);
153 -    t=0;
154 -    for i=1:K,
155 -        if mod(i-1,S)~=0
156 -            t=t+1;
157 -            est_data0noP(t,:) = est_data0(i,:);
158 -            vnoP(t,:) = v(i,:);
159 -        end
160 -    end
161
162     %error1(iter) = sum(abs(est_data0-cod_data)/2);
163 -    error1(iter) = sum(sum(abs(est_data0noP-vnoP)));

```

Pilot Sembol eklenen kanal benzetimi yapılmış ve Bit Hata Oranı (BER2) çıkarılmıştır.

```

167     % Pilot Symbol Assisted Channel Estimation
168
169 -    rec_pilot = zeros(1,P);
170     %rec_pilot = Rec3;
171 -    pilot_ch = zeros(1,P);
172     %channel = cod_data./rec_pilot;
173 -    est_data1 = zeros(1,K);
174     %est_data = Rec3.*channel;
175 -    for i=0:P-1,
176 -        rec_pilot(i+1) = Rec3(i*S+1);
177 -        pilot_ch(i+1) = rec_pilot(i+1)./cod_data(i*S+1);
178 -    end
179
180 -    ch=interp(pilot_ch,S);

```

```

189
190 -     for i=0:P-1,
191 -         est_data1(i*S+1) = cod_data(i*S+1); % Rec3(i*S+1) / ch(i*S+1);
192 -         for ii=2:S
193 -             est_data1(i*S+ii) = Rec3(i*S+ii) / ch(i*S+ii);
194 -         end
195 -     end
196
197 -     est_symb2=pskdemod(est_data1,8)';
198 -     est_data1bit=de2bi(est_symb2);
199
200 -     %error = sum(abs(est_data0-cod_data)/2);
201 -     t=0;
202 -     for i=1:K,
203 -         if mod(i-1,S)~=0
204 -             t=t+1;
205 -             est_data1bitnoP(t,:) = est_data1bit(i,:);
206 -             vnoP(t,:) = v(i,:);
207 -         end
208 -     end
209
210 -     %error1(iter) = sum(abs(est_data0-cod_data)/2);
211 -     error2(iter) = sum(sum(abs(est_data1bitnoP-vnoP)));

```

Parametreleri bilinen kanal benzetimi yapılmış ve Bit Hata Oranı (BER3) çıkarılmıştır.

```

251 -     % Calculation with the Known Channel Parameters
252 -     W=(2*pi)/K;
253 -     for m=1:K,
254 -         for k=1:K,
255 -             H0(m,k) = alpha(1)*exp(j*doppler(1)*(m-1))*exp(-j*delay(1)*W*(k-1));
256 -             H1(m,k) = alpha(2)*exp(j*doppler(2)*(m-1))*exp(-j*delay(2)*W*(k-1));
257 -             H2(m,k) = alpha(3)*exp(j*doppler(3)*(m-1))*exp(-j*delay(3)*W*(k-1));
258 -             H3(m,k) = alpha(4)*exp(j*doppler(4)*(m-1))*exp(-j*delay(4)*W*(k-1));
259 -             H4(m,k) = alpha(5)*exp(j*doppler(5)*(m-1))*exp(-j*delay(5)*W*(k-1));
260 -         end
261 -     end
262 -     H=H0+H1+H2+H3+H4;
263 -     B=fft(H);
264 -     SS=ifft(B. ');
265 -     w=2*pi/K:2*pi/K:2*pi; n=[1:K];

```

```

280 -     hk=H(1,:);
281 -     est_data2 = Rec3./hk;
282
283 -     est_symb3=pskdemod(est_data2,8)';
284 -     est_data2bit=de2bi(est_symb3);
285
286     %error = sum(abs(est_data0-cod_data)/2);
287 -     t=0;
288 -     for i=1:K,
289 -         if mod(i-1,S)~=0
290 -             t=t+1;
291 -             est_data2bitnoP(t,:) = est_data2bit(i,:);
292 -             vnoP(t,:) = v(i,:);
293 -         end
294 -     end
295
296     %error1(iter) = sum(abs(est_data0-cod_data)/2);
297 -     error3(iter) = sum(sum(abs(est_data2bitnoP-vnoP)));

```

Son olarak üç ayrı kanal türüne göre ortaya çıkan Bit Hata Oranlarının değerleri elde edilmiştir.

```

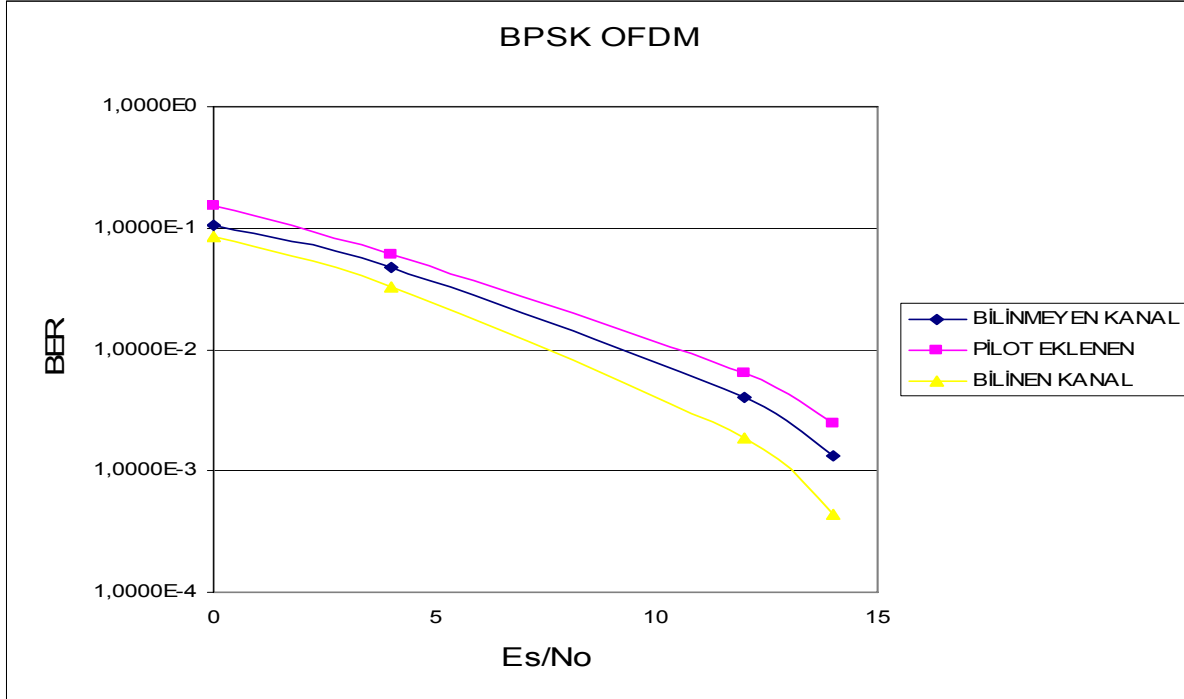
317 -     error1top=error1top+error1(iter);
318 -     error2top=error2top+error2(iter);
319 -     error3top=error3top+error3(iter);
320
321 -     BER1=error1top/(iter*3*(K-P));
322 -     BER2=error2top/(iter*3*(K-P));
323 -     BER3=error3top/(iter*3*(K-P));
324
325 -     fprintf('\n*****\n');
326 -     fprintf('%d frame is transmitted \n', iter);
327 -     fprintf('No Channel State Information\n');
328 -     fprintf('BER1=%4.5e\n',BER1);
329 -     fprintf('Pilot Assisted Channel State Information\n');
330 -     fprintf('BER2=%4.5e\n',BER2);
331 -     fprintf('Perfect Channel State Information\n');
332 -     fprintf('BER3=%4.5e\n',BER3);
333
334
335 -     end

```

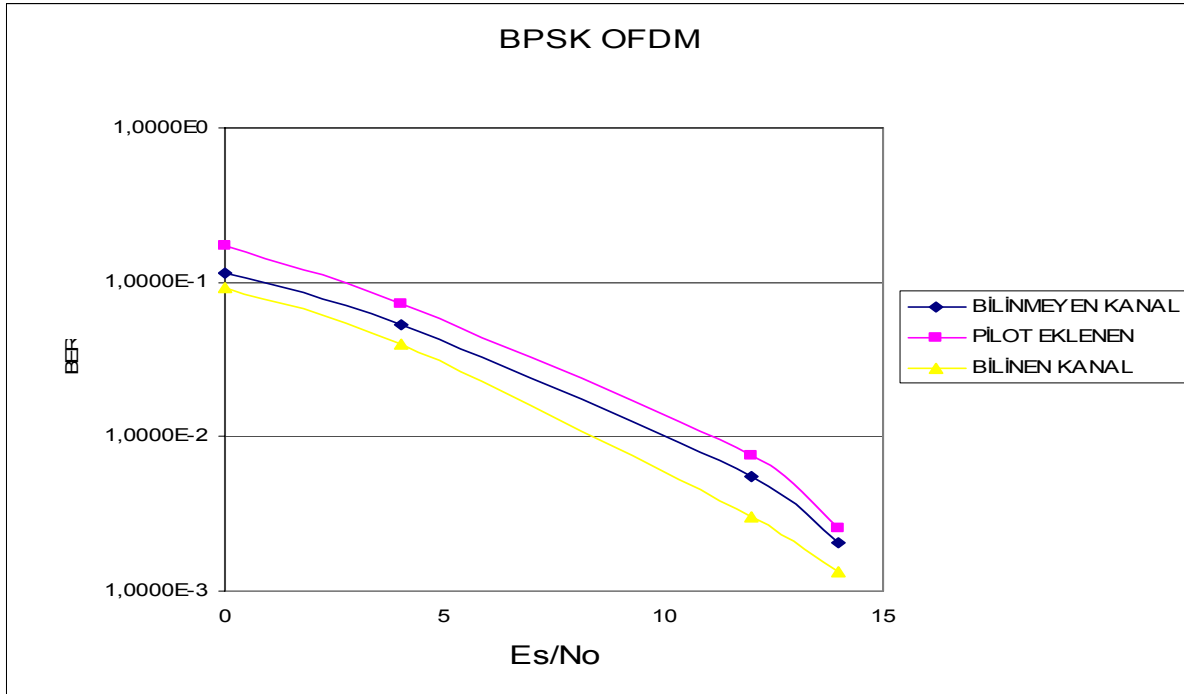
4.2.3. Benzetim Sonucunda Elde Edilen Grafikler

OFDM'i etkileyen bir çok parametre bulunmaktadır. Yapılan benzetim çalışmasında, yalnızca Doppler ve Pilot Sembol parametrelerinin etkisi gözlenmiştir. Benzetim ile elde edilen sonuçlar farklı grafikler ile aşağıda verilmiştir.

BPSK (Pilot Sembol Etkisi)



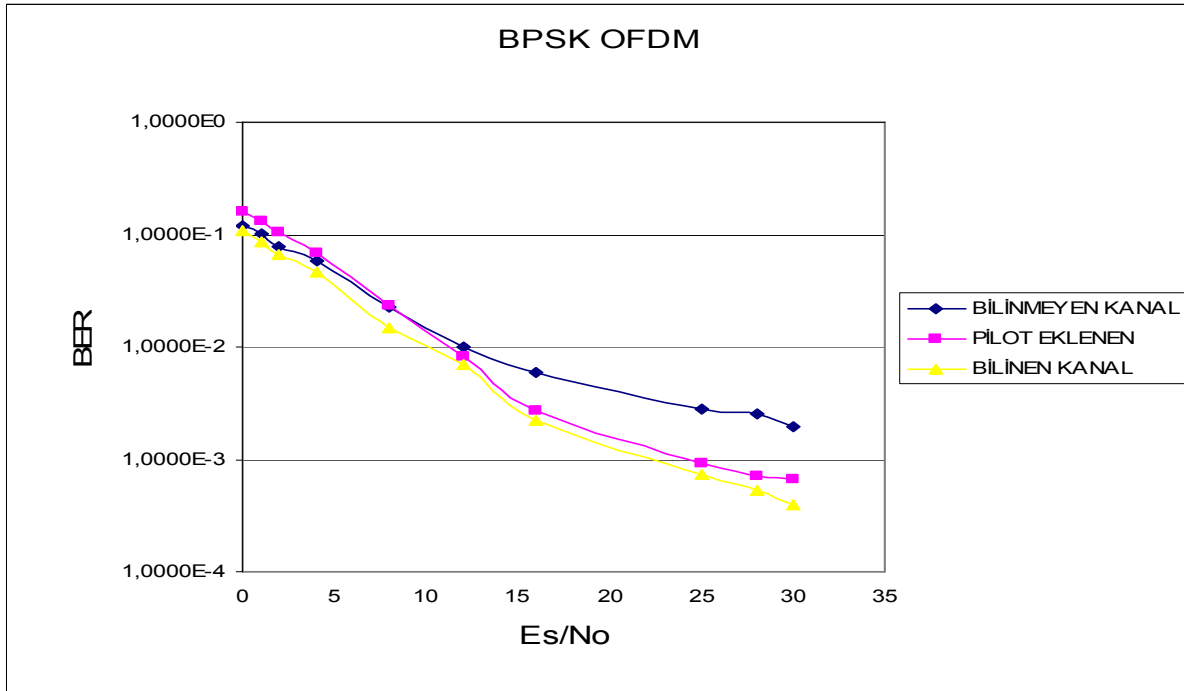
N=128	KANAL	1	0.55	0.25
P=64	GECİKME	0	4	8
CP:32	DOPPLER	0	0	0



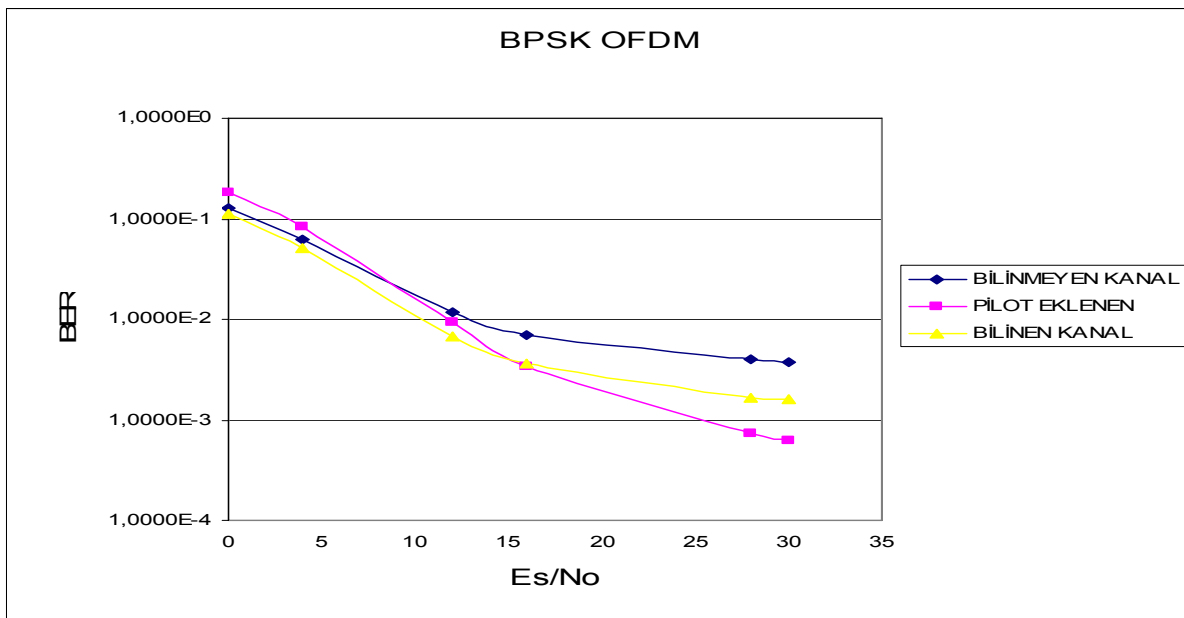
N=128	KANAL	1	0.55	0.25
P=16	GECİKME	0	4	8
CP:32	DOPPLER	0	0	0

Şekil 3.26: BPSK modülasyonlu OFDM'e Pilot Sembol etkisi

BPSK (Doppler Etkisi)

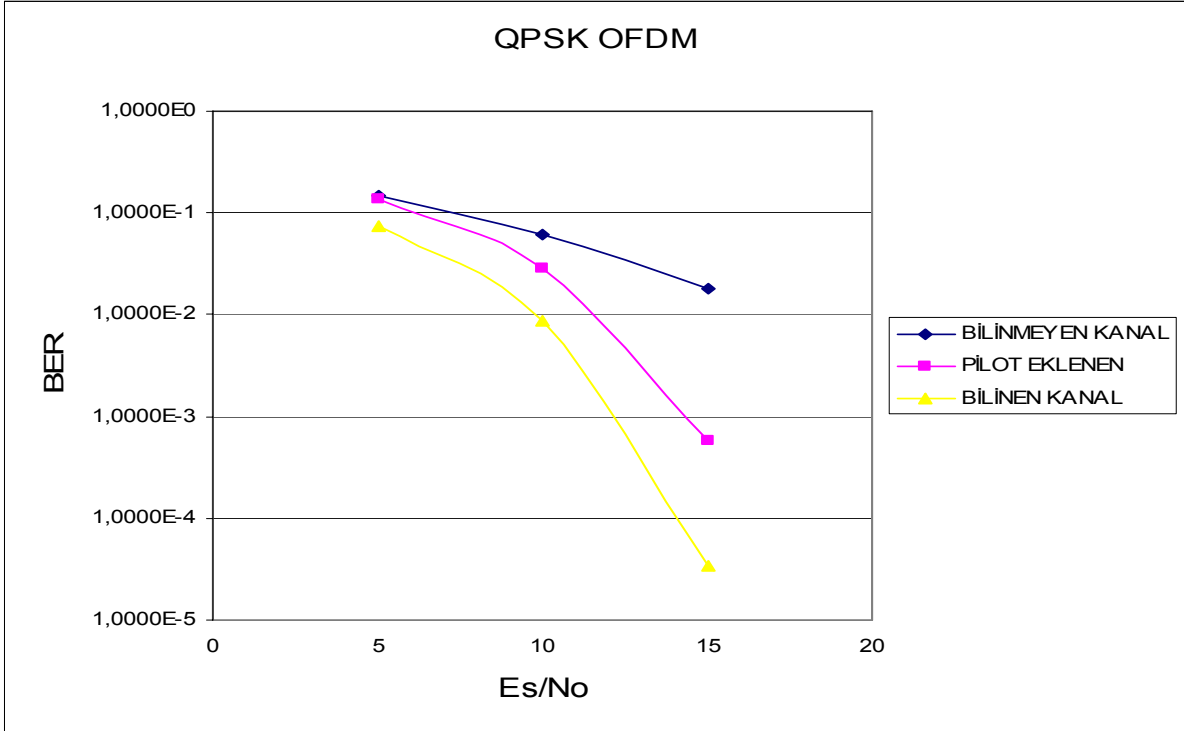


N=128	KANAL	1 0.55 0.25
P=64	GECİKME	0 4 8
CP:32	DOPPLER	0.0002*2*pi 0.0002*2*pi 0.0002*2*pi

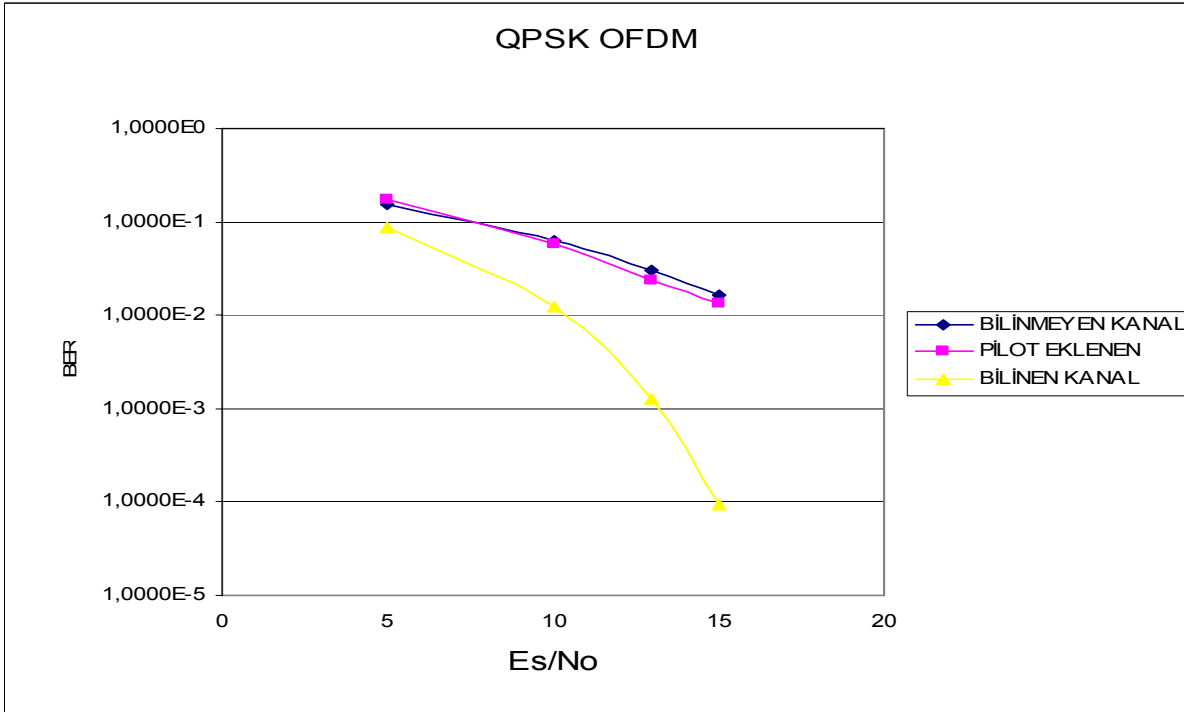


N=128	KANAL	1 0.55 0.25
P=16	GECİKME	0 4 8
CP:32	DOPPLER	0.0002*2*pi 0.0002*2*pi 0.0002*2*pi

Şekil 3.27: BPSK modülasyonlu OFDM'e Doppler etkisi



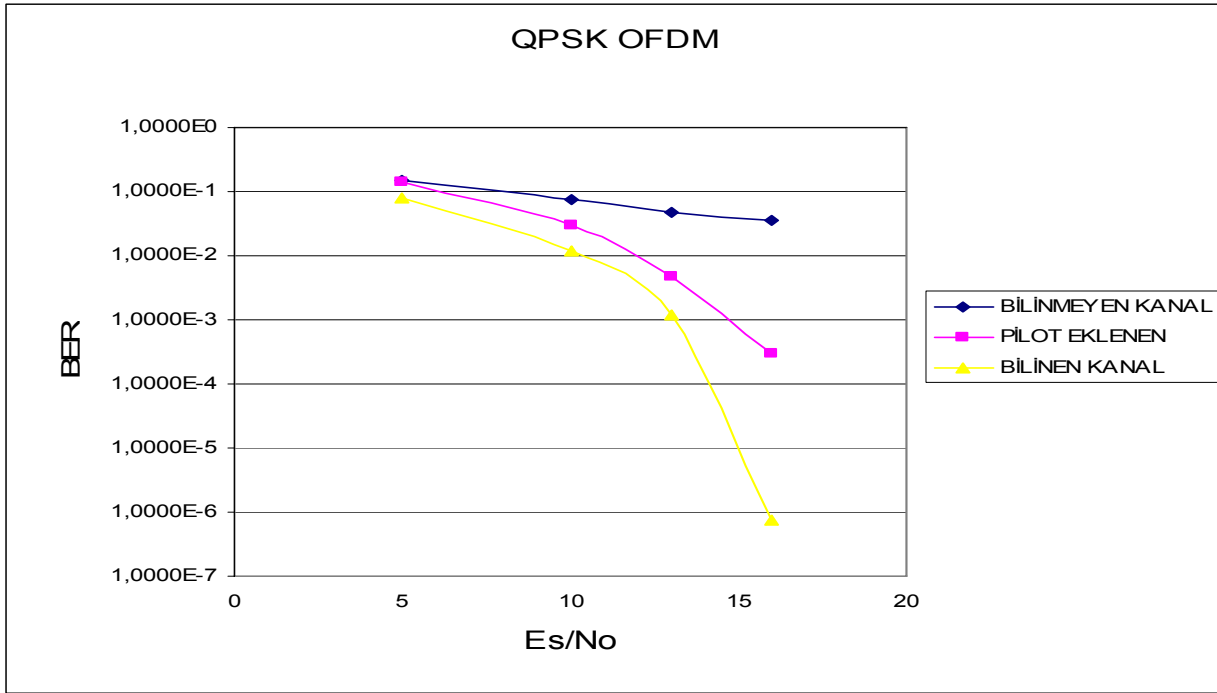
N=128 KANAL 1 0.55 0.25
 P=64 GECİKME 0 4 8
 CP:32 DOPPLER 0 0 0



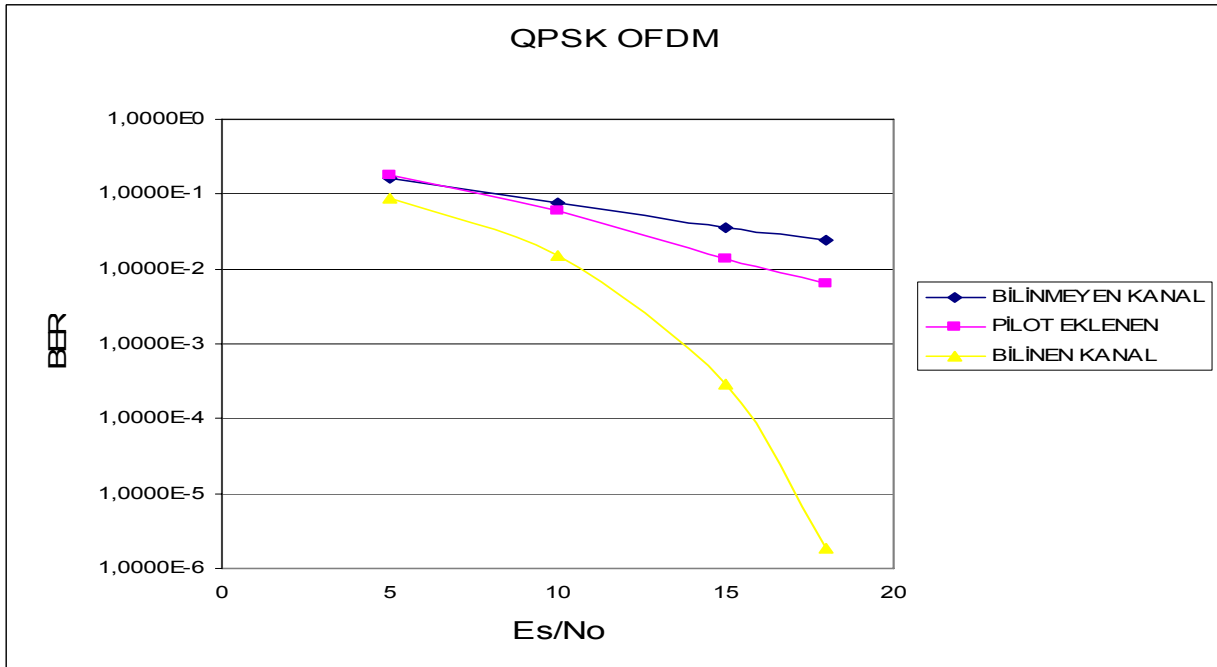
N=128 KANAL 1 0.55 0.25
 P=16 GECİKME 0 4 8
 CP:32 DOPPLER 0 0 0

Şekil 3.28: QPSK modülasyonlu OFDM'e Pilot Sembol etkisi

QPSK (Doppler Etkisi)



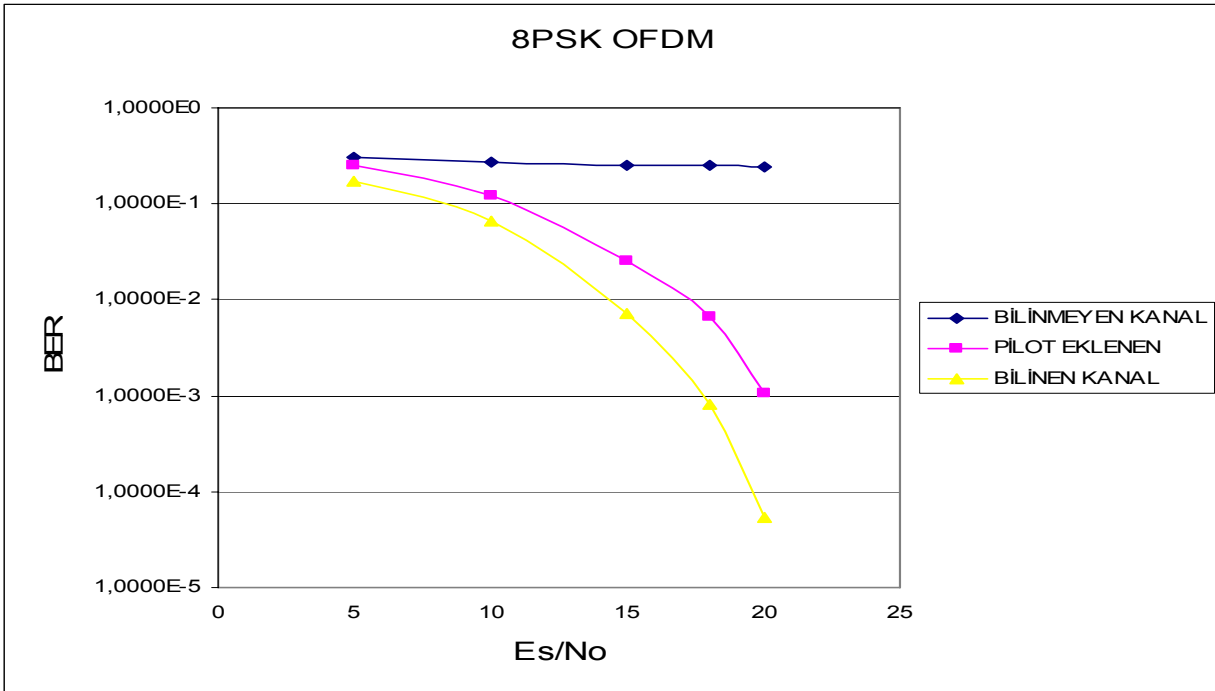
N=128	KANAL	1	0.55	0.25
P=64	GECİKME	0	4	8
CP:32	DOPPLER	0.0002*2*pi	0.0002*2*pi	0.0002*2*pi



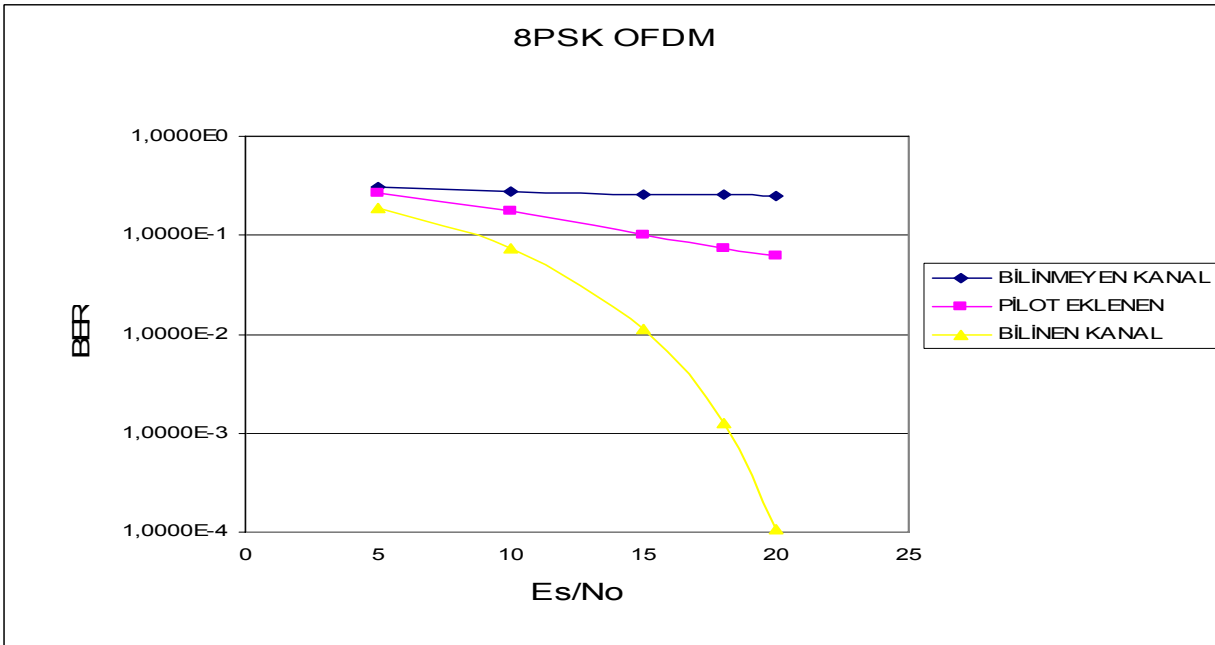
N=128	KANAL	1	0.55	0.25
P=16	GECİKME	0	4	8
CP:32	DOPPLER	0.0002*2*pi	0.0002*2*pi	0.0002*2*pi

Şekil 3.29: QPSK modülasyonlu OFDM'e Doppler etkisi

8PSK (Pilot Sembol Etkisi)



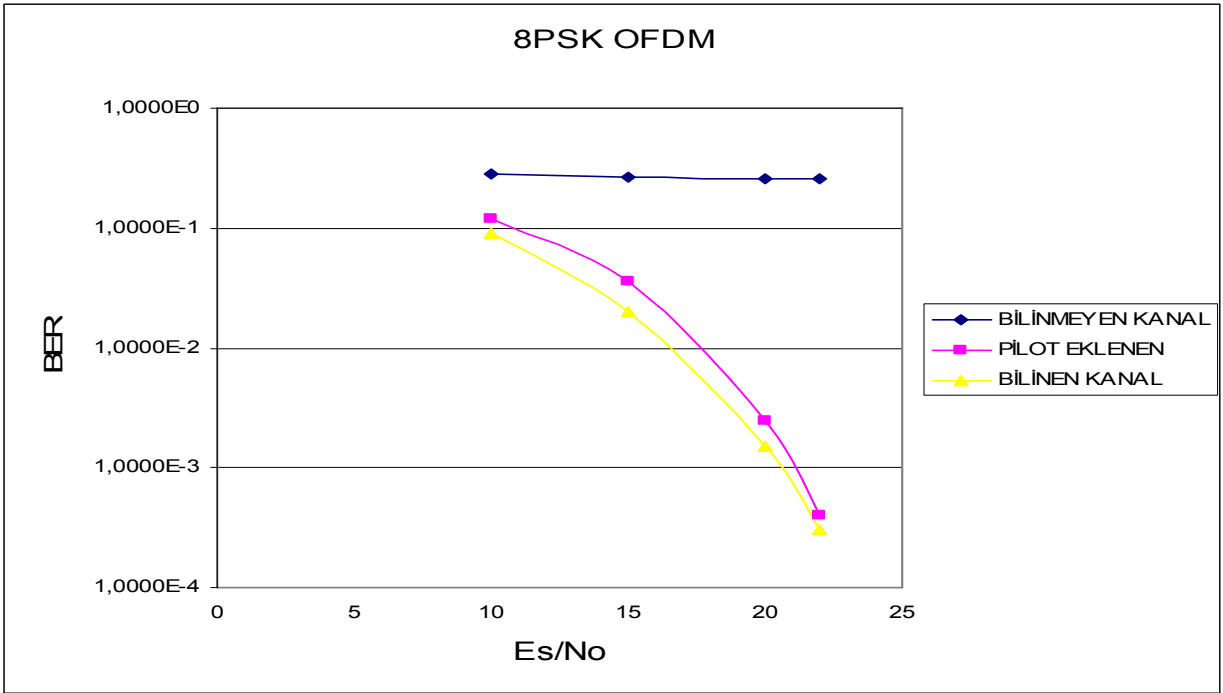
N=128	KANAL	1	0.55	0.25
P=64	GECİKME	0	4	8
CP:32	DOPPLER	0	0	0



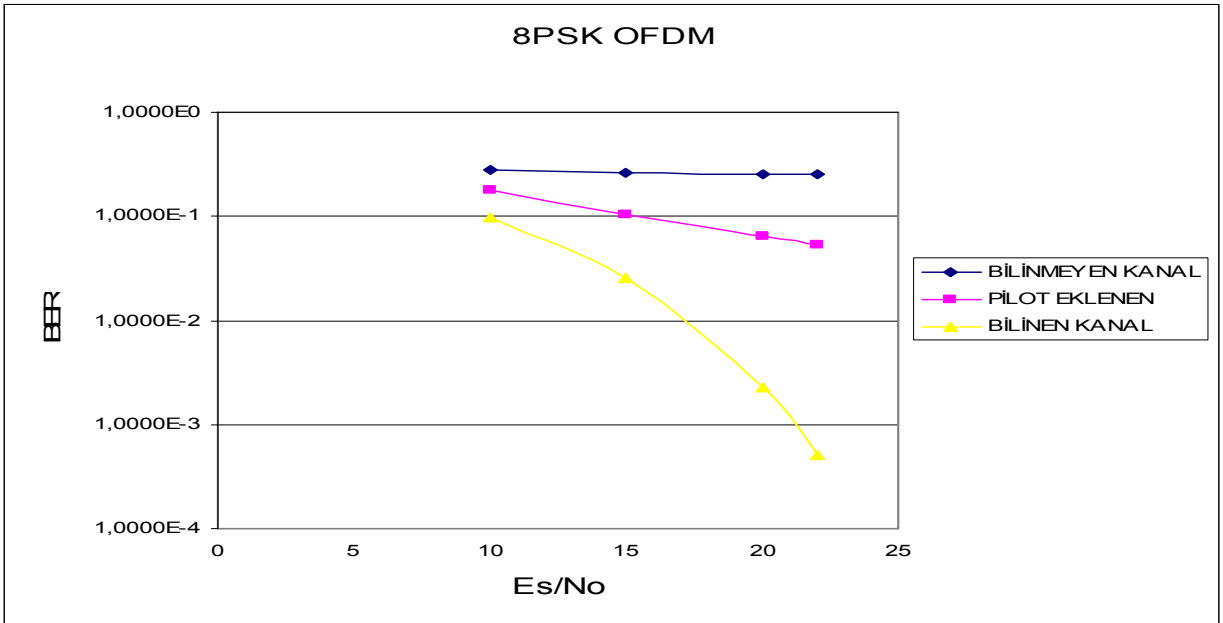
N=128	KANAL	1	0.55	0.25
P=16	GECİKME	0	4	8
CP:32	DOPPLER	0	0	0

Şekil 3.30: 8PSK modülasyonlu OFDM'e Pilot Sembol etkisi

8PSK (Doppler Etkisi)



N=128 KANAL 1 0.55 0.25
P=64 GECİKME 0 4 8
CP:32 DOPPLER 0.0002*2*pi 0.0002*2*pi 0.0002*2*pi



N=128 KANAL 1 0.55 0.25
P=16 GECİKME 0 4 8
CP=32 DOPPLER 0.0002*2*pi 0.0002*2*pi 0.0002*2*pi

Şekil 3.31: 8PSK modülasyonlu OFDM'e Doppler etkisi

4.2.4. Elde Edilen Bulgular

Benzetim sonucunda, OFDM kodlama yapısına, Doppler ve Pilot Sembolün etkisi gözlenmiştir. Elde edilen sonuçların beklenildiği gibi gerçekleştiği gözlenmiştir.

Şekil 3.26’da görüleceği gibi, BPSK modüleli OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve “Cyclic Prefix” sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenmemiştir. Bu durumda en iyi sonuç, beklendiği gibi parametreleri bilinen kanalda işlem yapılması durumunda elde edilmiştir. Pilot Sembol eklenmesi durumunda, parametreleri bilinmeyen kanaldan daha kötü bir sonuç elde edilmiştir. Çünkü, BPSK yapısında gönderilen bit sayısının az olması ve Pilot Sembolün bu bitler ile kıyaslanabilir seviyede olması nedeniyle, kod kelimesinde bilgi bitlerinin oranı azalmaktadır. Yine beklendiği gibi parametreleri bilinmeyen kanal kullanılması durumunda en kötü sonuçlar elde edilmiştir.

Şekil 3.27’de görüleceği gibi, tekrar BPSK modüleli OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve “Cyclic Prefix” sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenerek değişim gözlenmiştir. Bu durumda en iyi sonuç, beklendiği gibi parametreleri bilinen kanalda işlem yapılması durumunda elde edilmiştir. Pilot Sembolün eklenmesi ile, Doppler etkisinin olumsuz etkilerinin bertaraf edildiği ve beklenildiğinden daha iyi bir etki yaptığı gözlenmiştir. Yine beklendiği gibi parametreleri bilinmeyen kanal kullanılması durumunda en kötü sonuçlar elde edilmiştir.

Şekil 3.28’de görüldüğü gibi, QPSK modüleli OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve “Cyclic Prefix” sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenmemiştir. Bu durumda en iyi sonuç, beklendiği gibi parametreleri bilinen kanalda işlem yapılması durumunda elde edilmiştir. Genel olarak hata başarımlarının daha iyi olduğu gözlenmiştir. Pilot Sembol beklenen etkiyi göstermiştir. Pilot Sembol sayısının artması ile iyileşme gözlenmiştir. Karmaşık bir yapı olan QPSK modülasyonu kullanılmasına rağmen, hata başarımlarında ciddi bir performans kaybı gözlenmemiştir.

Şekil 3.29’da görüleceği gibi, QPSK modüleli OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve “Cyclic Prefix” sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenmiştir. Bu durumda en iyi sonuç, beklendiği gibi parametreleri bilinen kanalda işlem yapılması durumunda elde edilmiştir. Doppler etkisi nedeniyle sonuçlar kötüleşmesine

rağmen, Pilot Sembol sayısının arttırılması ile sonuçlar daha iyi duruma gelmiştir. Modülasyon türünün karmaşıklığına rağmen hata oranında kötüleşme gözlenmemiştir.

Şekil 3.30'da görüleceği gibi, 8PSK modüleli OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve "Cyclic Prefix" sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenmemiştir. Bu durumda en iyi sonuç, beklendiği gibi parametreleri bilinen kanalda işlem yapılması durumunda elde edilmiştir. 8PSK yapısı nedeniyle, genel sonuçlarda kötüleşme gözlenmekte ve parametreleri bilinmeyen kanal kullanılması durumunda alıcı tarafında neredeyse hiç veri alınmamaktadır. Pilot Sembol sayısının arttırılması ile iyi sonuçlar alınmıştır. Hatta, parametresi bilinen kanal kullanılması durumunda elde edilen hata değerlerine yakın sonuçlar alınmıştır.

Şekil 3.31'de görüleceği gibi, 8PSK modüleli OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve "Cyclic Prefix" sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenmiştir. Bu durumda en iyi sonuç, beklendiği gibi parametreleri bilinen kanalda işlem yapılması durumunda elde edilmiştir. Ancak Doppler eklenmesi nedeniyle genel olarak başarımlar sonuçları kötüleşmiştir. Pilot Sembol sayısının artması ile, sonuçların neredeyse parametreleri bilinen kanal ile aynı seviyede gerçekleştiği gözlenmiştir. Modülasyon derecesinin artması durumunda, Doppler etkisi ile sonuçlar kötüleşmektedir. Ancak Pilot Sembol sayısının arttırılmasının ise sonucu iyileştirdiği gözlenmiştir.

Sonuç olarak, Doppler etkisinin hata başarımında kötüleşmeye sebep olduğu, PSK derecesinin arttırılmasının, aynı zamanda gönderilen veri miktarının artmasına rağmen hata başarımını düşürdüğü ancak pilot Sembol sayısının arttırılmasının hata başarımını arttırdığı net bir şekilde söylenebilmektedir.

Elde edilen sonuçlar ışığında; gönderilmesi gereken veri hızına, kanal koşullarına ve kabul edilebilir hata başarımlar seviyesine göre sistem yapısının oluşturulmasında, ihtiyaca uygun modeller oluşturulabilecektir.

4.3. ÇOK SEVİYELİ LDPC KODLAMALI OFDM

4.3.1. Benzetim Ortamı

LDPC kodlam ile OFDM haberleşme ortamı birleştirilerek OFDM Tabanlı haberleşme Ortamında LDPC Kodlanmış haberleşme kanalı simüle edilmiştir. LDPC Kodlamalı OFDM haberleşme sisteminin genel yapısı Şekil 3.1'de verilmiştir. Bilindiği gibi OFDM ile ilgili birçok parametre vardır. Benzetimde kullanılan OFDM ile ilgili parametreler ise Tablo 3.1'de görülmektedir.

Parametreler	Değerleri
Toplam Taşıyıcı sayısı	$N=128$
Bilgi Taşıyan Taşıyıcı Sayısı	N_a
Pilot ve Kullanılmayan Taşıyıcılar	$N-N_a=64, 16$
Modülasyon türü	BPSK, QPSK, 8 PSK
Sembol İletim Süresi	T_s
Guard (koruma) Süresi	$T_G=T_s/4$
İşlem Süresi (Frame Process Time)	$T_w= T_s+ T_G$
İki komşu taşıyıcının arası	$1/ T_s$
Bant Genişliği	N/ T_s
Bir Pencere Süresinde iletilen bit	nN_a
Örnek Veri iletim Hızı	nN_a/ T_w [bps]
Sönümlenme modeli	Manuel olarak değiştirilebiliyor.
Doppler kayması	Manuel olarak değiştiriliyor.
Kanal kestirimi	Parametreleri bilinmeyen kanal, Pilot Sembol, Parametreleri bilinen kanal,

Tablo 3.1: OFDM parametreleri

Var olan OFDM Parametreleri belirlenirken, toplam taşıyıcı sayısı sabit tutulmasına rağmen, istendiğinde taşıyıcı sayısı değiştirilebilecek şekilde programlama yapılmış olup, Pilot Sembol sayısı, modülasyon türü, sönümlenme modeli, doppler kayması etkisi ve kanal kestirimi değişken parametreler olarak belirlenmiştir.

OFDM haberleşme benzetiminde taşıyıcı sayısı ve “Cyclic Prefix” uzunluğu sabit tutulmuştur. LDPC kodlama da ise Gb (302X1200) ve Hb (900x1200) matrisleri kullanılmıştır. Multilevel LDPC kullanıldığı için OFDM ortamına gönderilen LDPC bitleri 4/8 PSK ile modüle edilerek oluşturulmuştur. Böylece İki ve Üç Seviyeli LDPC OFDM kodlama elde edilmiştir.

4.3.2 Programlama Akışı

Aşağıda verilen program akışında, önceden belirtilmiş olan parametreler sabit tutularak, yalnızca bir senaryoya yönelik kod satırları görülmektedir.

Benzetimde ilk adım olarak global değişkenler tanımlanmıştır.

```

2 - clear all
3
4 - global Gb Hb
5 - load gb;
6 - load hb;
7
8 - [input output]=size(Gb);

```

Daha sonra parametreler tanımlanmıştır. Her test başlangıcında etkisi incelenen parametreler (P ve SNR) manuel olarak değiştirilerek sonuçlar alınmıştır.

```

10 - SNR=9;
11 - SNRv=10^(SNR/10);
12 - sigma=1/sqrt(1*SNRv/(4));
13
14 - EbN0dB=10*log10(10^(SNR/10)/(2*input/output));
15
16
17 - ofdmFrame=128;
18 - pilot=16;
19 - infoSymbol=ofdmFrame-pilot;
20 - ofdmFrameNo=floor(output/infoSymbol);
21
22 - error1top=0;
23 - error2top=0;
24 - error3top=0;

```

Parametreler belirlendikten sonra 4 PSK Sembolleri üretilmiştir.

```

26 - % 4PSK sembolleri
27 - s0=1; %00
28 - s1=i; %01
29 - s2=-1; %10
30 - s3=-i; %11

```


İterasyon sayısı, rastgele bit üretimi ve LDPC matris eşleştirme yapılmıştır.

```

32 - for iter=1:1000000
33
34 -     x1 = round(rand(1,input)); % random bits
35 -     x2 = round(rand(1,input)); % random bits
36
37 -     y1 = mod(x1*Gb,2);           % coding
38 -     y2 = mod(x2*Gb,2);           % coding
39
40 -     y=[y1;y2]';

```

Daha önce elde edilen OFDM kanal benzetimine ait alt program çağrılarak elde edile semboller OFDM kanalına gönderilerek OFDM frame'ler üretilmiştir.

```

42 -     for k=1:ofdmFrameNo
43 -         [data0 data1 data2]=ofdm_channel_4psk(y((k-1)*infoSymbol+1:k*infoSymbol,:),ofdmFrame,pilot,SNR)
44 -         recTam0((k-1)*infoSymbol+1:k*infoSymbol)=data0;
45 -         recTam1((k-1)*infoSymbol+1:k*infoSymbol)=data1;
46 -         recTam2((k-1)*infoSymbol+1:k*infoSymbol)=data2;
47 -     end
48 -     kalan=output-ofdmFrameNo*infoSymbol;
49 -     sonFrame=[y(ofdmFrameNo*infoSymbol+1:end,:); zeros(infoSymbol-kalan,2)];
50
51 -     [data0 data1 data2]=ofdm_channel_4psk(sonFrame,ofdmFrame,pilot,SNR);
52 -     recKalan0=data0(1:kalan);
53 -     recKalan1=data1(1:kalan);
54 -     recKalan2=data2(1:kalan);
55
56 -     z0=[recTam0 recKalan0];
57 -     z1=[recTam1 recKalan1];
58 -     z2=[recTam2 recKalan2];
59
60 -     r=z0;

```

Çağrılan alt programda öncelikle tanımlamalar yapılmıştır.

```

1 - function [est_data0noP est_data1noP est_data2noP]=ofdm_channel_4psk(v,K,P,SNR)
9 - Lcp = 32;           % length of the Cyclic Prefix
10 - N = K + Lcp;       % total duration of one OFDM symbol
12 - S = K/P;           % space between adjacent pilot symbols
19 - error1top=0;
20 - error2top=0;
21 - error3top=0;

```

Alt programda Pilot Semboller veri dizinine işlenmiştir.

```

26 - data=bi2de(v,'left-msb');
27 - org_data=pskmod(data,4);
28
29 - t=0;
30 - for i=0:K-1
31 -     if mod(i,S)==0
32 -         cod_data(i+1)=1; % inserting pilots
33 -     else
34 -         t=t+1;
35 -         cod_data(i+1)=org_data(t);
36 -     end
37 - end

```

Daha sonra OFDM sembol üretilmiştir.

```

40
41     % Generating one OFDM Symbol
42
43 - S1 = ifft(cod_data.',K)./sqrt(K);
44 - S2 = [S1(K - Lcp + 1 : K).' S1.'];

```

Bir sonraki aşamada kanal kodlaması yapılmıştır. Bir kanalda beş ayrı (beş tap) yansımanın olduğu ve her yansıyan sinyalin parametrelerinin manuel olarak değiştirilme imkanı sağlanmış, denemelerle optimum katsayılar belirlenmiştir. Ayrıca kanaldaki her yansımanın maruz kalacağı gecikme ve doppler etkisinin de değiştirilmesi olanağı sağlanmıştır.

```

48     % Channel Simulation
49
50 - L=5;
51 - alpha = [1 0.55 0.25 0 0]';
52 - delay = [0 4 8 0 0]';
53 %doppler = [0 0 0 0 0]';
54 - doppler = [0.001*pi 0.001*pi 0.001*pi 0 0]';
55
56 - m = [0 : N-1];
57 - m = [0 : N-1];
58 - y1 = alpha(1) * exp(j * doppler(1) .* m) .* [zeros(1,delay(1)) S2(1:N-delay(1))];
59 - y2 = alpha(2) * exp(j * doppler(2) .* m) .* [zeros(1,delay(2)) S2(1:N-delay(2))];
60 - y3 = alpha(3) * exp(j * doppler(3) .* m) .* [zeros(1,delay(3)) S2(1:N-delay(3))];
61 - y4 = alpha(4) * exp(j * doppler(4) .* m) .* [zeros(1,delay(4)) S2(1:N-delay(4))];
62 - y5 = alpha(5) * exp(j * doppler(5) .* m) .* [zeros(1,delay(5)) S2(1:N-delay(5))];
63 - Y = y1 + y2 + y3 + y4 + y5;
64
65
66
67 - SNRW=10^(SNR/10);
68 - varS2=var(S2);
69 - varN=varS2/SNRW/2;
70 - noise=sqrt(varN)*randn(1,N)+j*sqrt(varN)*randn(1,N);
71 - Rec=Y+noise;

```

Alıcı tarafından demodülasyon işleminin başlaması için veri içermeyen “Cyclic Prefix” bilgileri alınan veriden çıkartılmış ve demodülasyon işlemi gerçekleştirilmiştir.

```

75      % Demodulation
76
77 -    Rec2 = Rec(Lcp+1:N); %cyclic prefix is removed
78
79 -    Rec3 = fft(Rec2,K).*sqrt(K); % OFDM symbol is demodulated

```

Parametreleri bilinmeyen kanal, Pilot Sembol ekleme ve parametreleri bilinen kanallardan OFDM sembolleri geçirilerek sonuçlar elde edildikten sonra tekrar üst programa geçilerek, sınırlamalar ile birlikte, Öklid uzaklıkları yardımıyla, sembollerin olası değerleri elde edilmiştir.

```

62 -    for dongu2=1:size(Hb,2) % Oklid uzakliklari bulunuyor
63 -        dis1(dongu2,:) = abs((r(dongu2)-s0).^2);
64 -        dis2(dongu2,:) = abs((r(dongu2)-s1).^2);
65 -        dis3(dongu2,:) = abs((r(dongu2)-s2).^2);
66 -        dis4(dongu2,:) = abs((r(dongu2)-s3).^2);

82 -        prob_sifir1(dongu2)=1/2*(1/dis1(dongu2)+1/dis2(dongu2)); %1. bitin sifir olma olasiligi
83 -        prob_bir1(dongu2)=1/2*(1/dis3(dongu2)+1/dis4(dongu2)); %1. bitin bir olma olasiligi
84
85 -        prob_sifir2(dongu2)=1/2*(1/dis1(dongu2)+1/dis3(dongu2)); %2. bitin sifir olma olasiligi
86 -        prob_bir2(dongu2)=1/2*(1/dis2(dongu2)+1/dis4(dongu2)); %2. bitin bir olma olasiligi
87
88 -        rs1(dongu2)=1- 2*prob_bir1(dongu2)/ (prob_bir1(dongu2)+prob_sifir1(dongu2)); %1. bitin normalize
89 -        rs2(dongu2)=1- 2*prob_bir2(dongu2)/ (prob_bir2(dongu2)+prob_sifir2(dongu2)); %2. bitin normalize
90 -        rs=[rs1 rs2];

142 -    f01=1./ (1+exp(-2*rs1/sigma^2)); % likelihoods (Gauss)
143 -    f11=1-f01;
144
145 -    f02=1./ (1+exp(-2*rs2/sigma^2)); % likelihoods
146 -    f12=1-f02;

```

LDPC Decoder çıktısı, sembolü oluşturan her bit çıkış değeri için elde edilmiştir.

```

148 - [z_hat1, success, k] = ldpc_decode(rs1,f01,f11,Hb);% 1. decoder ciktisi (x1 icin)
149 - [z_hat2, success, k] = ldpc_decode(rs2,f02,f12,Hb);% 2. decoder ciktisi (x2 icin)
150
151 - x_hat1 = z_hat1(size(Gb,2)+1-size(Gb,1):size(Gb,2));%1. sistematik datayi elde ettik pariteleri attik
152 - x_hat1 = x_hat1'; %Transpoze aldik
153 - x_hat2 = z_hat2(size(Gb,2)+1-size(Gb,1):size(Gb,2));%2. sistematik datayi elde ettik pariteleri attik
154 - x_hat2 = x_hat2'; %Transpoze aldik
155 - err1(iter)=length(find(x_hat1~=x1))+length(find(x_hat2~=x2));
156
157
158 - r=z2;

```

İterasyonlar sonucunda parametreleri bilinmeyen, Pilot Sembol eklenen ve parametreleri bilinen kanallara ait hata değerleri elde edilmiştir.

```

206 - error1top=error1top+err0(iter);
207 - error2top=error2top+err1(iter);
208 - error3top=error3top+err2(iter);
209
210 - BER1=error1top/(iter*2*input);
211 - BER2=error2top/(iter*2*input);
212 - BER3=error3top/(iter*2*input);

```

Elde edilen hata değerleri her iterasyonda ekrana yansıtılarak, sonuçlar grafiklere manuel olarak işlenmiştir.

```

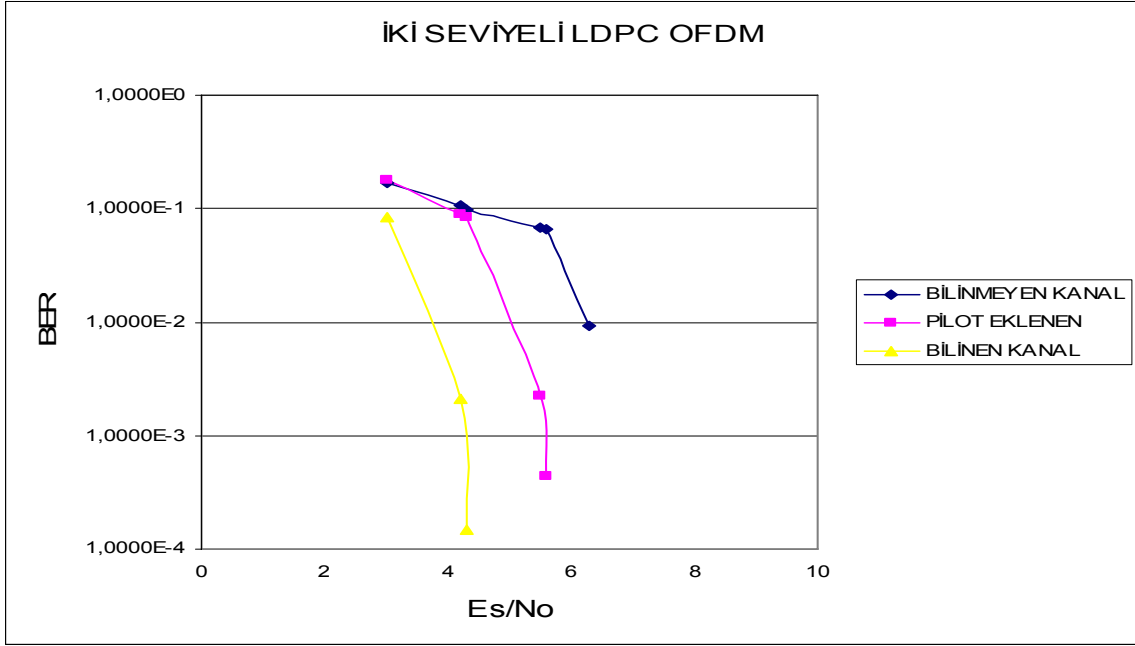
214 - fprintf('\n*****\n');
215 - fprintf('SNR=%4f, Eb/NO=%4f \n',SNR,EbNOdB);
216 - fprintf('%d frame is transmitted \n', iter);
217 - fprintf('No Channel State Information\n');
218 - fprintf('BER1=%4.5e\n',BER1);
219 - fprintf('Pilot Assisted Channel State Information\n');
220 - fprintf('BER2=%4.5e\n',BER2);
221 - fprintf('Perfect Channel State Information\n');
222 - fprintf('BER3=%4.5e\n',BER3);
223
224 - end

```

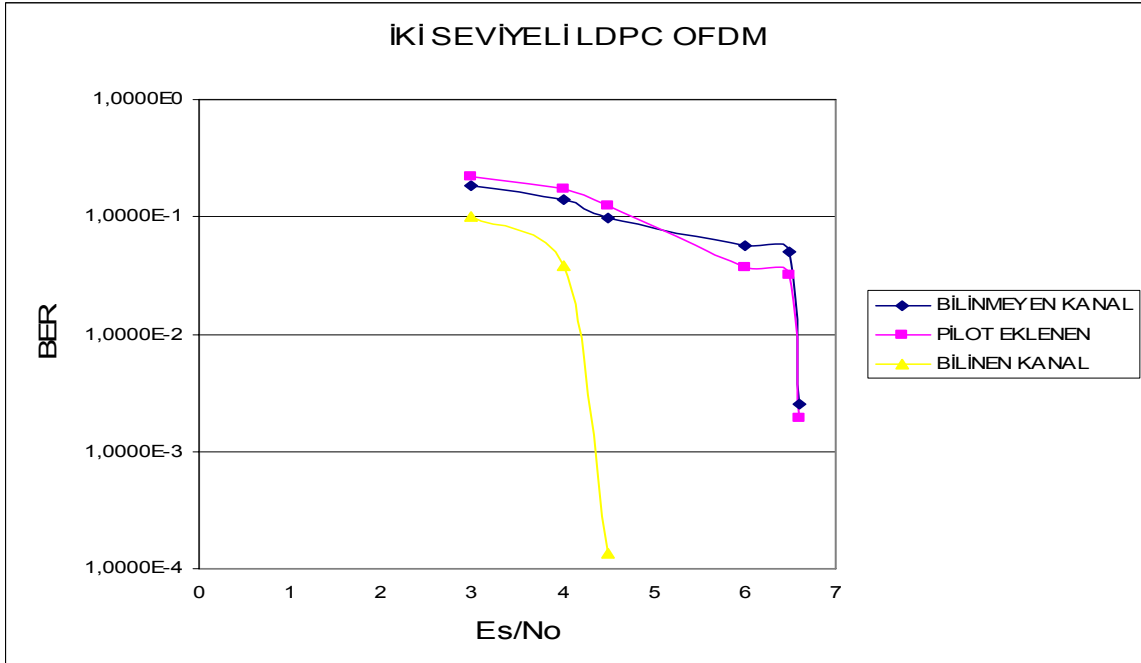
4.3.3. Benzetim Sonucunda Elde Edilen Grafikler

Benzetim sonucunda elde edilen sonuçlar değerlendirilerek aşağıdaki grafiklerde gösterilmiştir.

İki Seviyeli LDPC OFDM (Pilot Sembol Etkisi)



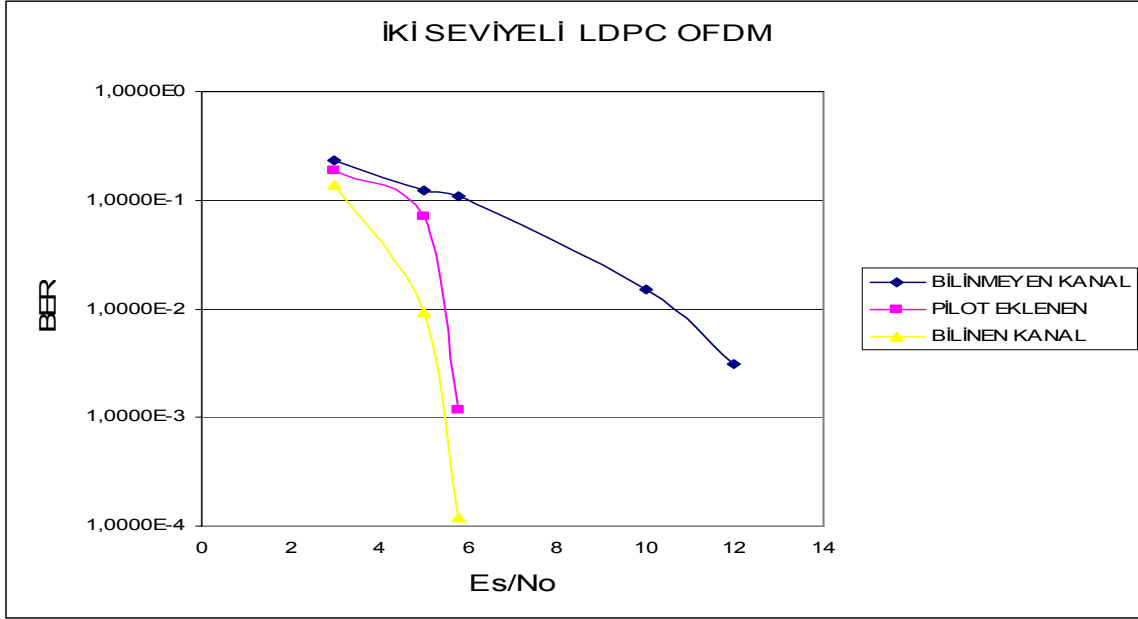
N=128	KANAL	1	0.55	0.25
P=64	GECİKME	0	4	8
CP:32	DOPPLER	0	0	0



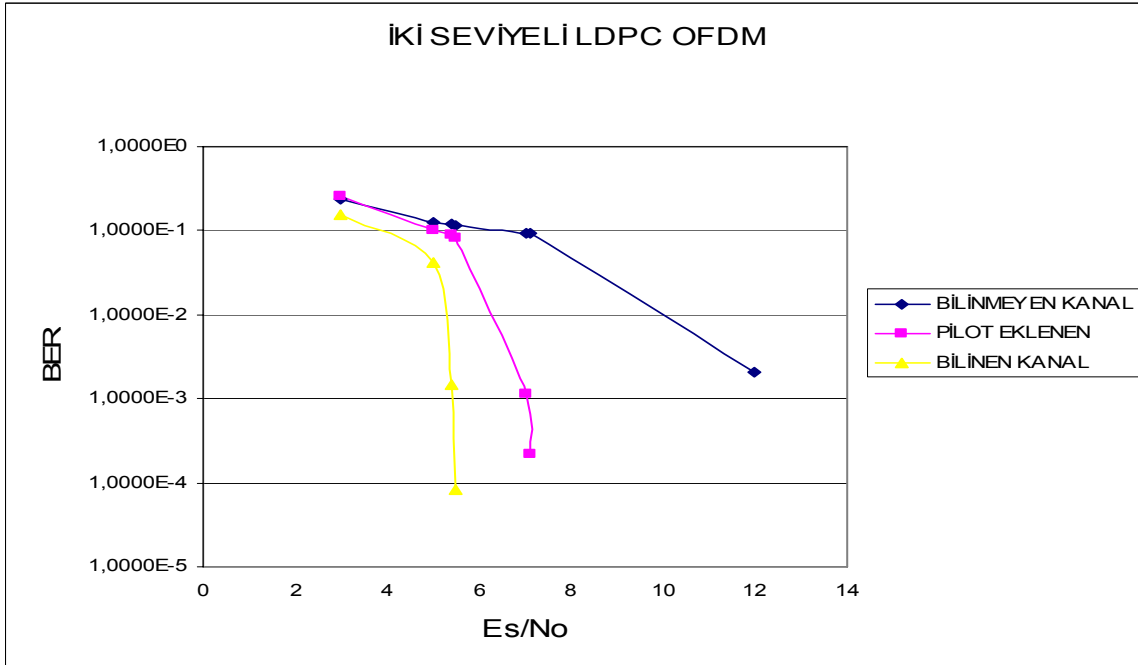
N=128	KANAL	1	0.55	0.25
P=16	GECİKME	0	4	8
CP:32	DOPPLER	0	0	0

Şekil 3.32: İki seviyeli LDPC kodlamalı OFDM'e Pilot Sembol etkisi

İki Seviyeli LDPC OFDM (Doppler Etkisi)



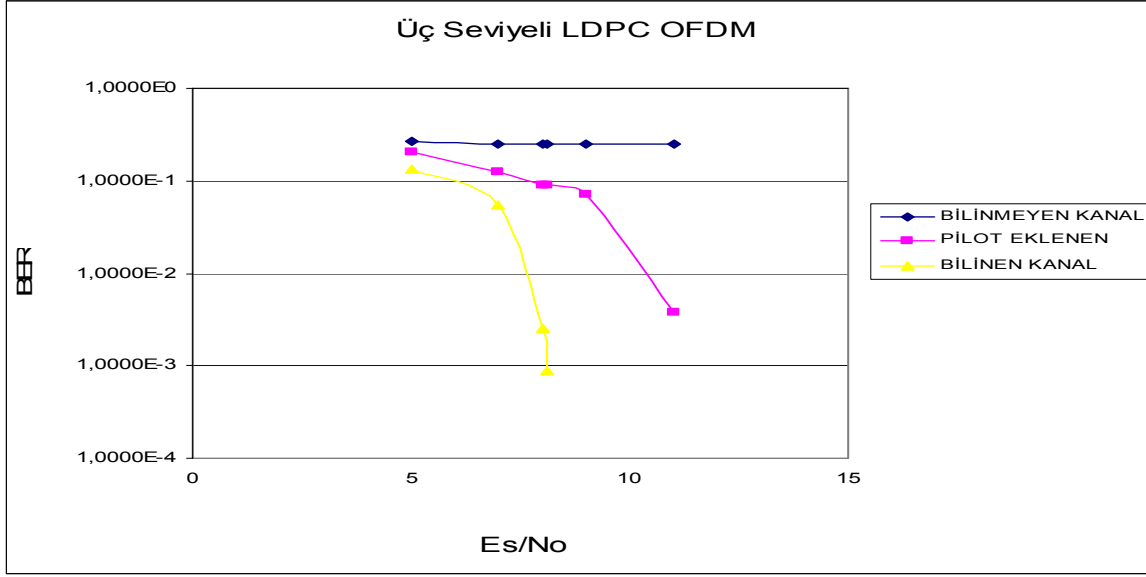
N=128	KANAL	1	0.55	0.25
P=64	GECİKME	0	4	8
CP:32	DOPPLER	0.0001*pi, 0.0001*pi, 0.0001*pi		



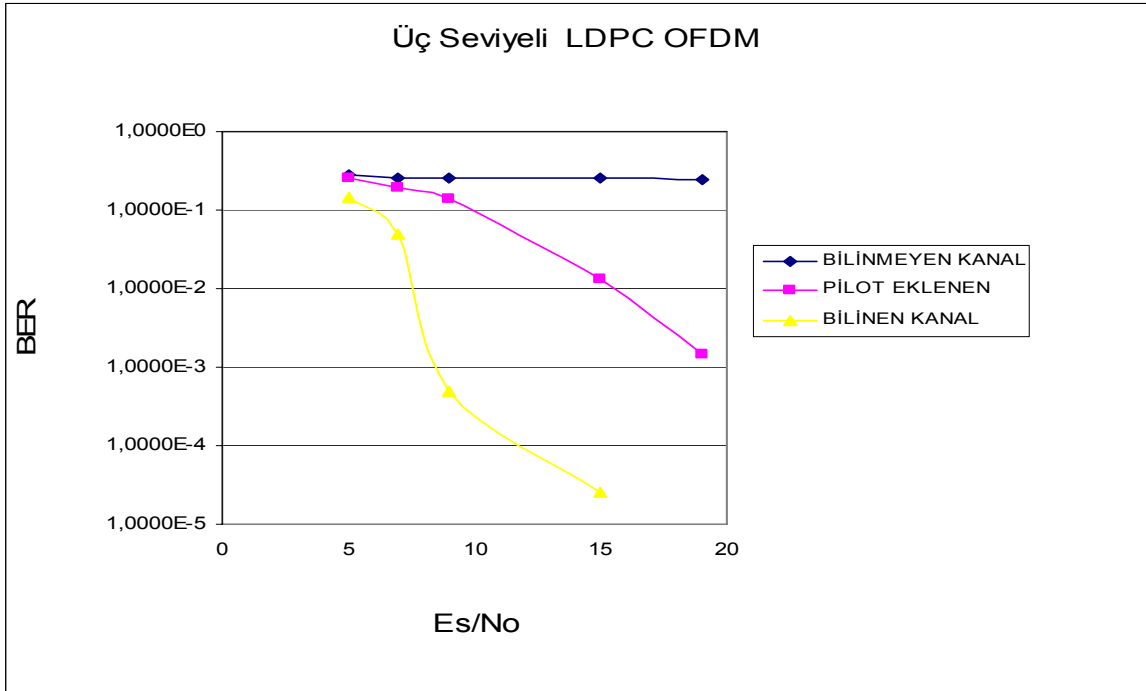
N=128	KANAL	1	0.55	0.25
P=16	GECİKME	0	4	8
CP:32	DOPPLER	0.0001*pi, 0.0001*pi, 0.0001*pi		

Şekil 3.33: İki seviyeli LDPC kodlamalı OFDM'e Doppler Etkisi

Üç Seviyeli LDPC OFDM (Pilot Sembol Etkisi)



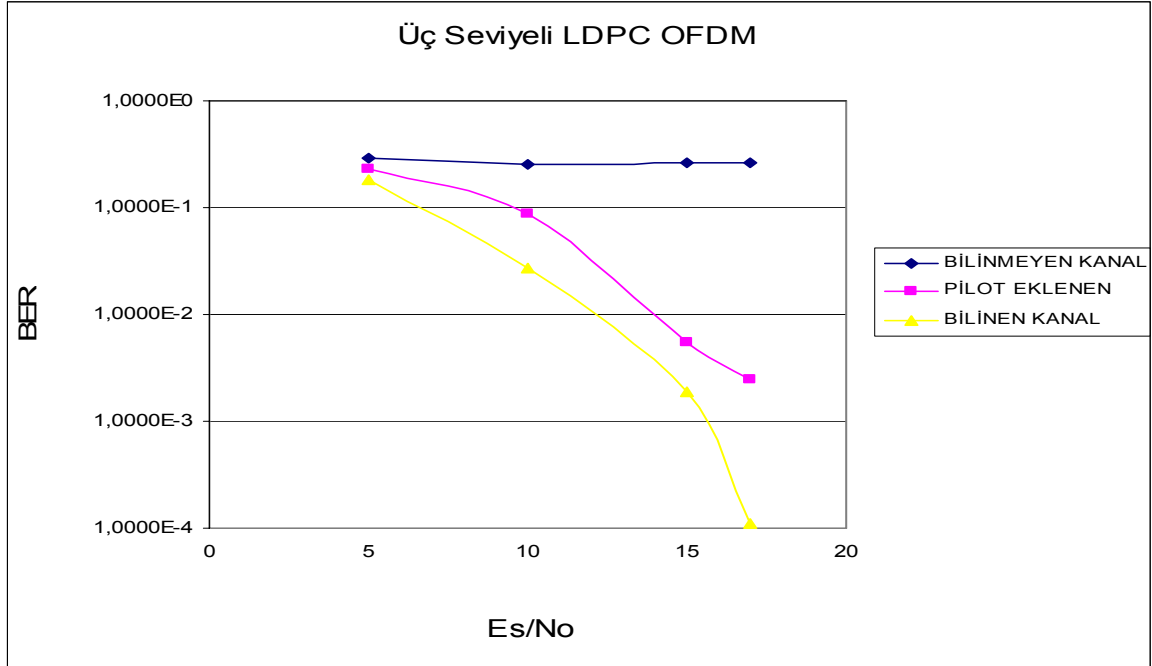
N=128	KANAL	1	0.55	0.25
P=64	GECİKME	0	4	8
CP:32	DOPPLER	0	0	0



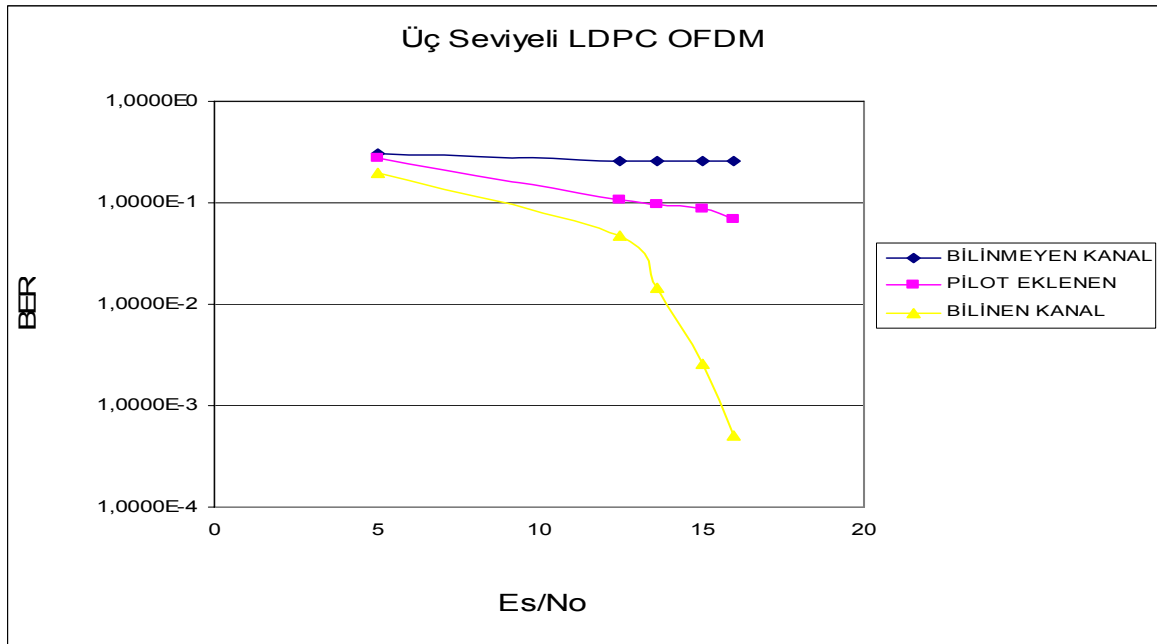
N=128	KANAL	1	0.55	0.25
P=16	GECİKME	0	4	8
CP:32	DOPPLER	0	0	0

Şekil 3.34: Üç seviyeli LDPC kodlamalı OFDM'e Pilot Sembol Etkisi

Üç Seviyeli LDPC OFDM (Doppler Etkisi)



N=128	KANAL	1 0.55 0.25
P=64	GECİKME	0 4 8
CP:32	DOPPLER	0.0001*pi, 0.0001*pi, 0.0001*pi



N=128	KANAL	1 0.55 0.25
P=16	GECİKME	0 4 8
CP:32	DOPPLER	0.0001*pi, 0.0001*pi, 0.0001*pi

Şekil 3.35: Üç seviyeli LDPC Kodlamalı OFDM'e Doppler Etkisi

4.3.4. Elde Edilen Bulgular

Çok seviyeli kodlama ve PSK Modülasyonu kullanılarak aynı veriyi paralel kollarda sembol haline getirerek daha hızlı iletme imkanı sağlanmaktadır.

Şekil 3.32’de görüleceği gibi, iki Seviyeli LDPC Kodlamalı OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve “Cyclic Prefix” sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenmemiştir. Bu durumda en iyi sonuç, beklendiği gibi parametreleri bilinen kanalda işlem yapılması durumunda elde edilmiştir. Tek başına LDPC kodlama ya da tek başına OFDM tekniği kullanılması ile edilen hata başarımlarına göre, Çok seviyeli LDPC Kodlamalı OFDM yapısının kullanılması durumunda, hata başarımında dikkat çekici seviyede iyileşme gözlenmiştir. Hatta parametreleri bilinmeyen kanal kullanılması durumunda bile hata başarımı çok iyi seviyeye getirilmiştir. Pilot Sembol sayısının arttırılması ile sonuçların daha da iyileştiği gözlenmiştir.

Şekil 3.33’de görüleceği gibi, iki Seviyeli LDPC Kodlamalı OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve “Cyclic Prefix” sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenmiştir. Bu durumda en iyi sonuç, beklendiği gibi parametreleri bilinen kanalda işlem yapılması durumunda elde edilmiştir. Doppler etkisi nedeniyle genel sonuçlarda kötüleşme görülmesine rağmen hata başarımında dikkat çekici seviyede iyileşme gözlenmiştir. Doppler etkisinin, pilot Sembol sayısının arttırılması ile azaldığı gözlenmiştir.

Şekil 3.34’de görüleceği gibi, üç seviyeli LDPC Kodlamalı OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve “Cyclic Prefix” sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenmemiştir. Beklendiği gibi, LDPC seviyesinin artması ile gönderilen veri miktarı arttırıldığından hata başarımında kötüleşme gerçekleşmiştir. Ancak Pilot Sembol sayısının arttırılması başarımlarının iyileştirilmesine yardımcı olmaktadır. Parametreleri bilinmeyen kanal kullanılması durumunda kötü hata başarımı sağlanmıştır. Aynı zamanda daha fazla verinin gönderilmesi için çoklu seviyenin arttırılması ile hata başarımının iyileşmesi ters orantılı olarak gerçekleştirilmiştir.

Şekil 3.35’te görüleceği gibi, üç seviyeli LDPC Kodlamalı OFDM yapısı kullanılmıştır. İletim kanalına gecikme eklenmiş ve “Cyclic Prefix” sayısı sabit tutulmuştur. İletim ortamına Doppler etkisi eklenmiştir. Beklendiği gibi, LDPC seviyesinin artması ile gönderilen veri miktarı arttırıldığından ve Doppler etkisi nedeniyle hata başarımında kötüleşme gerçekleşmiştir. Ancak

Pilot Sembol sayısının arttırılması başarımlarının iyileştirilmesine yardımcı olmaktadır. Pilot sembol sayısının arttırılması hata başarımlarını arttırmıştır.

İki Seviyeli yapıda, Doppler etkisi ile hata başarımlar oranı kötüleşmiştir. Doppler etkisinin olması durumunda, Pilot Sembol sayısının arttırılmasının olumlu katkı yaptığı gözlenmiştir. Girilen Pilot Sembol sayısının azaltılması ile hata başarımlarında kötüleşme gözlenmiştir. Katsayıları bilinen kanal ile en iyi sonuç elde edilmekle beraber katsayıları bilinmeyen kanalda, hata başarımlarının daha kötü olacağı bir gerçektir. Farklı kanal kestirim yöntemi kullanılarak hata başarımlarını arttırılabilir. Ancak bu yöntem karmaşık ve zaman gerektiren bir yöntemdir.

Üç seviyeli LDPC kodlamalı OFDM yapısında, Pilot Sembol eklenmesinin çok iyi sonuçlar verdiği gözlemlenmiştir. Hata Başarımlar oranı, Pilot Sembol sayısının artması ile iyileşmiş, Doppler etkisi ile de kötüleşmiştir. Doppler gibi bir parametrenin sistemde olması durumunda, Pilot Sembol sayısının arttırılmasının olumlu katkı yaptığı gözlenmiştir.

Katsayıları bilinen kanal ile en iyi sonuç elde edilmekle beraber Pilot Sembolün katkısı net olarak gözlemlenmiştir. Pilot Sembol sayısı arttırılarak yapılan haberleşme ve katsayıları bilinen kanal ortamında yapılan haberleşmede hata başarımlarları, paralel seyir izlemiştir. Pilot Sembol olmayan (ya da sayısı azaltılan) ve parametreleri bilinmeyen kanallarda ise hiçbir zaman hata oranı istenen seviyeye gelmemektedir. Pilot Sembol eklenmesi durumunda, hiç bir bilgi içermeyen semboller eklenerek, iletim süresinde bir yavaşlamaya neden olunmakla beraber haberleşme kalitesinin arttırılması açısından bu yapılar tercih edilebilir.

Belli bir SNR değerine ulaşıldıktan sonra hızlı bir eğimle hata oranında iyileşme sağlanmaktadır. Çok küçük sinyal gürültü oranlarında dikkate değer değişimler olabilmektedir. Ayrıca gecikmeli olarak alınan sinyallerin haberleşme kalitesine, ciddi etki edebileceği denemeler yapılarak gözlemlenmiştir. Kodlama türlerine göre en iyi hata başarımlarında iyileşme gözlenmiştir. Hata başarımlarının daha iyi olmasının yanı sıra, çok seviyeli LDPC kodlamalı OFDM tekniği, daha fazla veriyi aynı zamanda gönderme imkanı sağlaması nedeniyle tercih edilebilecektir.

5. TARTIŞMA VE SONUÇ

Bu tezde , “Shannon” Limitlerine yaklařmakta en uygun kodlama tekniđi olarak grlen ve hata dzeltmede ok iyi bir hata bařarım potansiyeline sahip olan LDPC kodlama ile OFDM tekniđine ait benzetimler yapılmıř ve ok seviyeli LDPC kodlamalı OFDM benzetimi yapılarak yeni bir yntem geliřtirilmiřtir.

LDPC benzetim ortamında, kullanılan rete matrisinin ve grlt ortamının karmařıklıđına gre literatr taramalarında elde edilen performanslar ile karřılařtırıldıđında daha iyi sonular elde edilmiřtir. Benzetim ortamında drt farklı grlt ortamı (AWGN, Rayleigh, Rician10, Rician20) ve  farklı rete matrisi kullanılmıřtır. İki seviyeli LDPC kodlamada en iyi hata bařarımının Gb rete matrisinin kullanılması durumunda ve  seviyeli LDPC kodlamada ise en iyi hata bařarımının SED rete matrisi kullanılması halinde elde edildiđi grlmřtr. Ga rete matrisi daha kk boyutta olmasına rađmen hata bařarımında en kt sonu alınmıřtır.

Paralel veri transferinin seviyesinin (iki seviyeden  seviyeye) arttırılması halinde, farklı rete matrislerine bađlı kalınmadan, sistem performansının greceli olarak dřtđ gzlenmiřtir. Transfer edilmek istenen verinin zelliđine gre daha kısa zamanda daha ok veri gnderilmek istendiđinde ok seviyeli olarak kodlanması ayrı bir yntemdir. Bu durumda, gnderilen veriye ait hata bařarımı gz ardı edilmemelidir. Gnderilecek verinin hata bařarım seviyesinden dn verilemiyor ise bu durumda daha dřk seviyeli kodlama yapılması gerektiđi belirlenmiřtir. nemli olan, uygun bir hata olasılıđı iin, bulunan sinyal grlt oranına bađlı olarak seviye sayısını belirlemektir. Diđer bir deyiřle hızı belirlemektir.

LDPC kodlamada farklı grlt ortamları birbirleriyle karřılařtırıldıđında en iyi hata bařarımının, AWGN grlt ortamında alındıđı grlmřtr. Bu beklenen bir durumdur. nk diđer grlt ortamlarına nazaran bu grltnn en az karmařıklıđa sebep olduđu dolayısıyla iletim ortamında gnderilen veriyi en az etkileyeceđi aıktır.

LDPC kodlamada genel olarak; kodlama hata başarımının üreteç matrislerine bağlı olduğu (üreteç matrislerinin büyük olması iyi sonuçlar verebiliyor iken daha küçük üreteç matrislerinden daha iyi sonuçlar alınamayabilmektedir) gözlenmiştir. Ayrıca, gürültü ortamının karmaşıklığına göre gönderilmek istenen veri kalitesinin değiştiği de gözlemlenmiştir.

Sayısal haberleşme teknikleri, değişken koşullara sahip bir ortamda sınırlı sembolleri ilettikleri için sorun yaratan etkiler (iç/dış gürültüler, iletim kanalı yapısı, yansımalar v.b.) nedeniyle oluşan bozulmaların hata düzeltmeli kodlama yöntemleri (error correction) ile düzeltilebilmesine, düzeltilemeyenlerin olumsuz etkilerinin mümkün olduğunca azaltılmasına (error concealment) imkan verirler. Bu koşullar altında, sayısal kodlanmış veriyi iletmek için tamamen yeni teknikler geliştirmek zorunlu hale gelmiş ve tüm veri iletim ihtiyaçlarının karşılanması için kuvvetli bir sayısal modülasyon yönteminin verimli bir temel band kodlamasıyla birlikte kullanılması ile çözüm sağlanabilmiştir.

Hızlı teknolojik gelişmeler ile, geniş kitlelere sayısal yayın mümkün ve ekonomik hale gelmiş, sayısal iletim yöntemleri yaygın şekilde uygulanmaya başlanmıştır. Farklı sayısal iletim kodlama ve modülasyon yöntemleri olmakla beraber, son yıllarda dikey frekans çoğullamalı modülasyon yöntemi (OFDM) önem kazanmış ve tek frekans ağlarına imkan tanınması nedeniyle gün geçtikçe tercih edilen bir teknik durumuna gelmiştir.

Günümüzde, çok taşıyıcılı yöntemlerin başında OFDM gelmektedir. OFDM yapılarında, hata düzeltme algoritmaları kullanılarak daha başarılı sonuçlar alınabileceği varsayıldığından, buna yönelik çalışmalar yapılmıştır. Güçlü hata düzeltme kodlama algoritmaları kullanıldığında, sönümlenmiş alt bantlar yerine diğer alt bantlar ile birlikte karşı tarafa aktarılan ortalama güç oranının referans olarak alınmasıyla, sönümlemeye uğramış alt bantların yanıtıcı etkisi yok edilebilmektedir

OFDM sistemler; çoklu yol bozulmasına karşı dayanıklı oluşu, iletim kanalının frekans spektrumunun ufak parçalara ayrılarak frekansla değişmez hale getirmesi, semboller arası girişimi engelleyebilmesi, daha basit kanal eşitleyicilere ihtiyaç duyması, frekans seçici sönümlemeli kanallara ve dar-band bozulmasına karşı sağlamlığı, farklı hata düzeltme kodları kullanarak daha güvenli iletim sağlayabilmesi nedeniyle tercih edilen bir yapı olarak geniş bir kullanım alanı bulmuştur. OFDM tek bir data dizisinin belli bir sayıdaki alt-taşıyıcılarla iletildiği çok-taşıyıcılı

iletimin özel bir halidir. Bir modülasyon çeşidi gibi görülebildiği gibi aynı zamanda çoklayıcı tekniğidir.

OFDM tekniği, Band genişliği verimi, sistem karmaşıklığı ve spektrum biçimlendirme gibi bir çok tasarım parametresini karşılamada oldukça esnektir. Ani kanal etkilerine karşı herhangi bir adaptasyon gerektirmemekte ve darbe girişimi ile kanal içi değişimlere karşı dayanıklılığı arttırmaktadır. Seri iletim tekniklerine göre örnekleme zamanında meydana gelen değişmelere karşı daha az duyarlıdır. Uygun kodlanması durumunda çok iyi bir hata başarımla potansiyeline sahiptir.

Bu tez çalışmasının bir parçası olarak hazırlanan OFDM benzetim ortamında, sisteme gecikme ve gürültü eklenip diğer OFDM parametreleri sabit tutularak (Cyclic Prefix, Taşıyıcı Sayısı v.b.) Doppler etkisi ile Pilot Sembol sayısının değişiminin etkileri gözlenebilmiştir.

BPSK modüleli OFDM yapısında, Pilot Sembol sayısının artmasının hata başarımla kötüleştirdiği gözlenmiştir. Çünkü, BPSK yapısında gönderilen bit sayısının az olması ve Pilot Sembolün bu bitler ile kıyaslanabilir seviyede olması nedeniyle, kod kelimesinde bilgi bitlerinin oranı azalmaktadır. Ancak doppler etkisi eklenmesiyle kötüleşen hata başarımla oranının Pilot Sembol etkisi ile iyileştiği gözlenmiştir.

QPSK modüleli OFDM yapısında, hata başarımla daha iyi olduğu, Pilot Sembolün artması ile hata başarımla daha da düzeldiği ve Doppler etkisinin beklendiği gibi etki yaptığı gözlenmiştir. QPSK ile, aynı zaman aralığında daha fazla veri gönderilmesine rağmen hata başarımla sonuçlarında gözle görünür bir düşme gözlenmemiştir.

8PSK modüleli OFDM yapısında, genel olarak hata başarımla bir kötüleşme gözlenmiş olup parametreleri bilinmeyen kanal kullanılması durumunda neredeyse alıcı tarafında hiç veri alınamadığı tespit edilmiştir. Pilot Sembol sayısının artırılması ile iyi sonuçlar alınmış olup neredeyse parametreleri bilinen kanal kullanılması durumunda elde edilen hata başarımla değerlerine yakın sonuçlar alınmıştır. Modülasyon derecesinin artması durumunda, Doppler etkisi sonucu kötüleştirmekte ancak Pilot Sembol sayısının artırılması hata başarımla iyileştirmektedir.

OFDM benzetimi sonucunda; Doppler etkisinin hata başarımında kötüleşmeye sebep olduğu, PSK derecesinin artırılmasının, gönderilen veri miktarını arttırmasına rağmen hata başarımını kötüleştirdiği ancak Pilot Sembol sayısının arttırılmasının hata başarımını arttırdığı net bir şekilde tespit edilmiştir. Elde edilen sonuçlar kullanılarak, gerektiğinde uygun parametre tanımlaması ve kanal modeli oluşturulabilecektir.

Farklı koşullara sahip bir ortamda sınırlı sembolleri ilettikleri için sorun yaratan etkiler nedeniyle, hata düzeltmeli kodlama yöntemleri ile olumsuz etkilerin mümkün olduğunca azaltılmasına imkan veren sayısal haberleşme teknikleri ile iletim kalitesi gün geçtikçe artmaktadır. Sinyal iletim ortamının zayıflama karakteristiği sürekli bir durumdur. Sayısal iletimde, vericiden uzaklaştıkça alıcıdaki sinyal kalitesinde de zayıflama, sürekli ve uzaklık ile orantılı olarak değişme sözkonusudur. Ancak, farklı teknikler kullanılarak gerçekleştirilebilen, sinyal-gürültü oranındaki iyileştirmeler nedeniyle daha iyi sonuçlar elde edilebilmektedir. Kullanılan sayısal yöntemlere göre aynı işaretin iletiminde çok farklı parametreler kullanılabilir.

Gönderilmesi gereken veri miktarının artması ile kısıtlı bant genişliğine sahip olunması nedeniyle, haberleşme sistemlerinde yeni gelişmeler olma zorunluluğu ortaya çıkmıştır. Farklı modülasyon tekniklerinin yanısıra, daha fazla verinin kısıtlı bant aralığında en az hata ile gönderilmesi için verinin paralel kollarda (çok seviyeli) kodlanarak gönderilmesi ile zaman tasarrufu sağlanabilecektir.

LDPC kodlama ve OFDM sistemlerin avantajları ile birlikte kullanılarak, diğer yapılara göre daha hızlı verinin, daha güvenilir olarak gönderilebilmesine olanak sağlanmaktadır. LDPC kodlamalı OFDM yapısı önceden bilinmesine rağmen, Çok Seviyeli LDPC ile kodlanması konusuna literatürde rastlanmadığından gelişmeye açık bir konu olduğu değerlendirilmektedir.

Bu tez çalışması ile hazırlanan, LDPC kodlamalı OFDM benzetim ortamında, her iki yapının kullanılması ile hata başarımının değişiminin gözlenebilmesinin yanı sıra, bu birleşik yapıda Doppler ve Pilot Sembol kullanılması durumunda nasıl bir etki oluşacağını gözlenmesi amaçlanmıştır.

İki Seviyeli LDPC Kodlamalı OFDM benzetim ortamında, iletim kanalına gecikme eklenmiş ve “Cyclic Prefix” sayısı sabit tutulmuştur. Bu durumda en iyi sonuç, beklendiği gibi parametreleri bilinen kanalda işlem yapılması durumunda elde edilmiştir. Tek başına LDPC kodlama ya da tek başına OFDM tekniği kullanılması ile edilen hata başarımlarına göre, Çok Seviyeli LDPC Kodlamalı OFDM yapısının kullanılması durumunda, hata başarımında dikkat çekici seviyede iyileşme gözlenmiştir. Hatta, parametreleri bilinmeyen kanal kullanılması durumunda bile hata başarımı çok iyi seviyeye getirilmiştir. Pilot Sembol sayısının artırılması ile sonuçların daha da iyileştiği ve negatif yöndeki Doppler etkisini azalttığı gözlenmiştir.

Üç seviyeli LDPC Kodlamalı OFDM yapısında, beklendiği gibi, doppler etkisi ya da LDPC seviyesinin artması ile gönderilen veri miktarı daha da artırılmış olduğundan, hata başarımında kötüleşme gözlenmiştir. Ancak Pilot Sembol sayısının artması ile, hata başarım sonuçlarının iyileştirilmesine yardımcı olunmaktadır. Doppler gibi bir parametrenin sistemde olması durumunda, Pilot Sembol sayısının artırılmasının olumlu katkı yaptığı gözlenmiştir. Daha fazla verinin gönderilmesi için çoklu seviyenin artırılması ile hata başarımı düşmektedir.

Benzetim ortamı ile yapılan değerlendirmeler sonucunda elde edilen grafiklerden de görüleceği gibi, belli bir SNR değerine ulaşıldıktan sonra hızlı bir eğimle hata oranında iyileşme sağlanmaktadır. Çok küçük değerdeki sinyal gürültü oranlarında dikkate değer değişimler olabilmektedir. Ayrıca gecikmeli olarak alınan sinyallerin (yansıyan sinyaller) haberleşme kalitesine ciddi etki edebileceği denemeler yapılarak gözlemlenmiştir.

Katsayıları bilinen kanal ile en iyi sonuç elde edilmekle beraber, Pilot Sembolün katkısı net olarak gözlenebilmiştir. Pilot Sembol sayısı artırılarak yapılan haberleşme ve katsayıları bilinen kanal ortamında yapılan haberleşmelere ait hata başarım oranları paralel seyir izlemektedir. Pilot Sembol olmayan (ya da sayısı azaltılan) ve parametreleri bilinmeyen kanallarda ise hiçbir zaman hata oranı istenen seviyeye gelmemektedir.

Bu tez çalışması ile elde edilen başarılı sonuçları ile birlikte, Çok Seviyeli LDPC kodlamalı OFDM kodlamanın, haberleşme literatüründe ilk defa kullanılması nedeniyle, ilgi uyandıran bir çalışma olarak görülebileceği değerlendirilmektedir.

KAYNAKLAR

- [1] **A. SHOKROLLAHI**, April 2, 2003 “LDPC Codes:An Introduction”, Digital Fountain, Inc.39141 Civic Center Drive, Fremont.
- [2] **J.WU, H.N.Lee** , 2001 “ Best applying for LDPC coded modulation on SISO, MIMO and MAC channels”, University of pitsburg publishes.
- [3] **R.G. GALLAGAR** , Jan 1962, “ Low Density Parity Check Codes”, IRE Trans. On. Inform. Theory, Vol. IT-8, pp. 21-28.
- [4] **R.G. GALLAGAR** ,1963, “ Low Density Parity Check Codes”, no.21 in Resarch Monographs Series Cambridge, MA:MIT press.
- [5] **D.J.C MACKAY and R.M.NEAL**, Aug.1996, “Near Shannon Limit Performance of Low Density Parity Check Codes”, Electron. Lett.,Vol.32, no.18, pp. 1645-1646.
- [6] **M. KARKOOTI, P. RADOSAVLJEVIC and J.R. CAVALLARO**, 2006 “Configurable, High Throughput, Irregular Low Density Parity Check Decoder Architecture: Tradeoff Analysis and Implementation”, IEEE International Conferance on Application Specific Systems, Architectures and Processors.
- [7] **T.J. RICHARDSAN, M.A.SHOKROLLAHI, R.L. URBANKE**, 2001, “Design of capacity approaching irregular low-density parity check codes”, IEEE Trans. Inform. Theory, Vol.47, pp.619-637.
- [8] **C. E. SHANNON**, 1948,“A mathematical theory of communication”, Bell System Technical Journal, vol. 27, pp. 379-423.
- [9] **H. IMAI and S. HIRAKAWA**, 1977, “A new multilevel coding method using error-correcting codes”, IEEE Trans. Inform. Theory, Vol. 23, pp. 371/377.
- [10] **U. WASCHSMANN, R. F. H. FISHER and J. B. HUBER**, 1999, “Multilevel Codes: Theoretical concepts and Practical Design Rules”, IEEE Trans. Inform. Theory, Vol. 45, pp. 1361/1391.
- [11] **T. WUORZ and J. HAGENAUER**, December 1992, “Iterative decoding for multilevel codes using reliability information”, Proc. IEEE Global Telecommun. Conf., pp. 1779{1784, Orlando, FL, USA.
- [12] **K. R. NARAYANAN and J. LI**, July 2000, “Bandwidth Efficient Low Density Parity Check Coding Using Multi Level Coding and Iterative Multi Stage Decoding”, Proc. 2nd Turbo Symp. on Turbo Codes andRelated Topics, pp. 165-168, Brest, France.

- [13] **M. ISAKA and H. IMAI** , May 2001, “On the iterative decoding of multilevel codes”, IEEE Journal on Selected Areas in Communications, Vol. 19, pp. 935/943.
- [14] **M.UYSAL**, Spring 2003, ECE710 “Space Time Coding for wireless Communication lesson handouts”, University of Waterloo.
- [15] **M. PAZARCI**, 2007 ,“Radyo Televizyon Sistemleri”, İTÜ Elektrik Elektronik Fakültesi, ders notları.
- [16] **W. EBERT** , Oct. 1971, “Data Transmission by frequency division multiplexing using the DFT”, IEEE Transactions on Communications Technology, Vol.COM-19, No:15, pp. 628-634.
- [17] **R. R. MOSIER and R. G. CLABAUGH** "Kineplex, a bandwidth-efficient binary transmission system", AIEE Trans, vol. 76, pp. 723-728, Jan. 1958.
- [18] **EUROPEAN TELECOMMUNICATION STANDART**, Nov. 1994, “Radio Broadcast Systems: digital audio broadcasting (DAB) to mobile receiver”, ETSI final draft pr ETS300 401.
- [19] **M.C.D MADDOCKS**, 1993/10, “An introduction to digital modulation and OFDM techniques”, BBC Research Department Report No. RD.
- [20] **E. HAAS**, March 2002, “Aeronatical Channel Modelling”, IEEE Transactions on Vehicular Technology, Vol 51, NO.2.
- [21] **RCTA**, Jan 1993, “Signal in Space”, Aviation System Performance Standarts for VHF digital Communications, Compatibility with Digital Communications Techniques. Draft Doc. 26-93/SC 172-68.
- [22] **G.L STUBER, J.R BARRY, S.W. MCLAUGHLIN, YE LI INGRAM, M.A. PRATT, T.G.**, 2001 “Broadband OFDM wireless communications”, Sch. of Electr. & Comput. Eng., Georgia Inst. of Technol., Atlanta, GA, USA.
- [23] **J. ARMSTRONG**, 2002, La Trobe University Department of Electronic Engineering handouts,.
- [24] **H. FUTAKI, T. OHTSUKI**, 2001, “Low-density parity-check (LDPC) coded OFDM systems”, IEEE Veh. Tech. Conf. Vol. 1, no. 54ND, pp. 82-86.
- [25] **Z. QI, L.HAO, F.GUANZENG**, November 2006, “Performance Analysis of Channel Estimation for LDPC Coded OFDM System In Multipath Fading Channel”, J. Electronics (PRC), Vol.23, No. 6.
- [26] **I. B. DJORDJEVIC and B. VASIC**, 2008 “LDPC Coded OFDM in fiber-optics communication systems [Invited]”, J.Opt.Netw. 7,217-226.
- [27] **K.WITRISAL , Y.H.KIM and R.PRASAD**, 19-22 Sep 1999 “A novel approach for performance evaluation of OFDM with error correction coding and interleaving”, VTC 1999-Fall, IEEE VTS 50th, Vol.1, pp.294-299.

- [28] **Y. XIN, S.A. MUJTABA**, 7-10 Nov. 2004, "A Hybrid decoding Approach for LDPC Coded MIMO-OFDM Systems", Signals Systems and Computers Conference Record of the Thirty-Eight Asilomar Conference on Vol.1, pp.1173-1177.
- [29] **C.BERROU, A.GGLAVIEUX and P. THITIMASHIMA**, May. 1993, "Near Shannon Limit error-correcting coding and decoding: Turbo Codes", in Proc. IEEE International Conference on Comm., pp. 1064-1070.
- [30] **D.J.C. MACKAY**, Mar.1999 "Good error correcting codes based on very sparse matrices", IEEE Trans. Inform. Theory, Vol.45, pp.399-431.
- [31] **H. ZHANG, D. YUAN, F. ZHAO**, 2004, "Performance research between Turbo and LDPC Coded WOFDM on Rayleigh fading channels", Proceedings of second international conference on information technology for application (ICITA).

ÖZGEÇMİŞ

Aziz YILMAZ, 1972 yılında Batman da doğdu. İskenderun/Hatay‘da, Mimar Sinan İlkokulundan 1983 yılında mezun olduktan sonra, orta öğrenimine Payas/Yakacık Ortaokulu'nda devam etti. Payas/Yakacık Ortaokulu ve Lise'sinden mezun olduktan sonra 1990 yılında Anadolu Üniversitesi Mühendislik Fakültesi, Elektronik Mühendisliği Bölümünde lisans öğrenimine başladı. 1991 yılında yatay geçiş yaparak İstanbul Üniversitesi Mühendislik Fakültesi, Elektronik Mühendisliği Bölümünde lisans öğrenimine devam ederek, 1994 yılında dönem birincisi olarak mezun oldu. 2004 yılında Osman Gazi Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Ana Bilim Dalında, Yüksek Lisans öğrenimini tamamladı. 2004-2005 Akademik yılında İstanbul Üniversitesi Fen Bilimleri Enstitüsü, Elektronik Mühendisliği Ana Bilim Dalında Doktora öğrenimine başlamıştır. 1994 yılı Lisans mezuniyetinden sonra halen bir kamu kuruluşunda çalışmaya devam etmektedir.