



**İSTANBUL
ÜNİVERSİTESİ
FEN BİLİMLERİ
ENSTİTÜSÜ**

DOKTORA TEZİ

TURBO KODLAMA TEKNİKLERİNİN BAŞARIMI

**Yük. Müh. Onur OSMAN
Elektrik-Elektronik Mühendisliği Ana Bilim Dalı**

**Danışman
Doç. Dr. Osman Nuri UÇAN**

Mart 2004

İSTANBUL



**İSTANBUL
ÜNİVERSİTESİ
FEN BİLİMLERİ
ENSTİTÜSÜ**

DOKTORA TEZİ

TURBO KODLAMA TEKNİKLERİNİN BAŞARIMI

**Yük. Müh. Onur OSMAN
Elektrik-Elektronik Mühendisliği Ana Bilim Dalı**

**Danışman
Doç. Dr. Osman Nuri UÇAN**

Mart 2004

İSTANBUL

İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bu çalışma 1/3/2004 tarihinde aşağıdaki jüri tarafından Elektrik-Elektronik Mühendisliği Anabilim Dalında Doktora Tezi olarak kabul edilmiştir.

Tez Jürisi

Doç. Dr. Osman Nuri Uçan (Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof. Dr. Mahmut Ün
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof. Dr. Mithat Uysal
Doğuş Üniversitesi
Mühendislik Fakültesi

Prof. Dr. Halit Pastacı
Yıldız Teknik Üniversitesi
Elektrik-Elektronik Fakültesi

Doç. Dr. Hakan A. Çırpan
İstanbul Üniversitesi
Mühendislik Fakültesi

ÖNSÖZ

Doktora öğrenimimin henüz başında tanışmış olduğum, bana bu tez boyunca ve tüm akademik hayatımın ilerlemesinde çok büyük katkıları olmuş, bundan sonraki tüm akademik hayatım boyunca beraber çalışmak ve yanımda olduğunu bilmek istediğim sevgili hocam Doç. Dr. Osman Nuri Uçan'a sonsuz teşekkürlerimi sunarım.

Tez çalışmam boyunca bana her zaman yardımcı olan Yrd. Doç. Dr. Selçuk Parker'e, tezimin daha başlarındayken bir çok zaman beraber çalıştığım Aykal Gümüş'e ve beraber yaptığımız bir çok çalışmada göstermiş olduğu performanstan dolayı Araş. Gör. Niyazi Odabaşıoğlu'na teşekkür ederim.

Son olarak, bana bu tez süresince her zaman destek veren ve katlanmak zorunda kalan sevgili eşim Aygen Osman'a ve beni her zaman en iyi şekilde yetiştirmek için çalışan sevgili annem ve babama şükranlarımı sunarım.

Mart, 2004

Onur OSMAN

İÇİNDEKİLER

ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİL LİSTESİ	v
TABLO LİSTESİ	x
SEMBOL LİSTESİ	xi
ÖZET	xii
SUMMARY	xiv
1. GİRİŞ	1
1.1. Tezin İçeriği	4
1.2. Tez ile Getirilen Yenilikler	6
2. MALZEME VE YÖNTEM	8
2.1. Kodlama Yöntemleri	8
2.1.1. Enformasyon Teorisi	9
2.1.2. Kanal Kapasitesi	10
2.1.3. Blok Kodlar	11
2.1.3.1. Kod Uzaklığı	12
2.1.3.2. Kod Uzunluğu	12
2.1.4. Cebrik Kodlar	13
2.1.5. Değişken Uzunluklu Kodlar	14
2.1.6. Döngüsel Kodlar	17
2.1.7. Katlamalı Kodlar	18
2.1.8. Geri Beslemeli Sistemik Katlamalı Kodlar	23
2.1.9. Kafes Kodlamalı Modülasyon	26
2.1.9.1. Kafes Yapıları ve Kafes Kodlayıcılar	29
2.1.9.2. Viterbi Algoritması	32
2.2. Turbo Kodlama	35
2.2.1. Turbo Kodlayıcı	35
2.2.2. Başarım Analizi	38
2.2.3. Turbo Kod Çözme	40
2.2.3.1. Yumuşak-Giriş Yumuşak-Çıkış Kod Çözümü	41
2.2.3.2. En Büyük İleriye Dönük Algoritma	45
2.2.3.3. Max-log-MAP ve log-MAP Algoritmaları	49

2.3. Turbo Kafes Kodlamalı Modülasyon.....	53
2.3.1. Turbo Kafes Kodlayıcı	53
2.3.2. Faz Kaydırmalı Modülasyon	53
2.3.3. Turbo Kafes Kod Çözücü.....	55
2.3.3.1. Dışsal, Ön ve Sistematik Bileşenler	57
2.3.3.2. Ön Ölçekleyici hesaplaması	58
2.3.3.3. Turbo Kafes Kodlamalı Modülasyon için MAP Algoritması	59
3. BULGULAR	65
3.1 Sönümlemeli Kanalların ve Faz Bozulmasının Turbo Kafes Kodlara Etkisi	65
3.1.1. Sönümleme Etkisi	65
3.1.2. Faz Bozulması Etkisi.....	66
3.1.3. Sönümlemeli Kanallarda Faz Bozulması Etkisi Altında Turbo Kafes Kodlamalı Modülasyon'un Başarımı	70
3.2. Gözü Kapalı Kanal Parametre Kestirimi	85
3.2.1. Baum-Welch Algoritması	88
3.2.2. Simülasyon Sonuçları.....	93
3.3. Dik Frekans Bölümlemeli Modülasyon	99
3.3.1. IFFT Kullanılarak Alt-Taşıyıcıların Oluşturulması	103
3.3.2. Koruma Süresi ve Periyodik İlaveler	107
3.3.3. Pencereleme	109
3.3.4. Simülasyon Sonuçları.....	111
3.4. Turbo Kafes Kodlamalı Sistemlerin Uzay Zaman Kodları ile Başarımı	120
3.4.1. Uzay Zaman Blok Kodları	121
3.4.2. Tekrarlamalı Uzay Zaman Kodları	124
3.4.3. Uzay Zaman Blok Kodları ve Tekrarlamalı Uzay Zaman Kodlar için TTCM Başarımı	129
3.5. Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama	134
3.5.1. Sürekli Faz Modülasyonu.....	135
3.5.2. Sürekli Faz Frekans Kaydırmalı Anahtarlama	138
3.5.2.1. 2'li CPFSK , $h=1/2$	146
3.5.2.2. 4'lü CPFSK , $h=1/2$	148
3.5.2.3. 8'li CPFSK, $h=1/2$	150
3.5.2.4. 16'lı CPFSK, $h=1/2$	153

3.5.2.5. 4'lü CPFSK, $h=1/4$	158
3.5.3. Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama Yapısı	161
3.5.3.1. Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama Kodlayıcı Yapısı	161
3.5.3.2. Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama Kodçözücü Yapısı.....	167
3.5.4. Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama Simülasyon Sonuçları.....	172
4. TARTIŞMA VE SONUÇ.....	186
KAYNAKLAR	188
ÖZGEÇMİŞ.....	195

ŞEKİL LİSTESİ

Şekil 1.1 Sayısal bir haberleşme sisteminin genel yapısı.....	2
Şekil 2.1 Huffman kodlamanın 1. adımı	15
Şekil 2.2 Huffman kodlamanın 2. adımı	15
Şekil 2.3 Huffman kodlamanın 3. adımı	16
Şekil 2.4 Huffman kodlamanın 4. adımı	16
Şekil 2.5 Huffman kodlamanın sonucu	16
Şekil 2.6 $g(D)=1 + D + D^2$ için döngüsel kodlayıcı.	18
Şekil 2.7 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcı.	20
Şekil 2.8 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcının durum diyagramı	21
Şekil 2.9 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcının kafes diyagramı	22
Şekil 2.10 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcı.	24
Şekil 2.11 Kodlama oranı $r = 1/2$ ve $K_c=3$ için RSC kodlayıcı durum diyagramı	25
Şekil 2.12 Kodlama oranı $r = 1/2$ ve $K_c=3$ için RSC kodlayıcı kafes diyagramı	25
Şekil 2.13 Kafes Kodlamalı Modülasyon blok şeması	28
Şekil 2.14 8-PSK işaret kümesinin ayrıştırılması	28
Şekil 2.15 2 durumlu kodlanmış 8-PSK modülasyon	29
Şekil 2.16 4 durumlu kodlanmış 8-PSK modülasyon	29
Şekil 2.17 8 durumlu kodlanmış 8-PSK modülasyon	30
Şekil 2.18 2 bellekli 4 durumlu 8-PSK katlamalı kodlayıcı	31
Şekil 2.19 3 bellekli 8 durumlu 8-PSK katlamalı kodlayıcı	31
Şekil 2.20 2 bellekli 4 durumlu 8-PSK geri beslemeli sistematik katlamalı kodlayıcı.....	32
Şekil 2.21 3 bellekli 8 durumlu 8-PSK geri beslemeli sistematik katlamalı kodlayıcı.....	32
Şekil 2.22 Turbo Kodlayıcı	36
Şekil 2.23 2 bellekli Turbo Kodlayıcı	37
Şekil 2.24 $1/2$ oranlı RSC kodlayıcı için yumuşak-giriş yumuşak-çıkış (SISO) kod çözücü	41
Şekil 2.25 BPSK Modülatör.....	42
Şekil 2.26 Turbo Kod Çözücü.....	44
Şekil 2.27 Turbo Kafes Kodlayıcısı	54
Şekil 2.28 Kodlayıcı girişi ve eşleyici çıktısı.....	54
Şekil 2.29 TTCM Kod Çözücü	55

Şekil 3.1 AWGN kanal için TTCM hata başarımı, $N=1024$	72
Şekil 3.2 AWGN kanal, $\alpha=20\text{dB}$ 'lik faz bozulması ve kanal bilgisinin olmadığı durumda TTCM hata başarımı, $N=1024$	72
Şekil 3.3 AWGN kanal, $\alpha=10\text{dB}$ 'lik faz bozulması ve kanal bilgisinin olmadığı durumda TTCM hata başarımı, $N=1024$	73
Şekil 3.4 AWGN kanal için TTCM hata başarımı, $N=256$	73
Şekil 3.5 AWGN kanal, $\alpha=20\text{dB}$ 'lik faz bozulması ve kanal bilgisinin olmadığı durumda TTCM hata başarımı, $N=256$	74
Şekil 3.6 AWGN kanal, $\alpha=10\text{dB}$ 'lik faz bozulması ve kanal bilgisinin olmadığı durumda TTCM hata başarımı, $N=256$	74
Şekil 3.7 $K=10\text{dB}$ 'lik Rician kanal için kanal bilgisinin tam olarak bilinmesi durumunda TTCM hata başarımı, $N=1024$	76
Şekil 3.8 $K=10\text{dB}$ 'lik Rician kanal için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$	76
Şekil 3.9 $K=10\text{dB}$ 'lik Rician kanal ve $\alpha=20\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$	77
Şekil 3.10 $K=10\text{dB}$ 'lik Rician kanal ve $\alpha=10\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$	77
Şekil 3.11 $K=10\text{dB}$ 'lik Rician kanal için kanal bilgisinin tam olarak bilinmesi durumunda TTCM hata başarımı, $N=256$	78
Şekil 3.12 $K=10\text{dB}$ 'lik Rician kanal için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=256$	78
Şekil 3.13 $K=10\text{dB}$ 'lik Rician kanal ve $\alpha=20\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=256$	79
Şekil 3.14 $K=10\text{dB}$ 'lik Rician kanal ve $\alpha=10\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=256$	79
Şekil 3.15 Rayleigh kanal için kanal bilgisinin tam olarak bilinmesi durumunda TTCM hata başarımı, $N=1024$	81
Şekil 3.16 Rayleigh kanal için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$	81
Şekil 3.17 Rayleigh kanal ve $\alpha=20\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$	82

Şekil 3.18 Rayleigh kanal ve $\alpha=10\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$	82
Şekil 3.19 Rayleigh kanal için kanal bilgisinin tam olarak bilinmesi durumunda TTCM hata başarımı, $N=256$	83
Şekil 3.20 Rayleigh kanal için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=256$	83
Şekil 3.21 Rayleigh kanal ve $\alpha=20\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=256$	84
Şekil 3.22 Rayleigh kanal ve $\alpha=10\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=256$	84
Şekil 3.23 $K=10\text{dB}$ Rician kanal için BW kanal kestirimi kullanılarak elde edilen TTCM başarımı	94
Şekil 3.24 $K=10\text{dB}$ Rician kanal için kanal bilgisi olmadan alınan TTCM başarımı	95
Şekil 3.25 $K=10\text{dB}$ Rician kanal için mükemmel kanal bilgisi olduğu, BW kestirimi yapıldığı ve kanal bilgisi olmadığı durumlarda TTCM 4. iterasyon başarımı	95
Şekil 3.26 Rayleigh kanal için BW kanal kestirimi ile alınan TTCM başarımı	96
Şekil 3.27 Rayleigh kanal için kanal bilgisi olmadan alınan TTCM başarımı	96
Şekil 3.28 Rayleigh kanal için mükemmel kanal bilgisi, BW kestirimi ve kanal bilgisi olmadan alınan TTCM 4. iterasyon başarımı	97
Şekil 3.29 (a) geleneksel çok-taşıyıcı tekniği (b) dik çok-taşıyıcılı modülasyon tekniği ..	100
Şekil 3.30 (a) bir OFDM alt-kanalının spektrumu (b) bir OFDM sinyalinin spektrumu.	102
Şekil 3.31 OFDM modülatörü	105
Şekil 3.32 Dört alt-taşıyıcılı OFDM sinyali	105
Şekil 3.33 Radix-4 kelebeği	107
Şekil 3.34 Koruma süresi boyunca hiç bir sinyal olmamasının çoklu-yol ortamlardaki etkisi	108
Şekil 3.35 Periyodik önekli OFDM sinyali	109
Şekil 3.36 OFDM periyodik eki ve pencereleme	111
Şekil 3.37 OFDM'li bir TTCM sistemini alıcı ve verici genel yapısı	114
Şekil 3.38 TTCM-OFDM kodlayıcı yapısı	114
Şekil 3.39 TTCM-OFDM kodçözücü yapısı	115
Şekil 3.40 AWGN kanalda $N=1024$ ve alt-taşıyıcı=1024 olduğu durumda TTCM-OFDM'in 4. iterasyondaki başarımı	116

Şekil 3.41 Rician Kanalda ($K=10\text{dB}$) $N=1024$ ve alt-taşıyıcı= 1024 olduğu durumda TCM-OFDM'in 4. iterasyondaki başarımı	116
Şekil 3.42 Rayleigh kanalda $N=1024$ ve alt-taşıyıcı= 1024 olduğu durumda TCM-OFDM'in 4. iterasyondaki başarımı	117
Şekil 3.43 AWGN kanalda $N=256$ ve alt-taşıyıcı= 256 olduğu durumda TCM-OFDM'in 4. iterasyondaki başarımı	117
Şekil 3.44 Rician Kanalda ($K=10\text{dB}$) $N=256$ ve alt-taşıyıcı= 256 olduğu durumda TCM-OFDM'in 4. iterasyondaki başarımı	118
Şekil 3.45 Rayleigh kanalda $N=256$ ve alt-taşıyıcı= 256 olduğu durumda TCM-OFDM'in 4. iterasyondaki başarımı	118
Şekil 3.46 Uzay Zaman Blok Kodlayıcısı.....	121
Şekil 3.47 ST-TTCM alıcı tarafı	123
Şekil 3.48 Tekrarlamalı Uzay Zaman kodlayıcısı.....	125
Şekil 3.49 STBC/RSTC ile TCM kodlayıcı yapısı.....	129
Şekil 3.50 ST-TTCM'in AWGN ortamındaki performansı.....	130
Şekil 3.51 ST-TTCM'in Rician $K=10\text{dB}$ ortamındaki performansı	130
Şekil 3.52 ST-TTCM'in Rayleigh ortamındaki performansı.....	131
Şekil 3.53 RSTC-TTCM'in AWGN ortamındaki performansı.....	131
Şekil 3.54 RSTC-TTCM'in Rician $K=10\text{dB}$ ortamındaki performansı.....	132
Şekil 3.55 ST-TTCM'in Rayleigh ortamındaki performansı.....	132
Şekil 3.56 Sürekli faz modülatörü.....	137
Şekil 3.57 (a) CPFSK'ya ait $f(t)$. (b) $f(t)$ 'nin integrali $g(t)$	139
Şekil 3.58 İkili CPFSK faz ağacı	140
Şekil 3.59 $M=2$, $h=1/2$ için faz kafes diyagramı.....	141
Şekil 3.60 M 'li CPFSK, $h=J/P$ 'nin fiziksel eğik faz kafesi (işaret kümesi).....	144
Şekil 3.61 2'li CPFSK $h=1/2$ 'nin işaret kümesi.....	146
Şekil 3.62 4'lü CPFSK, $h=1/2$ 'nin işaret kümesi.....	148
Şekil 3.63 8'li CPFSK, $h=1/2$ 'nin işaret kümesi.....	150
Şekil 3.64 16'lı CPFSK, $h=1/2$ 'nin işaret kümesi.....	153
Şekil 3.65 4'lü CPFSK, $h=1/4$ 'ün işaret kümesi.....	158
Şekil 3.66 TTC-CPFSK'nın genel yapısı.....	162
Şekil 3.67 16-CPFSK $h=1/2$ için TTC-CPFSK kodlayıcı yapısı.....	163
Şekil 3.68 16- CPFSK için faz geçişleri.....	164

Şekil 3.69 16-CPFSK $h=1/2$ için TTC-CPFSK kodlayıcı yapısı	165
Şekil 3.70 32-CPFSK $h=1/4$ faz geçişi	167
Şekil 3.71 TTC-CPFSK kod çözücü yapısı	172
Şekil 3.72 16-CPFSK, $h=1/2$, $N=256$ için AWGN ortamda TTC-CPFSK başarımı	173
Şekil 3.73 16-CPFSK, $h=1/2$, $N=256$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı	174
Şekil 3.74 16-CPFSK, $h=1/2$, $N=256$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı	174
Şekil 3.75 16-CPFSK, $h=1/2$, $N=1024$ için AWGN ortamda TTC-CPFSK başarımı	175
Şekil 3.76 16-CPFSK, $h=1/2$, $N=1024$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı	176
Şekil 3.77 16-CPFSK, $h=1/2$, $N=1024$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı	176
Şekil 3.78 16-CPFSK, $h=1/2$, $N=5000$ için AWGN ortamda TTC-CPFSK başarımı	177
Şekil 3.79 16-CPFSK, $h=1/2$, $N=5000$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı	178
Şekil 3.80 16-CPFSK, $h=1/2$, $N=5000$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı	178
Şekil 3.81 32-CPFSK, $h=1/4$, $N=256$ için AWGN ortamda TTC-CPFSK başarımı	179
Şekil 3.82 32-CPFSK, $h=1/4$, $N=256$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı	180
Şekil 3.83 32-CPFSK, $h=1/4$, $N=256$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı	180
Şekil 3.84 32-CPFSK, $h=1/4$, $N=1024$ için AWGN ortamda TTC-CPFSK başarımı	181
Şekil 3.85 32-CPFSK, $h=1/4$, $N=1024$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı	182
Şekil 3.86 32-CPFSK, $h=1/4$, $N=1024$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı	182
Şekil 3.87 32-CPFSK, $h=1/4$, $N=5000$ için AWGN ortamda TTC-CPFSK başarımı	183
Şekil 3.88 32-CPFSK, $h=1/4$, $N=5000$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı	184
Şekil 3.89 32-CPFSK, $h=1/4$, $N=5000$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı	184

TABLO LİSTESİ

Tablo 3.1 Sönümlemeli kanallar için Turbo kodlama akış şeması	68
Tablo 3.2 Kanal kestirimli Turbo kodlama akış şeması.....	91
Tablo 3.3 OFDM ile bütünleşik Turbo kodlama akış şeması	112
Tablo 3.4 OFDM parametreleri.....	115
Tablo 3.5 Uzak Zaman kodlar ile bütünleşik Turbo kodlama akış şeması	127
Tablo 3.6 2'li CPFSK, $h=1/2$ değerleri	147
Tablo 3.7 4'lü CPFSK, $h=1/2$ değerleri	149
Tablo 3.8 8'li CPFSK, $h=1/2$ değerleri	151
Tablo 3.8 8'li CPFSK, $h=1/2$ değerleri (Devam).....	152
Tablo 3.9 16'lı CPFSK, $h=1/2$ değerleri	154
Tablo 3.9 16'lı CPFSK, $h=1/2$ değerleri (Devam).....	155
Tablo 3.9 16'lı CPFSK, $h=1/2$ değerleri (Devam).....	156
Tablo 3.9 16'lı CPFSK, $h=1/2$ değerleri (Devam).....	157
Tablo 3.10 4'lü CPFSK, $h=1/4$ değerleri	159
Tablo 3.10 4'lü CPFSK, $h=1/4$ değerleri (Devam).....	160
Tablo 3.11 16-CPFSK için giriş çıkış ve sinyal dağılımı.....	164
Tablo 3.12 32-CPFSK $h=1/4$ için TTC-CPFSK giriş çıkış tablosu	166
Tablo 3.13 TTC-CPFSK akış şeması	170

SEMBOL LİSTESİ

α	İleri olasılık
β	Geri olasılık
γ	Geçiş olasılığı
ρ	Sönümleme katsayısı
θ	Faz kayması
v	Bellek sayısı
σ^2	Gürültü varyansı
Δf	Alt-taşıyıcı aralığı
B	Band genişliği
d	Giriş biti
E	Sinyal enerjisi
E_b	Bit enerjisi
E_s	Sembol enerjisi
f_c	Taşıyıcı frekans
$f_d T$	Doppler kayması
G	Jeneratör matris
m	Giriş sayısı
M	CPFSK boyutu
N	Çerçeve uzunluğu
n	Gürültü değeri
n_R	Alıcı anten sayısı
n_T	Gönderici anten sayısı
s	Sinyal
T	Sembol süresi
T_g	Gard Süresi
T_s	Etkin sembol süresi
x	Eşlenmiş çıkış sembolü
y	Gürültülü alınan sembol

ÖZET

“Turbo Kodlama Tekniklerinin Başarımı” tezinde Turbo Kafes Kodlamalı Modülasyon (Turbo Trellis Coded Modulation, TTCM) sistemleri ayrıntılı olarak açıklanarak simülasyon sonuçları ile elde edilen çeşitli ortamlardaki başarımları incelenmiştir. Tezin özgün kısmı, Tekrarlamalı Uzay Zaman Kodları (Repeating Space Time Codes, RSTC) ve Turbo Kafes Kodlamalı-Sürekli Faz Frekans Kaydırmalı Anahtarlama (Turbo Trellis Coded-Continuous Phase Frequency Shift Keying, TTC-CPFSK) isimli iki yeni kodlama tekniğinin önerilmesi ve bu yöntemlerin diğer yöntemlerle karşılaştırılmasıdır.

Bu tezin ilk bölümlerinde kodlama yöntemleri, Turbo kodlama ve Turbo kafes kodlama açıklanmıştır. Daha sonraki bölümlerinde ise öncelikle klasik Turbo kafes kodlamalı modülasyon yapısının sönmülemeli ve faz bozulmasının olduğu kanal yapılarında, kanal bilgisinin tam olarak bilindiği ve hiç bilinmediği durumlarda performans analizi yapılmıştır.

Bir sonraki bölümde, Baum-Welch (BW) algoritması Turbo kafes yapıları için düzenlenmiş, kanal parametreleri Turbo kod çözme işleminden önce BW algoritması kullanılarak kestirilmiştir. Bu algoritma ile elde edilen kanal bilgileri kullanılarak simülasyon sonuçları alınmış ve kanal bilgisinin tam olarak bilinmesi durumundaki sonuçlara ne kadar yakın olduğu gösterilmiştir.

Daha sonra, Dik Frekans Bölümlemeli Çoğullama (Orthogonal Frequency Division Multiplexing, OFDM) yöntemi açıklanmış, TTCM'in OFDM ile birlikte kullanılması durumunda performansın ne olacağı incelenmiş, bu durumda kanal parametrelerinin BW algoritması ile kestirilmesi ile hata başarımının ne kadar iyileştiği gösterilmiştir.

Sonraki bölümde, TTCM Uzay-Zaman Blok Kodları (Space Time Block Codes, STBC) ile kullanılmış ve yeni bir Uzay Zaman kodlaması yöntemi olarak Tekrarlamalı Uzay Zaman Kodları (Repeating Space Time Codes, RTSC) sunulmuştur. Her iki sistemin performans analizleri klasik yöntemlerle karşılaştırılmıştır.

Tezin son bölümünde TTCM ve Sürekli Faz Modülasyonu (Continuous Phase Modulation, CPM) birleştirilmiştir. Sürekli faz özelliğine sahip yeni bir Turbo kodlama yöntemi olarak Turbo Kafes Kodlamalı-Sürekli Faz Frekans Kaydırmalı Anahtarlama (Turbo Trellis Coded-Continuous Phase Frequency Shift Keying, TTC-CPFSK) önerilmiştir. Bu yeni sistemin başarımı bugüne kadar elde edilen sonuçların en iyisi ve klasik TTCM yapısından 7-8 dB daha kazançlıdır.

SUMMARY

In this thesis “Performance of Turbo Coding Techniques”, Turbo Trellis Coded Modulation (TTCM) systems are explained in detail and evaluated their performances in different environments. Two new coding techniques; Repeating Space Time (RST) Codes and Turbo Trellis Coded-Continuous Phase Frequency Shift Keying (TTC-CPFSK) are proposed and compared to related schemes.

In the first chapters of the thesis, coding methods, Turbo coding and Turbo trellis coding are explained. The error performance of the classical Turbo trellis structure over different fading environments with imperfect phase is given in the next chapters whether or not the channel state information (CSI) is known.

In the next chapter, Baum-Welch (BW) algorithm is modified for Turbo trellis structure. Channel parameters are estimated using modified BW algorithm before Turbo trellis decoding process. Simulation results show that the error performance of the system with BW estimated CSI is very close to the case of perfect CSI.

Then, Orthogonal Frequency Division Multiplexing (OFDM) is explained. The performance of TTCM with OFDM system and the details about the effective parameters on the performance are given.

In the next chapter, Space Time Block Codes (STBC) are explained and a new Space Time coding method, called Repeating Space Time Codes (RTSC) are introduced. These two methods are used with TTCM and results are compared with classical approaches.

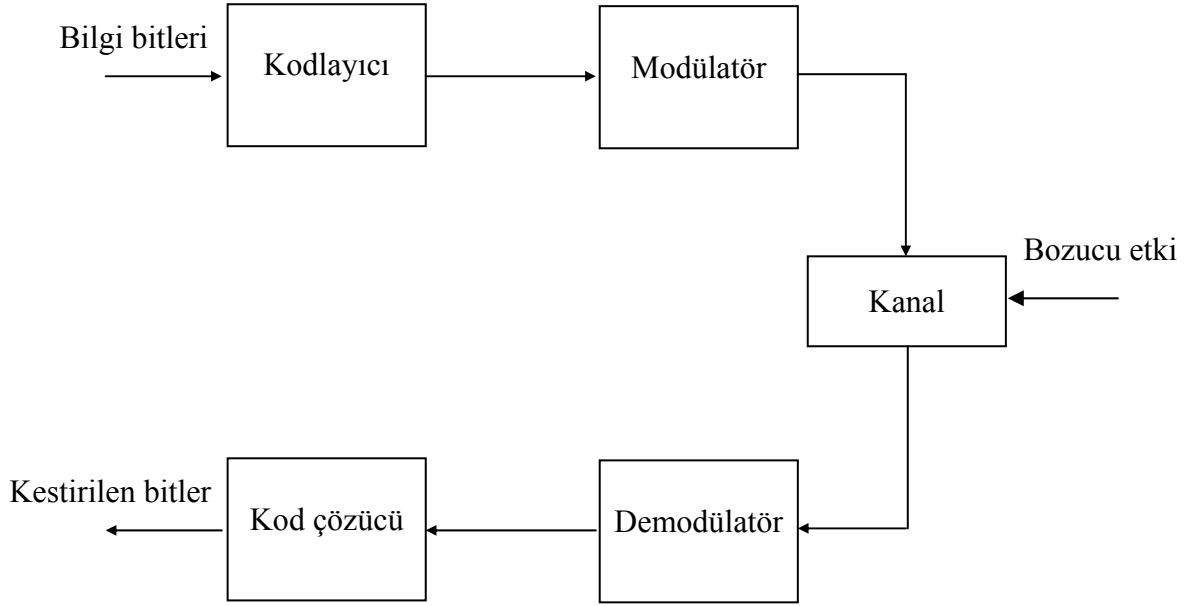
In the last chapter, TTCM and Continuous Phase Modulation (CPM) are combined. A new Turbo coding scheme with continuous phase property, Turbo Trellis Coded-Continuous Phase Frequency Shift Keying (TTC-CPFSK) is introduced. The performance results of our new scheme are extraordinary, the best till today and 7-8 dB better than classical TTCM system.

1. GİRİŞ

Kablosuz haberleşme sistemleri günümüzde çok önemli bir rol oynamaya başlamıştır. Uydu haberleşmesi dışında, özellikle cep telefonu kullanımının ve bununla birlikte iş yerlerindeki ağ sistemlerinin (Local Area Networks, LAN) kablosuz ağ sistemlere (Wireless Local Area Networks, WLAN) dönüştürülmeye başlanarak mobil haberleşme özgürlüğünün artması ve bunun getirdiği bir çok avantaj kablosuz haberleşmenin önemini daha da arttırmıştır. Şu anda sektörel olarak bakıldığında en büyük gelişme yaklaşık on yıldan beri haberleşmeye aittir.

Genel olarak sayısal habereşme süreci Şekil 1.1’de gösterildiği gibi, bilginin kodlanması, modüle edilerek analog bir sinyal haline getirilerek kanala verilmesi, kanaldan gürültülü olarak alınan sinyalin demodüle edilmesi ve son olarak da kod çözme işlemi ile gönderilmek istenen bilginin elde edilmesi işlemlerinden oluşur.

Modern sayısal haberleşme sistemlerinde hata düzeltmek için yapılan kodlama işlemi yaklaşımı, 1940’ların sonuna doğru Shannon [1] ve Hamming [2] ile başladı. Shannon [1] bu makalesinde, Enformasyon Teorisi olarak bilinen, kodlamanın teorik temellerini ortaya koymuştur. Bilgi kaynağının entropisi ile haberleşme kanalının kapasitesini matematiksel olarak tanımlamış ve gürültülü kanallarda kaynağın entropisinin kanal kapasitesinden küçük olduğu sürece güvenilir bir haberleşme sağlanabileceğini göstermiştir.



Şekil 1.1 Sayısal bir haberleşme sisteminin genel yapısı

Son yıllarda hızlı haberleşmeye gereksinim duyulması nedeniyle bant verimliliğini ve hata başarımını iyileştiren yaklaşımlar önem kazanır. Haberleşme sistemlerinin demodülasyon işlemi sırasında alınan kanal simgelerinin doğrudan ikili dizilere karşı düşürülmesi ve Viterbi Algoritması'na [3] göre çalışan kod çözücüde bu ikili dizilerin kullanılması, yani “sert kararlı” bir kod çözücü kullanılması, sonradan geri alınamayacak kayıplara neden olmaktaydı. Bant genişliğini arttırmaksızın, kodlama kazancının yüksek olmasını sağlayan kodlama ve modülasyonun birlikte düşünüldüğü kafes kodlama tekniği (TCM) Ungerboeck tarafından 1982 yılında önerilmiştir [4-6]. Bundan sonra TCM ile kodlanmış işaretlerin toplamsal beyaz Gauss gürültülü (AWGN) [7-11] ve sönmülemeli (fading) kanallar için yeni sistemler geliştirildi [12-16].

1993 yılında Berrou, Glavieux ve Thitimajshima Turbo kodları [17] buldular. Bu yöntemde ikili tek bir bit dizisi, genellikle iki kodlayıcı ve bir karıştırıcıdan oluşan bir kodlayıcıya girer, çıkış ise sistematik bilgi ve kodlayıcılar tarafından kodlanan parite bilgisidir. Kodlayıcı çıkışındaki bitler ikili faz kaydırmalı anahtarlama (Binary Phase Shift Keying, BPSK) yapılarak gönderilir. Bu sistemin hata başarımı Viterbi algoritmasından çok daha iyi ve Shannon limite de çok yakındır. Kod çözme işlemi *en büyük ileriye dönük* (Maksimum a posteriori, MAP) yöntemi ile iteratif olarak yapılmakta ve burada da genellikle Bahl Cocke Jelinek Raviv (BCJR) [18] algoritması kullanılmaktadır. BCJR algoritması çok fazla çarpma ve üstel işlemlerinden oluşmaktadır. İlk defa Erfanian ve Pasupathy [19], Jacobian logaritması ile basitleştirilmiş MAP algoritmasını geliştirdiler. Bu algorithmada çarpma işlemleri toplama işlemine, toplama işlemleri de maksimumunu seçme işlemine dönüşmüştür.

1996 yılında Blackert ve Wilson, Turbo Kafes Kodlamalı Modülasyon (TTCM) [20] yöntemini ortaya çıkarmışlardır. Bu yöntemde birinci olarak birden fazla giriş bit dizisi kodlanabilmekte ve ikinci olarak da kodlanan bilgi direkt olarak gönderilmeyip eşlenmekte, Faz Kaydırmalı Anahtarlama (Phase Shift Keying, PSK) ve Dördül Genlik Modülasyonu (Quadrature Amplitude Modulation, QAM) gibi modülasyon yöntemleri ile modüle edilerek gönderilmektedir. Bu yöntemle Turbo kodlama yöntemi, kafes kodlara uygulanmış olmaktadır. 1998 yılında Robertson ve Wörz [21] delikli kodlar için TTCM'i incelemiş ve ikili olmayan kodlar için MAP algoritmasını ayrıntılı olarak vermişlerdir.

1.1. Tezin İeriđi

Bu tezin amacı ilk olarak Turbo kafes kodlamalı modlasyon yntemini ayrıntılı olarak aıklamak ve eřitli ortamlarda bu sistemin hata bařarımlarını sunmaktır. Bu gne kadar yapılan tezler [22-24] daha ok ikili Turbo kodlama zerinedir. Bu tezde ise literatrede olmayan yeni TTCM yapıları verilerek hata bařarımları elde edilecek ve mevcut sistemlerle karřılařtırılacaktır.

Malzeme ve Yntem blmnde ilk olarak, kodlama yntemleri; Blok kodlar, cebrik kodlar, Huffman kodları, periyodik kodlar, katlamalı kodlar ve geri beslemeli katlamalı kodlar hakkında bilgi verilecek, kafes kodlama ve Viterbi algoritması kısaca aıklanacaktır. Bu blmde daha sonraki Turbo kodlama ve Turbo kafes kodlamalı modlasyon yntemlerinin aıklandığı blmlere bir temel oluřturması iin kodlama yntemleri zerinden kısaca geilmiřtir.

Daha sonra Turbo kodlama yntemi ve buna bađlı olarak analitik hata olasılıkları verilmiřtir. Kod özme yntemi iin de MAP algoritması ayrıntılı olarak aıklanmış, Jacobian logaritması kullanılarak log-MAP ve Max log-MAP algoritmaları elde edilmiş ve aralarındaki farktan bahsedilmiřtir.

Malzeme ve Yöntem Bölümünün sonunda ise Turbo kafes kodlamalı modülasyon ayrıntılı olarak açıklanmış, kodlayıcı ve kod çözücü yapıları gösterilmiş, birden çok girişe sahip kodlayıcılar için MAP algoritması verilmiştir.

Bulgular bölümünde Turbo kafes kodlamalı modülasyonunun hata başarımı çeşitli kanallar için bulunmuştur. Genel olarak AWGN ve Rician kanalları göz önünde bulundurulmuş ayrıca faz bozulmasının hata başarımına olan etkisi de incelenmiştir. Bu simülasyonlar, farklı çerçeve uzunlukları ve iterasyon sayıları için yapılmıştır.

Yine aynı bölümde kanalların oluşturduğu sönümleme ve faz bozulma etkilerinin Baum-Welch algoritması ile kestirimi açıklanmaktadır. Böylece kanal katsayıları yaklaşık olarak elde edilecek ve bu katsayılar kod çözücüde kullanılarak hata başarımının artması sağlanacaktır.

Daha sonra, dik frekans bölümlmeli çoğullama'dan (Orthogonal Frequency Shift Keying, OFDM) bahsedilecek, TTCM'e nasıl uygulanacağı ve temel özellikleri açıklanacaktır. Belirli bir taşıyıcı sayısı ve band genişliği için hata başarımı elde edilecektir.

OFDM'den sonra çok antenli sistemler için Uzay-Zaman Blok kodları (Space Time Block Codes, STBC) açıklanacak 2 vericili - 1 alıcılı sistem TTCM'e uygulanacak,

çeşitli kanallar için hata başarımı elde edilecektir. Daha sonra Tekrarlamalı Uzay-Zaman Kodları (Repeating Space Time Codes, RSTC) önerilecek, klasik yöntemle karşılaştırılacaktır.

Malzeme ve Yöntem bölümünün sonunda ise sürekli faz modülasyonunun özel bir durumu olan sürekli faz frekans kaydırmalı anahtarlama (CPFSK) yöntemi gösterilecektir. Daha sonra CPFSK ve TTCM'nin birleştirilmesiyle yeni oluşan Turbo Kafes Kodlamalı-Sürekli Faz Frekans Kaydırmalı Anahtarlama (TTC-CPFSK) sisteminin kodlayıcı ve kod çözücüsünün nasıl olması gerektiği ayrıntılı olarak açıklanacaktır. TTC-CPFSK'nın, CPFSK'nın çeşitli değerleri, kodlayıcının farklı çerçeve uzunlukları ve çeşitli kanallar için hata başarımı elde edilecek ve klasik TTCM ile karşılaştırılacaktır.

1.2. Tez ile Getirilen Yenilikler

Bu tez, Turbo kafes kodlamalı modülasyon yönteminin çeşitli ortamlardaki hata başarımlarını, çeşitli sistemlerle birlikte kullanılması durumunda elde etmek ve karşılaştırmak amacıyla yapılmıştır.

Şimdiye kadar yapılan çalışmalarla, TTCM'in çeşitli kanallardaki başarımları, OFDM ve Uzay-Zaman kodları ile birlikte kullanılması literatürde ayrıntılı olarak vardır. Tez ile getirilen yenilikler kısaca şu şekilde maddelenebilir.

- 1) TTCM kodlarında kanalın neden olduğu genlikteki ve fazdaki bozulmayı kestirebilmek için Baum-Welch algoritması ilk defa bu tezde kullanılmaktadır.
- 2) Tekrarlamalı Uzay-Zaman kodları ismi verilen yeni bir Uzay-Zaman kodlama yöntemi geliştirilmiş ve TTCM'e uygulanmıştır. Sonuçlar klasik yöntemden 0.6 dB daha iyi çıkmıştır. Fakat bu sistem, anten sayısına bağlı olarak iletişim hızında yavaşlamaya neden olmaktadır.
- 3) Tezin en önemli noktası Malzeme ve Yöntem bölümünün son üç alt bölümünde verilmektedir. Burada CPFSK, TTCM ile birleştirilmekte ve TTC-CPFSK isimli yeni bir model ortaya konulmaktadır. Kodlayıcıda bir anahtarın olması ve bu nedenle gönderilen ardışık iki sinyalin aynı kodlayıcıdan gelmemesine rağmen fazdaki sürekliliğin sağlanabileceği gösterilmiştir. TTC-CPFSK ile hem Turbo kafes kodlamanın hem de CPFSK'nın avantajlarını kullanmakta, band verimliliği arttırılmakta ve mükemmel bir hata başarımı elde edilmektedir. Bu yöntem, klasik TTCM ile karşılaştırıldığında en az 7-8 dB lik bir fark görülmektedir. TTC-CPFSK ile elde edilen sonuçlar şu ana kadar literatürdeki en iyi sonuçlardır.

2. MALZEME VE YÖNTEM

Hata olasılığını azaltmak ya da başka bir deyişle hata başarımını arttırmak için yapılan işleme genel olarak kodlama denir. Kodlama yöntemlerine geçmeden önce Enformasyon Teorisi ve Kanal Kapasitesinden bahsetmek daha doğru olacaktır. Daha sonra Blok kodlardan, kod uzaklığından, kod uzunluğundan, Cebrik kodlardan, Huffman kodlarından, Döngüsel (Cyclic) kodlardan, Katlamalı (Convolutional) kodlardan ve Kafes Kodlamadan bahsedilecektir. Bilindiği gibi bu bilgiler, burada yararlanılmış olan [25,26] gibi bir çok sayısal haberleşme kitabında bulunmaktadır. Kodlama yönteminin hata başarımında önemi oldukça fazladır. Kullanılan kodlama yöntemine bağlı olarak, kodlayıcının bellek sayısı ve kodlama işlemi sonucunda elde edilen işaretlerin uzaydaki yerleşimleri de önemlidir. Bunun dışında, Shannon'un *Gürültülü Kanal Kodlama Teoremi*, uzun rasgele kodlar kullanılması durumunda, gereken minimum sinyal gürültü oranında en güvenilir haberleşmeyi sağlayacağı şeklindedir. Fakat pratikte rasgele kodları uygulamak çok mümkün değildir. Çünkü kodçözme algoritması ile bir yol izlenebilmesi için kodların bir yapıya sahip olması gerekmektedir. Bu nedenle yukarıda bahsedilen kodlama yöntemleri gerçekleştirilebilir yapıya sahiptirler.

2.1.1. Enformasyon Teorisi

Enformasyonun miktarının tanımlaması, olasılık arttıkça monotonik olarak azalması ve olasılık 1 olduğunda 0 olmasıdır. Diğer bir özelliği de toplanabilir olmasıdır. Yani aynı anda iki tane birbirinden bağımsız mesaj varsa, toplam enformasyon her iki mesajın ayrı ayrı enformasyonlarının toplamlarına eşit olmasıdır. Bu durumda bir x mesajın Enformasyon miktarı I_x

$$I_x = \log_2\left(\frac{1}{P_x}\right) \quad (2.1)$$

olarak tanımlanır. Bu formüle göre, Enformasyon miktarı x mesajının olasılığının $\frac{1}{2}$ olması durumunda maksimum olur.

Mesaj başına ortalama enformasyon miktarına da *entropi* denir. Bunu hesaplamak için, mesajlara karşılık düşen enformasyon miktarlarını olasılıkları ile çarpıp toplamaları alınır. n adet mesaj varsa, x_1 den x_n 'e kadar, entropi aşağıdaki gibi hesaplanır.

$$H = \sum_{i=1}^n P_{x_i} I_{x_i} = \sum_{i=1}^n P_{x_i} \log_2\left(\frac{1}{P_{x_i}}\right) \quad (2.2)$$

2.1.2. Kanal Kapasitesi

Shannon Kanal Kapasitesi Teoremine göre; bir kanalın en fazla taşıyabileceği enformasyon miktarına kanal kapasitesi denir ve C ile gösterilir. Kanal kapasitesi bir kanaldan bir saniyede kaç bit geçebileceğini gösterir. Band sınırlı beyaz Gauss gürültülü kanallar için kanal kapasitesi,

$$C = B \log_2(1 + S / N) \quad (2.3)$$

burada, B band genişliğini S/N ise sinyal gürültü oranını göstermektedir.

Eğer enformasyon oranı $R=rH$, ki burada r mesaj oranını göstermektedir, kanal kapasitesi C den küçük ise bazı kodlama yöntemleri ile gönderilirken oluşan hatalar önlenabilir. Aksi durumda, yani R nin C den büyük olduğu durumda, kodlama teknikleri kullanılsa bile hatalar önlenemez veya düzeltilemez. Bundan dolayı $R \leq C$ seçilir

Gürültünün tek yanlı güç yoğunluk spektrumu $N_0/2$ olduğu düşünülürse, kanal kapasitesi

$$C = B \log_2\left(1 + \frac{S}{N_0 B}\right) \quad (2.4)$$

maksimum kanal kapasitesini bulmak için B yi sonsuza götürürsek,

$$\begin{aligned}
C &= \lim_{B \rightarrow \infty} B \log_2 \left(1 + \frac{S}{N_0 B} \right) \\
&= \lim_{B \rightarrow \infty} \frac{S}{N_0} \log_2 \left(1 + \frac{S}{N_0 B} \right)^{N_0 B / S} \\
&= \frac{S}{N_0} \log_2 e = 1.44 \frac{S}{N_0}
\end{aligned} \tag{2.5}$$

elde edilir.

2.1.3. Blok Kodlar

Blok kod, belli sayıdaki mesaj parçalarının kabul edilebilir birleşimlerine kod kelimelerini atama kuralları kümesidir. Gönderdiğimiz bitlerin alıcıda tek çözülebilir (*uniq decipherable*) olması gerekir. Örneğin 4 olası durum için 1, 01, 001 ve 0001 ikili kodlarının gönderildiğini düşünelim. Bu örnekte başlangıç zorunluluğuna (*prefix restriction*) uyulmuştur ve bu mesaj kodları anında çözülebilir (*instantaneously decipherable*) mesaj kodlarıdır. Fakat tek çözülebilirlik için bu yeterli olsada zorunlu değildir. Örneğin 1, 10, 100 ve 1000 ikili kodlarının gönderildiğini düşünelim. Burada başlangıç zorunluluğuna uyulmamıştır. Yani her mesajın başındaki bit veya bitler diğer mesajlardan oluşmaktadır. Böyle bir blok kodlama kullanıldığı durumda, alıcı tarafa 10 gelmiş olsun, sonraki iki bite bakmadan bunun hangi mesaj bilgisi olduğuna karar veremeyiz yani bu mesaj kodları anında çözülebilir değildir, fakat tek çözülebilirdir.

2.1.3.1. Kod Uzaklığı

İki eşit boyuttaki kodlar arasında farklı bit sayısına Hamming uzaklığı (*Hamming distance*) ya da kod uzaklığı denir. Kod uzaklığı gürültünün olduğu durumlarda ve hatanın düzeltilmesi için önemlidir. İki kod arasında kod uzaklığı ne kadar çok olursa, alıcıda doğru alınma olasılığı o kadar artacaktır. Bu durumda kod uzaklığının çok olabilmesi için kod uzunluğunun da çok olması gerekir. Bu yüzden mesaj kodunun arkasına tekrar bitleri koyulabilir. Kodlar arasındaki en düşük uzak $D_{\min}$ ile gösterilir. Alıcıda $D_{\min}-1$ bit'e kadar hatanın olduğu anlaşılabilir ve $D_{\min}$ çift sayı ise $(D_{\min}/2-1)$ bit hata tek sayı ise $[(D_{\min}/2)-(1/2)]$ bit hata düzeltilebilir.

2.1.3.2. Kod Uzunluğu

E tane hatayı düzeltebilecek n uzunluğundaki M tane mesaj kelimesi arasında aşağıdaki gibi bir eşitsizlik sözkonusudur.

$$M \leq \frac{2^n}{\sum_{i=0}^E \binom{n}{i}} \quad (2.6)$$

Burada $\binom{n}{i}$ n 'in i 'li kombinasyonudur. Pratik açıdan bu formülle n uzunluklu M tane mesaj kelimesi için kaç hatanın düzeltilebileceği veya M tane mesaj kelimesinde E tane hatanın düzeltilebilmesi için kod uzunluğu n 'in minimum değeri bulunabileceği görülür.

2.1.4. Cebrik Kodlar

Cebrik kodlar, mesaj bitlerinin arkasına parite bitleri eklenmesi ile oluşturulur. Parite bitleri mesaj bitleri arasında çeşitli işlemler yapılarak oluşturulur. Eğer mesaj kelimemizde m tane bit varsa birbirinden farklı 2^m adet mesaj kelimesi oluşturulabilir. Buna n adet parite bit eklenmesi halinde yeni oluşacak mesaj kelimesi $m+n$ adet bitten meydana gelecektir. Eğer asıl mesaj bitlerini a_i ile gösterirsek ve parite bitleri de c_i ile gösterirsek yeni oluşan mesaj kelimesi, $a_1a_2...a_m c_1c_2...c_n$ olur. Parite bitleri

$$[H]\bar{T} = 0 \quad (2.7)$$

denklemini sağlarlar. Burada $\bar{T}$, iletilen bileri gösteren $m+n$ lik bir sütun matristir.

$$\bar{T} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \\ c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (2.8)$$

$[H]$ ise, göndericide parite bitleri yaratmayı, alıcıda ise kontrol etmeyi sağlayan $n \times (n+m)$ lik bir matristir.

$$[H] = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1m} & 1 & 0 & 0 & 0 & \dots & 0 \\ h_{21} & h_{22} & \dots & h_{2m} & 0 & 1 & 0 & 0 & \dots & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \cdot & \cdot & \dots & \cdot \\ h_{n1} & h_{n2} & \dots & h_{nm} & 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (2.9)$$

Önceden tanımlanmış bir $[H]$ matrisi, denklem (2.7) da kullanılarak parite bitlerinin hangi mesaj bitlerinin toplamından oluşacağı bulunur. Gönderilmek istenen mesaj kelimesinin arkasına bu kuralla parite bitleri eklenir.

Alıcı tarafında ise, $[H]$ gürültülü olarak alınan bitlerle çarpılır, sonuç sıfır sütun matrisi çıkarsa alınan bitlerde bir hata yok demektir. Eğer başka bir sütun matrisi çıkarsa, çıkan matrisin, $[H]$ matrisinin ilk m sütunundan kaçınıcı sütununa eşit olduğu bulunur. Bu sütun sayısı, alınan mesaj bitlerinden ilk m tanesinden hangisinde hata olduğunu gösterir.

2.1.5. Değişken Uzunluklu Kodlar

Blok kodlar hep eşit uzunluklu kod kelimelerinden oluşurlar. Fakat birçok mesajın eşit olasılıklı olmadığı durumda verimlilik farklı uzunluklu kod kelimeleri kullanılarak sağlanır. Huffman kodları, tek ve anında çözülebilir kodlar oluşturmayı sağlar. Örneğin sırasıyla $1/16$, $1/8$, $1/4$, $1/16$ ve $1/2$ olasılıklı s_1 , s_2 , s_3 , s_4 ve s_5 mesajlarını kodlamak isteyelim. Huffman kodlama bunu dört adımda yapabilir.

İlk olarak mesajlar olasılıklarına göre azalan bir sırada dizilir. Eğer eşit olasılıklı olanlar varsa önce hangisinin yazılacağını bir önemi yoktur.

Mesaj	Olasılık
s_5	$1/2$
s_3	$1/4$
s_2	$1/8$
s_1	$1/16$
s_4	$1/16$

Şekil 2.1 Huffman kodlamanın 1. adımı

En alttaki iki olasılık toplanarak yeniden azalan düzende sıralanır.

Mesaj	Olasılık	Olasılık
s_5		$1/2$
s_3		$1/4$
s_2		$1/8$
s_1	} $\rightarrow$	$1/8$
s_4		

Şekil 2.2 Huffman kodlamanın 2. adımı

En düşük iki olasılığı toplayarak yeniden düzenleme işlemi iki olasılık kalıncaya kadar devam eder.

Mesaj	Olasılık	Olasılık	Olasılık	Olasılık
s_5		$1/2$	$1/2$	$1/2$
s_3		$1/4$	$1/4$	$\left. \begin{array}{c} 1/4 \\ 1/4 \\ 1/8 \\ 1/8 \end{array} \right\} \xrightarrow{s_1 \ s_4 \ s_2 \ s_3} 1/2$
s_2		$1/8$	$\left. \begin{array}{c} 1/8 \\ 1/8 \end{array} \right\} \xrightarrow{s_1 \ s_4 \ s_2} 1/4$	
s_1	$\left. \begin{array}{c} \\ \\ \end{array} \right\} \xrightarrow{s_1 \ s_4} 1/8$	$1/8$		
s_4				

Şekil 2.3 Huffman kodlamanın 3. adımı

Sağ taraftan başlanarak kod kelimeleri atanır. Sola doğru gidildiğinde bir ayrılma varsa atanan bit sola kaydırılır. Belirlenen bitlerin altları çizili olarak gösterilmektedir.

Mesaj	Olasılık	Olasılık	Olasılık	Olasılık
s_5	1/2	1/2	1/2	1/2 <u>0</u>
s_3	1/4	1/4	1/4 <u>10</u>	1/2 <u>1</u>
s_2	1/8	1/8 <u>110</u>	1/4 <u>11</u>	
s_1	1/16 <u>1110</u>	1/8 <u>111</u>		
s_4	1/16 <u>1111</u>			

Şekil 2.4 Huffman kodlamanın 4. adımı

Böylece kod kelimeleri aşağıdaki gibi bulunmuş olur.

s_1	→	1110
s_2	→	110
s_3	→	10
s_4	→	1111
s_5	→	0

Şekil 2.5 Huffman kodlamanın sonucu

Ortalama kod uzunluğu,

$$L = 4 \times 1/16 + 3 \times 1/8 + 2 \times 1/4 + 4 \times 1/16 + 1 \times 1/2 = 15/8$$

Eğer Huffman kodlama yerine blok kodlar kullanılmış olsaydı 5 mesaj için 3 bit kullanılması gerekirdi. Aynı zamanda bu olasılıklar için (2.2) denklaminden entropi bulunacak olursa,

$$H = 1/16 \log_2(16) + 1/16 \log_2(16) + 1/8 \log_2(8) + 1/4 \log_2(4) + 1/2 \log_2(2) = 15/8 \text{ bit}$$

elde edilir. Bu değer ortalama kod uzunluğu ile aynıdır. Tüm olasılıklar 2 nin bir gücü olduğu sürece Huffman kodları en verimli kodlardır.

2.1.6. Döngüsel Kodlar

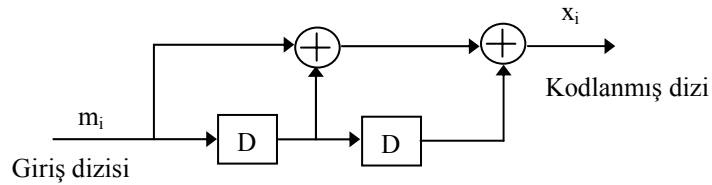
Döngüsel kodlar bir kod kelimesinin kaydırılması ile oluşan kodlardır. C blok kod kümesinde her kod kelimesi için, $\mathbf{x}=(x_0, x_1, \dots, x_{n-2}, x_{n-1})$ bu C blok kod kümesinin bir elemanı ise ve $\mathbf{x}=(x_{n-1}, x_0, x_1, \dots, x_{n-2})$ de bu kümenin bir elemanı ise C döngüseldir. Döngüsel kodları oluşturma matrisi (*generator matrix*) aşağıdaki şekildedir.

$$G = \begin{bmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & \cdots & 0 \\ \vdots & & \ddots & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_{n-k} \end{bmatrix} \quad (2.10)$$

Döngüsel kodlar oluşturma polinomu ile $g(D) = g_0 + g_1D + g_2D^2 + \dots + g_{n-k}D^{n-k}$ biçiminde oluşturulurlar. Benzer şekilde mesaj, mesaj polinomu $m(D)$ ile, kod kelimesi, kod polinomu $x(D)$ ile gösterilir. Böylece kodlama işlemi polinom çarpımı işlemine dönüşür $x(D)=m(D)g(D)$. Polinom çarpımı konvolüsyona eşdeğer olduğu için kodlama işlemi şöyledir,

$$x_i = \sum_{j=0}^{n-k} m_{i-j} g_j . \quad (2.11)$$

Oluşturma polinomu $g(D)=1 + D + D^2$ için döngüsel kodlayıcı aşağıdaki şekilde verilmiştir.



Şekil 2.6 $g(D)=1 + D + D^2$ için döngüsel kodlayıcı.

2.1.7. Katlamalı Kodlar

Katlamalı kodlar bellek elemanı kullanımı bakımından döngüsel kodlara benzerler. Fakat döngüsel kodlar blok kodlardır ve kod uzunlukları belirlidir. Katlamalı kodlar ise giriş bit dizisinin sürekli olarak kodlandığı için sabit kod uzunluğuna sahip değildirler. Diğer bir farklılık ise döngüsel kodlar lineer bir sistem olarak düşünülebilir, tek girişli

ve tek çıkışlı bir sistem olarak karakterize edilebilirler. Katlamalı kodlayıcılar ise çok girişli ve çok çıkışlı sistemlerdir.

Katlamalı kodların blok kodlar gibi kod oranı $r = k/n$ olarak gösterilir. Fakat bunların anlamları farklıdır. k katlamalı kodlayıcıya giren giriş dizisi sayısı, n ise kodlayıcının çıkışındaki çıkış dizisi sayısıdır. Giriş mesajı $\mathbf{m}$, k diziden oluşmaktadır $\mathbf{m}^{(p)} = (m_p, m_{p+k}, m_{p+2k}, \dots)$ $p \in (0, \dots, k-1)$. k adet giriş dizisi katlamalı kodlayıcıya girer ve n adet çıkış dizisi oluşur $\mathbf{x}^{(q)}$, $q \in (0, \dots, n-1)$.

Katlamalı kodlar $n \times k$ adet oluşturma polinomu $g^{(p,q)}(D)$ tarafından meydana gelir. Her biri bir giriş dizisi için, k tane kaymalı kaydedici (shift register) vardır. Kaymalı kaydedici'deki bellek sayısı p girişi için M_p ile gösterilir ve toplam bellek sayısı M_c dir.

$$M_c = \sum_{p=0}^{k-1} M_p \quad (2.12)$$

Kısıtlama uzunluğu (*constraint length*) genellikle her hangi bir giriş bitinden etkilenen, tek bir çıkış dizisindeki maksimum bit sayısına denir ve

$$K_c = 1 + \max_p M_p \quad (2.13)$$

olarak tanımlanır.

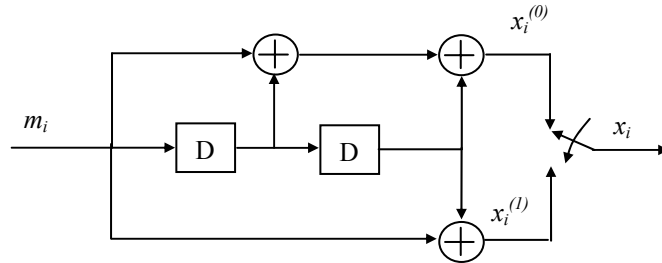
Kodlama işlemi, her oluşturma polinomunun katsayıları ile ilgili giriş arasında konvolüsyon yapılarak devam eder.

$$v_i^{(p,q)} = \sum_{j=0}^{M_p} m_{i-j}^{(p)} g_j^{(p,q)} \quad (2.14)$$

Katlamalı kodlayıcının çıkış dizisi $x^{(q)}$ aşağıdaki gibidir

$$\mathbf{x}^{(q)} = \sum_{p=0}^{k-1} \mathbf{v}^{(p,q)} \quad (2.15)$$

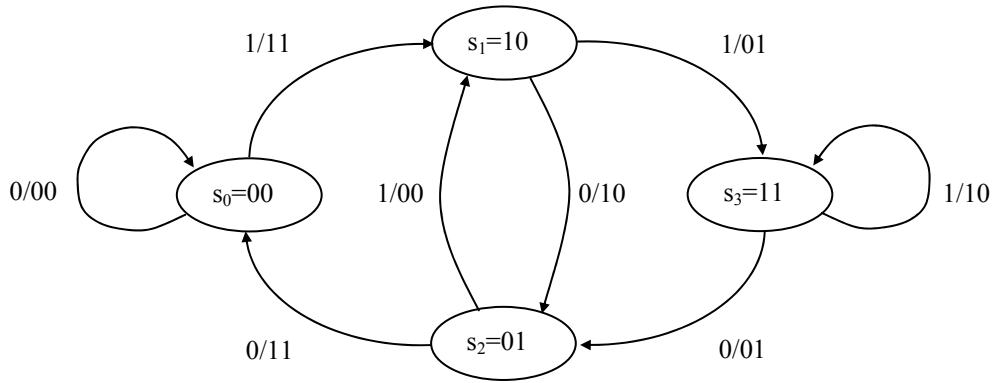
Girişte tek bir dizi olduğu düşünüldüğünde yani $k = 1$ durumu için kodlayıcı oranı $r = 1/n$ olur. Oran $1/n$ olunca $p = 0$ olur ve ihmal edilir. Bu durumda $\frac{1}{2}$ oranlı oluşturma polinomu $g^{(0)} = 1 + D + D^2$ ve $g^{(1)} = 1 + D^2$ olan bir katlamalı kodlayıcı Şekil 2.7 de verilmiştir. Her oluşturma polinomunun katsayıları genellikle sekiz tabanında (*octal*) verilir. Şekil 2.7 deki kodlayıcı için bu değer $(7,5)_{\text{octal}}$ dir.



Şekil 2.7 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcı.

Katlamalı kodlayıcılar durum diyagramı ve kafes yapı ile gösterilebilen, sonlu durumlu bir Markov sürecidir. Katlamalı kodlayıcının durumu belleklerin içerikleri ile belirlenir. M_c adet bellek için toplam 2^{M_c} adet durum sözkonusudur. Belleklerin durumları, giriş bitinin 1 veya 0 olmasına bağlı olarak başka bir duruma geçerler. Kodlayıcı bir durumdan başka bir duruma geçtiğinde buna bağlı olarak n bitlik bir çıkış gerçekleşir.

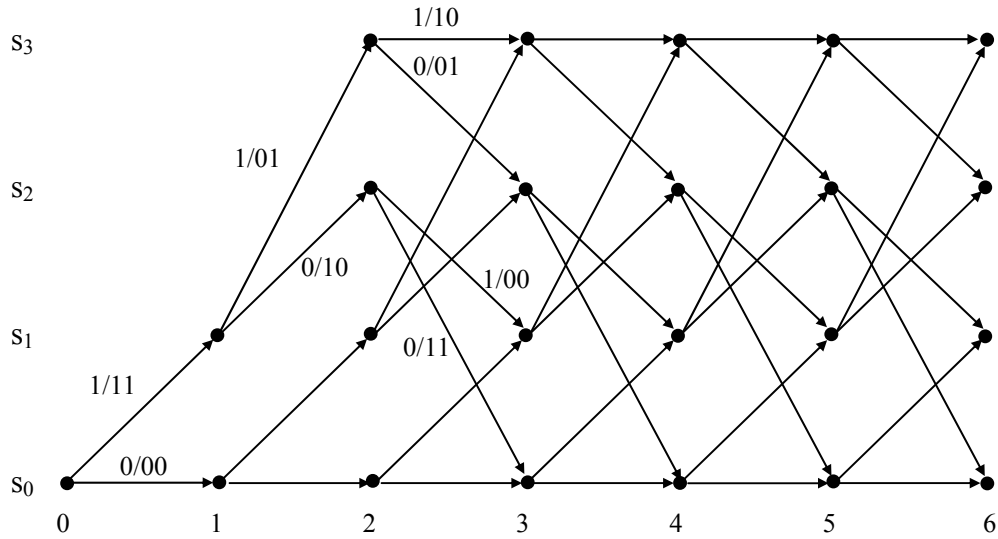
Katlamalı kodlayıcının işlevi durum diyagramı ile daha kolay anlaşılır. Durum diyagramında 2^{M_c} adet durum sözkonusudur ve bunlar s ile gösterilir. Tüm durumlar dallarla birbirlerine bağlıdır ve bu dalların üzerinde herhangi bir durumdayken m giriş yapıldığında x çıkış olacağını gösteren m/x ifadeleri vardır. Şekil 2.7 için durum diyagramı Şekil 2.8 de verilmiştir.



Şekil 2.8 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcının durum diyagramı

Durum diyagramı, her kodlama adımı için kafes yapı ile çok açık bir şekilde gösterilebilir. Şekil 2.9 da 6 kodlama adımı için kafes yapısı çizilmiştir. Katlamalı kodlarda her bir kod kelimesi kafesteki tek bir yola karşılık gelir. Bu yola durum zinciri

(*state sequence*) denir ve $\mathbf{s} = (s_0, \dots, s_L)$ ile gösterilir. Kafes yapı belleklerin sıfır durumu ile başlar. Her yolun durum zinciri $\mathbf{s}$, o yol ile ilgili kod kelimesi $\mathbf{x}$ ve o kod kelimesini oluşturan mesaj $\mathbf{m}$, birbirleriyle bire bir ilişkilidirler.



Şekil 2.9 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcının kafes diyagram

Bir katlamalı kodun minimum uzaklığına minimum serbest uzaklık (*minimum free distance*) denir ve d_{free} ile gösterilir. Bu sıfırıncı kodlama adımında sıfır durumlu yoldan uzaklaşan $\mathbf{s}$ yolları ile ilgili tüm kod kelimelerinin en küçük Hamming ağırlığıdır.

$$d_{free} = \min\{w(\mathbf{x}) : x_0 = 1\} \quad (2.17)$$

Bu bilgi en önemli başarımlı ölçütüdür.

2.1.8. Geri Beslemeli SistematiK Katlamalı Kodlar

Döngüsel kodların Hamming uzaklığını azaltmadan sistematiK kod yapılabildiği gibi katlamalı kodlar da Hamming uzaklığını azaltmadan sistematiK yapılabilir. 1/2 oranlı bir katlamalı kod $m(D)/g^{(0)}(D)$ polinom bölümünün kalanı $r(D)$ yi hesaplayarak yapılır. Parite çıkış polinomu $x^{(1)}(D)=r(D)g^{(1)}(D)$ polinom çarpımı ile bulunur. SistematiK çıktı ise $x^{(0)}=m(D)$ dir. $r(D)$ kalanı geribeslemeli bir shift register ile bulunabilir. Burada $g^{(0)}(D)$ 'ye geri besleme polinomu, $g^{(1)}(D)$ 'ye de ileri besleme polinomu denir. Bu şekilde oluşturulan kodlara geri beslemeli sistematiK katlamalı (Recursive Systematic Convolutional, RSC) kodlar denir. Kodlama işlemi geri besleme değişkenini hesaplamakla başlar,

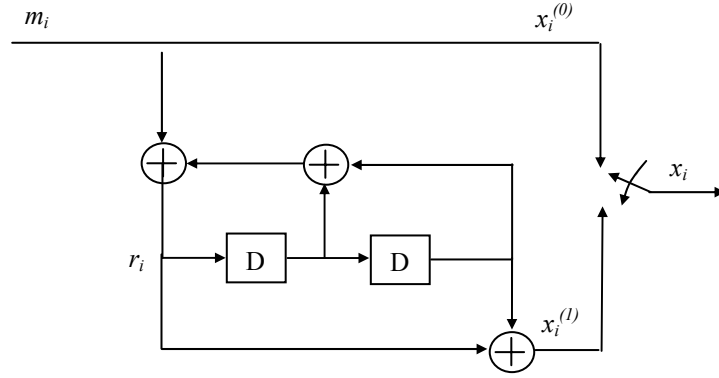
$$r_i = m_i + \sum_{j=1}^{M_c} r_{i-j} g_j^{(0)} \quad (2.18)$$

ve daha sonra parite çıktısı hesaplanır,

$$x_i^{(1)} = \sum_{j=1}^{M_c} r_{i-j} g_j^{(1)} \quad (2.19)$$

Şekil 2.10'da geri beslemeli sistematiK katlamalı kodlayıcı verilmiştir. RSC kodlayıcıları sonlu durumludur. Durum ve kafes diyagramları ile gösterilirler. Şekil 2.11 ve Şekil 2.12'de durum ve kafes diyagramları verilmiştir. Dikkat edilirse geri beslemeli sistematiK katlamalı kodlayıcıların durum ve kafes diyagramları katlamalı

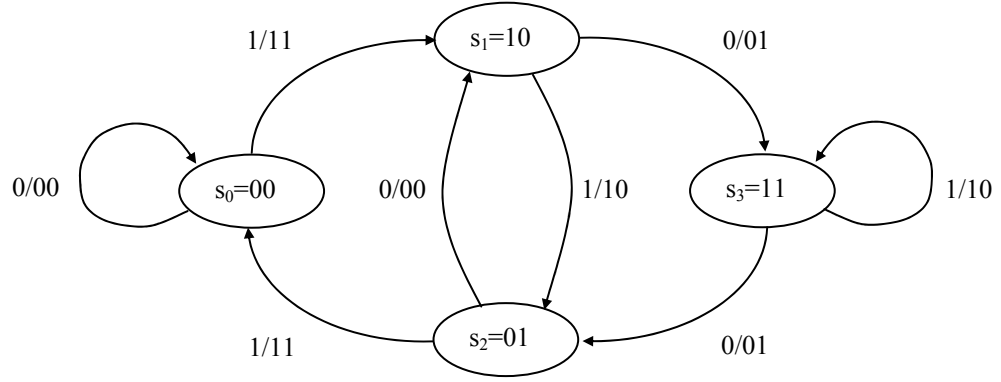
kodlayıcılarının aynısıdır. Tek fark Şekil 2.8'deki s_1 ve s_2 'den çıkan kollar için sadece giriş değerinin farklı olmasıdır. Bu durumda serbest uzaklık da değişmez.



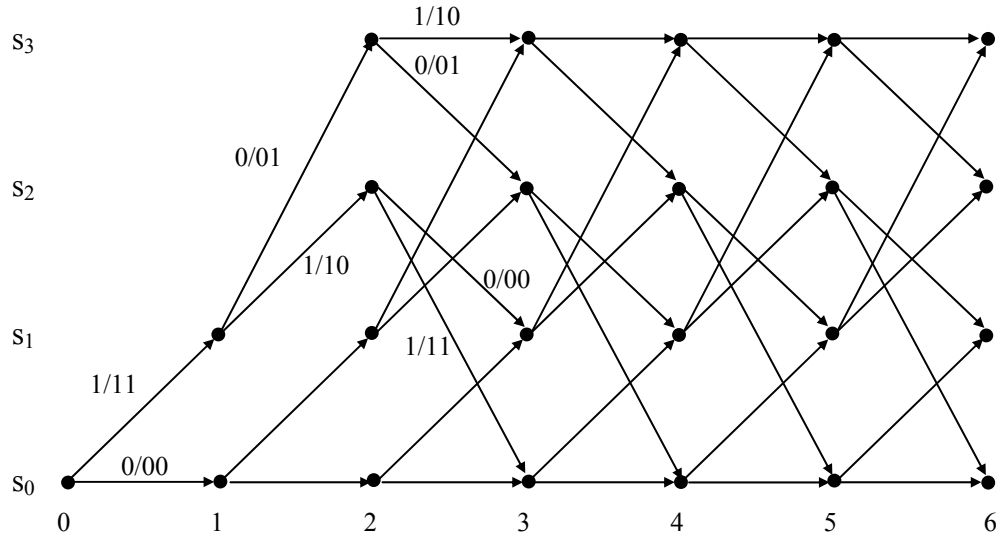
Şekil 2.10 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcı.

Genellikle sayısal haberleşmede her çerçeve gönderildikten sonra, durum s_0 'a geri getirilir, böylece bir sonraki çerçeve gönderilmeye başlanıldığında kafes yine s_0 'dan başlamış olur. Klasik katlamalı kodlayıcılarda M_c tane sıfır biti mesajın sonuna eklenerek kafes yapı tekrar s_0 a getirilir. Fakat RSC kodlayıcılar için aynı şey geçerli değildir. RSC kodlayıcılarda kafes sonlandırılmak istendiğinde, $L-M_c \leq i \leq L-1$ için r_i 'yi sıfır yapacak m_i 'ler seçilir. Bu durumda kafesi s_0 'a geri getirecek son M_c tane bit şu şekilde bulunur.

$$m_i = \sum_{j=1}^{M_c} r_{i-j} g_j^{(0)} \quad (2.20)$$



Şekil 2.11 Kodlama oranı $r = 1/2$ ve $K_c=3$ için RSC kodlayıcı durum diyagramı



Şekil 2.12 Kodlama oranı $r = 1/2$ ve $K_c=3$ için RSC kodlayıcı kafes diyagramı

2.1.9. Kafes Kodlamalı Modülasyon

Sayısal haberleşme sistemlerinin tasarımında, temel amaçlardan biri sistem hata başarımının yüksek tutulmasıdır. Hata sezme ve düzeltme özelliği, iletilecek işarete kanal kodlaması uygulanarak kazandırılır. Klasik kanal kodlayıcıları, n -bitlik ikili veri dizilerini m - n tane kontrol biti ekleyerek m ($m > n$) uzunluklu ikili kod dizilerine dönüştürürler. Kodlanmış ikili diziler arasındaki Hamming uzaklıkları kod tasarımının temel ölçütü olmaktadır. Daha sonra kodlanmış ikili bilgi, sayısal modülasyon yöntemlerinden biriyle modüle edilerek kanala verilir. Dizi uzunluklarının artması sonucu oluşan veri iletişim hızındaki azalma, modülasyon hızı ve dolayısıyla band genişliği artırılarak engellenebilir. Band sınırlaması altında ise kanal işaretleri kümesi genişletilerek aynı amaca ulaşılabilir. Ancak genişletilmiş işaret kümesi yardımıyla tasarım, kodlama ve modülasyon işlemlerinin ayrı ayrı düşünülmesi sonucu başarıdan uzaklaştırmaktadır. Bunun temel sebebi, kodlama sırasında Hamming uzaklığına dayanarak yapılan bir tasarımın, modülasyonlu işaret dizilerinin işaret uzayında optimum olarak birbirlerinden uzaklaşmasını sağlamamasıdır.

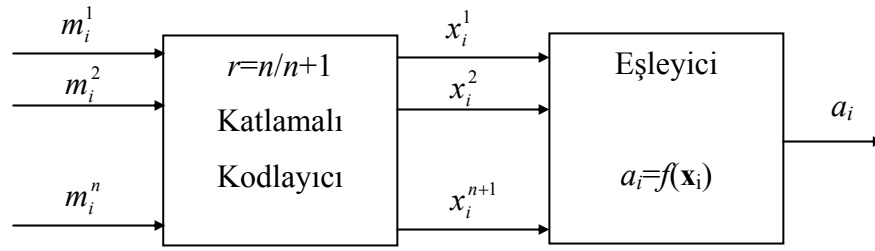
Kodlama ve modülasyonun birlikte düşünüldüğü kafes kodlamalı modülasyon (Trellis Coded Modulation, TCM) tekniği ilk defa Ungerboeck tarafından 1982 yılında önerilmiştir. TCM tekniği ile hata başarımını iyileştirmek için kanal işaretlerinin gereğinden fazla seçilmesi yeterli olmaktadır. Kafes kodlamalı modülasyon sistemi tasarımında, kafes yapısı üzerindeki ikili kod dizileri yerine bu dizilere atanan modülasyonlu kanal işaretleri yerleştirilmiştir. Kanal işaretleri a_n ve a_n' arasındaki Öklid uzaklığı $d^2(a_n, a_n')$ ile gösterilir. “Minimum karesel uzaklık”, d_{free}^2 , kodlayıcının

üretebildiği aynı durumdan ayrılan ve bir veya daha fazla geçişten sonra tekrar birleşen tüm kanal işaret dizileri arasındaki karesel uzaklıkların en küçüğüdür ve

$$d_{serbest}^2 = \min \sum_{a_n \neq a'_n} d^2(a_n, a'_n) \quad (2.21)$$

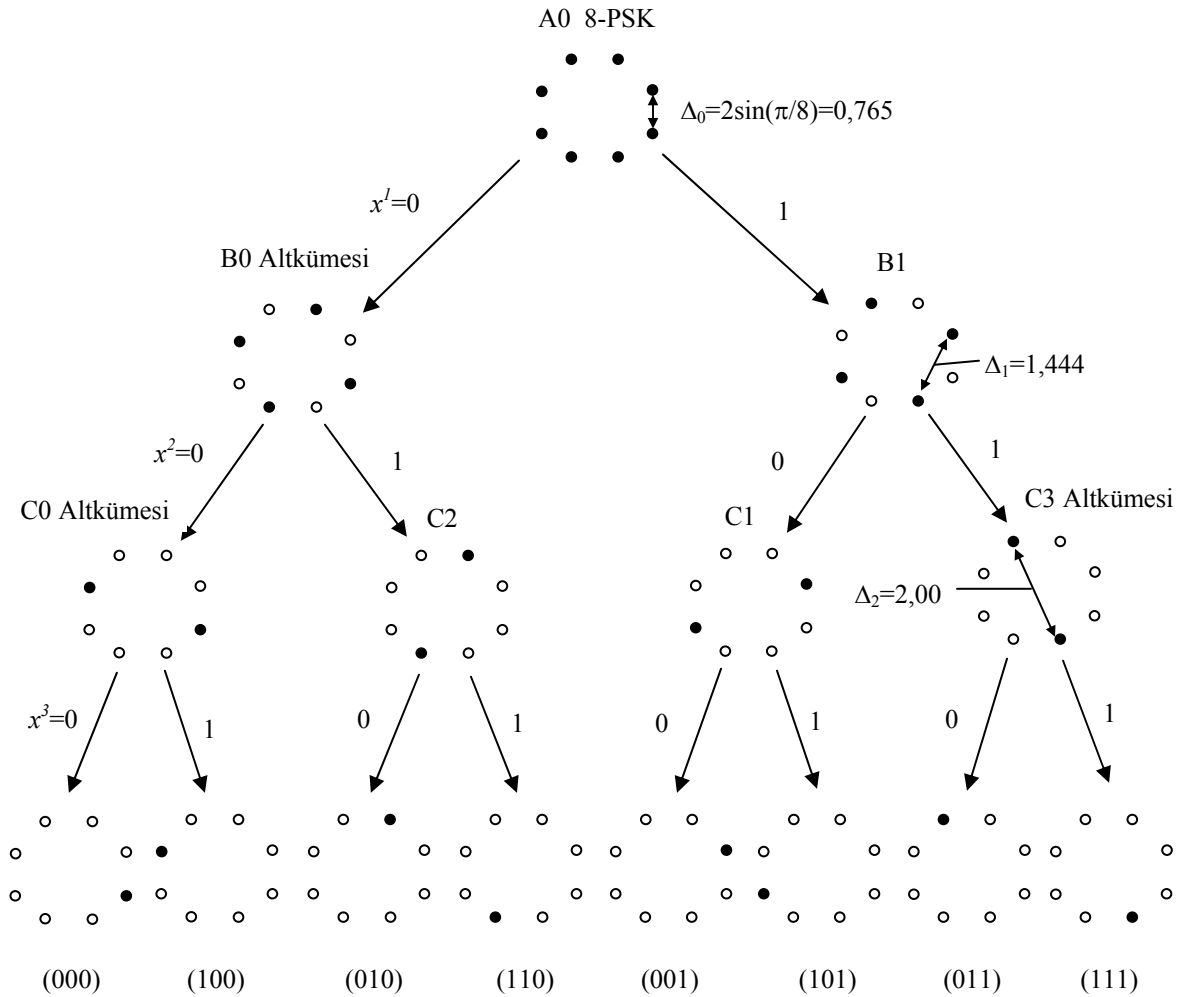
Şekil 2.13 de gösterilen birleşik yapıda, her iletim aralığında n uzunluklu giriş dizilerini $n+1$ uzunluklu dizilere dönüştüren $r=n/n+1$ oranlı katlamalı kodlayıcı kullanılmakta ve ve daha sonra $n+1$ uzunluklu diziler 2^{n+1} kanal işareti ile “küme ayırıştırarak eşleme” (*mapping by partitioning*) olarak adlandırılan kurala göre eşlenmektedir. Bu kurala göre, gereğinden iki kat fazla elemandan oluşan kanal işaretleri kümesi belli bir simetri altında aralarındaki serbest Öklid uzaklığı giderek artacak biçimde, daha az elemanlı alt kümelere bölünmektedir. Daha sonra oluşturulan bu alt kümeler, seçilen kafes yapının durum geçişlerine, kodlayıcının üreteceği kanal işareti dizileri arasındaki serbest Öklid uzaklığı maksimum olacak biçimde atanmaktadır. Bu işlem sırasında, bütün kanal işaretlerinin eşit sayıda kullanılmasına, kalkış ve varış durumlarına atanan alt kümelerin maksimum olmasına dikkat edilmelidir.

Alıcı tarafında genellikle Viterbi algoritmasına göre kod çözme işlemi yapılır. Alıcıda yumuşak kararlı kod çözücü (*soft decision decoder*) kullanıldığında daha iyi bir hata başarımına ulaşılmaktadır.



Şekil 2.13 Kafes Kodlamalı Modülasyon blok şeması

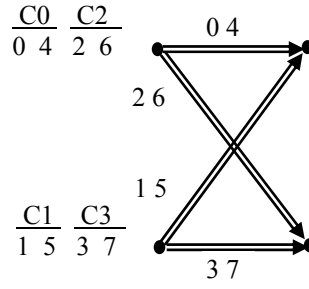
8-PSK modülasyonu için kanal işaretlerinin ayrıştırılması Şekil 2.14’de verilmiştir. Burada Δ_i , i . Alt kümeye ilişkin en yakın iki işaret arasındaki karesel uzaklığı vermektedir. Küme ayrıştırarak eşleme mantığına uygun olarak $\Delta_0 < \Delta_1 < \Delta_2$ olmaktadır.



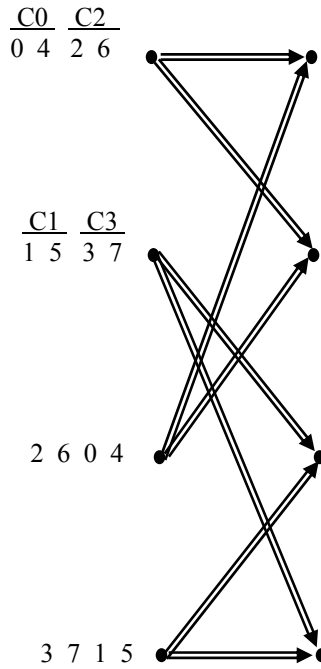
Şekil 2.14 8-PSK işaret kümesinin ayrıştırılması

2.1.9.1. Kafes Yapıları ve Kafes Kodlayıcılar

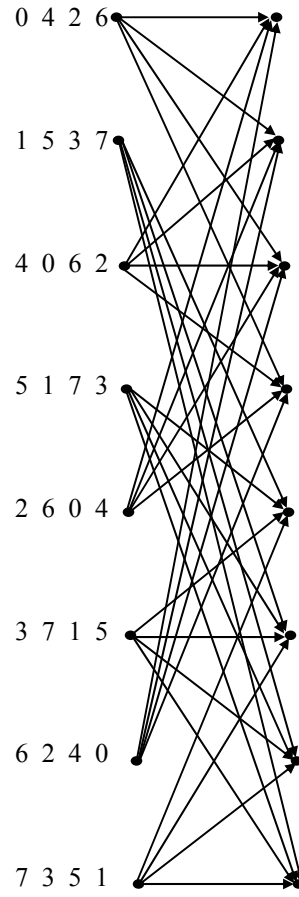
2, 4 ve 8 durumlu 8-PSK işaretleri Şekil 2.15, Şekil 2.16 ve Şekil 2.17 de gösterilmektedir.



Şekil 2.15 2 durumlu kodlanmış 8-PSK modülasyon

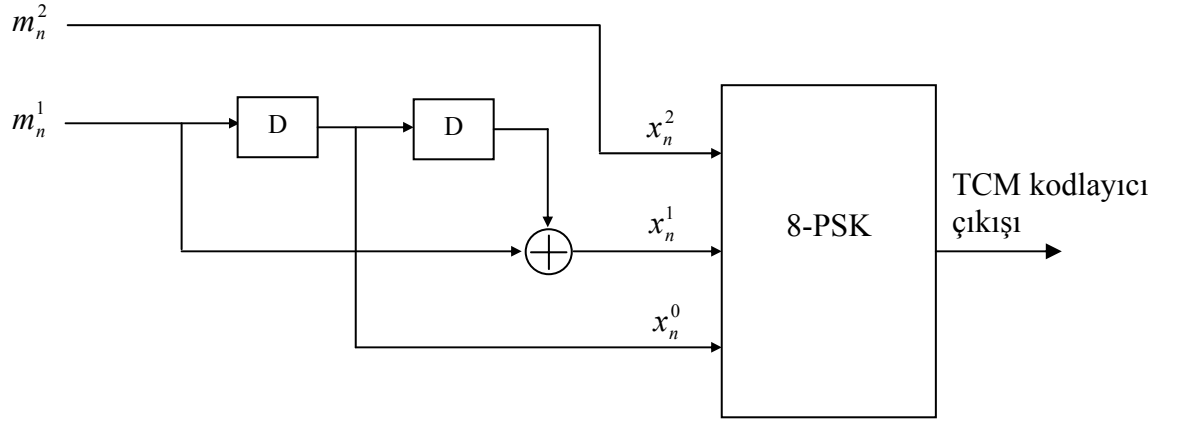


Şekil 2.16 4 durumlu kodlanmış 8-PSK modülasyon

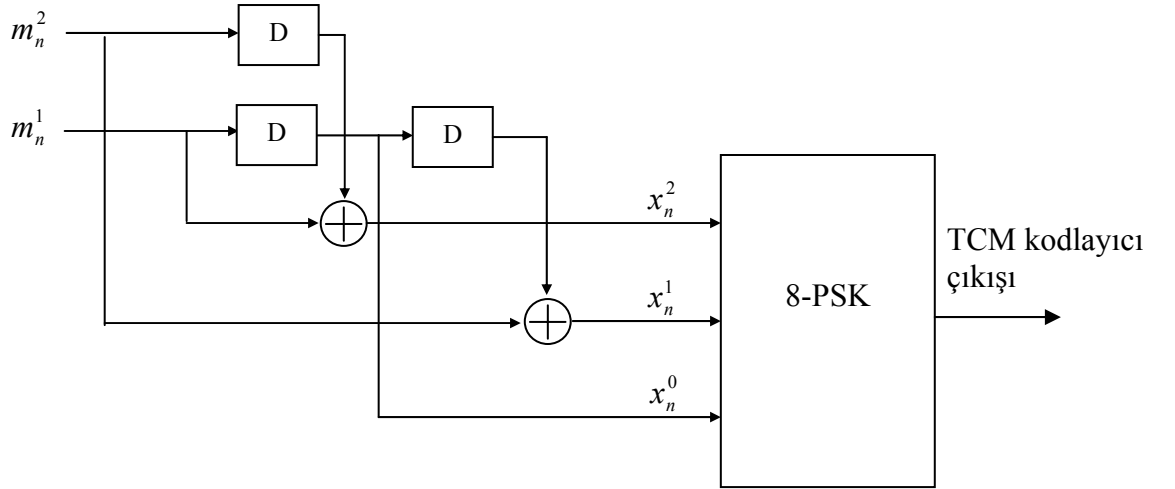


Şekil 2.17 8 durumlu kodlanmış 8-PSK modülasyon

Şekil 2.15, 2.16 ve 2.17, Şekil 2.14'deki ayrıştırma işlemine uygun ve Şekil 2.18 ve 2.19'da verilen kodlayıcılar için verilmiştir. Şekil 2.18 ve 2.19, 2 ve 3 bellekli 4 ve 8 durumlu kafes yapıları göstermektedir.

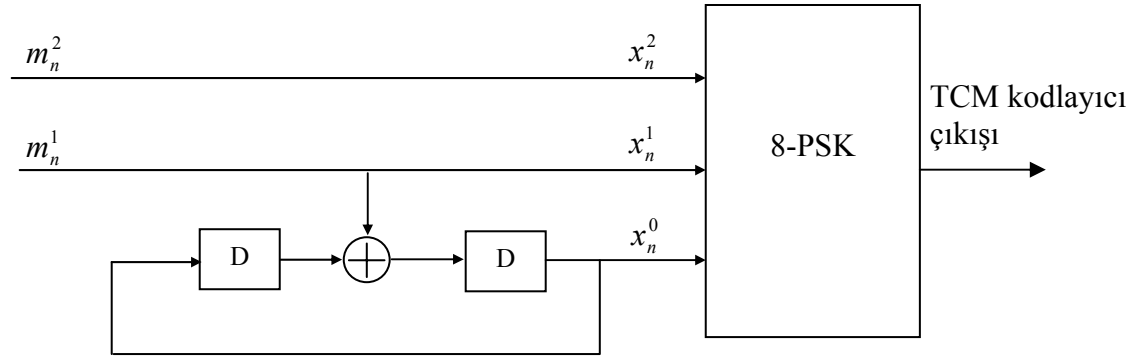


Şekil 2.18 2 bellekli 4 durumlu 8-PSK katlamalı kodlayıcı

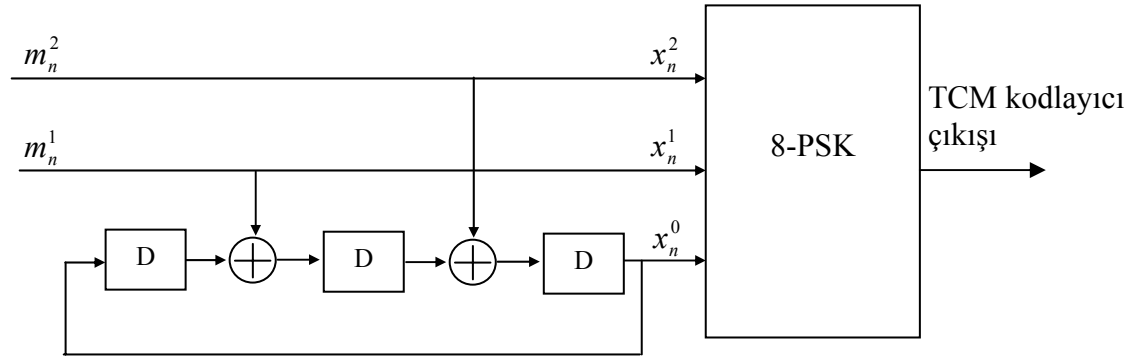


Şekil 2.19 3 bellekli 8 durumlu 8-PSK katlamalı kodlayıcı

Şekil 2.18 ve Şekil 2.19'daki katlamalı kodlayıcılara karşılık gelen geribeslemeli sistematik katlamalı kodlayıcı yapıları Şekil 2.20 ve 2.21'de verilmiştir.



Şekil 2.20 2 bellekli 4 durumlu 8-PSK geri beslemeli sistematik katlamalı kodlayıcı



Şekil 2.21 3 bellekli 8 durumlu 8-PSK geri beslemeli sistematik katlamalı kodlayıcı

2.1.9.2. Viterbi Algoritması

Viterbi algoritması kafes kodlayıcılar için en yaygın kod çözme algoritmasıdır. Şimdiye kadar [27,28] gibi bir çok sistem için kullanılmıştır. Bu algoritmanın temeli, mesaj bitlerini kodlayan kodlayıcının kafes yapısını belirli bir çerçevesi boyunca kesintisiz olarak takip eden ve Gauss dağılım fonksiyonunu minimum yapan yolu

bulmaktır. Bu sayede bir çerçeve boyunca kafes yapının tüm durumları elde edilmiş olur. Durumlar belirlendikten sonra bu durum geçişlerine karşılık gelen mesaj bitlerinin bulunmasında hiç bir zorluk yoktur.

Bu algoritmayı gerçeklemek için öncelikle alınan gürültülü işaret, olası tüm işaretlerle Gauss dağılımına sahip (2.22) denklemine bağlı olarak karşılaştırılır ve her biri için bir ölçekleyici (metrik) değeri elde edilir.

$$P(s_n) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(|s_n - s'|^2)}{\sigma^2}} \quad (2.22)$$

Burada s_n olması muhtemel gürültüsüz işaretleri, σ^2 ise gürültünün varyansını göstermektedir. 8-PSK modülasyon için n değeri 0 ile 7 arasındadır. s' ise alınan gürültülü işareti gösterir. Her durum için, alınan her bir gürültülü işarete karşı, denkleme (2.22) kullanılarak n adet ölçekleyici değeri elde edilir. Bu işlemten sonra her bir durum için o durumun oluşmasına neden olabilecek durum geçişlerine karşılık gelen ölçekleyici değerleri ile bunlara bağlı önceki durumların olasılık değerleri toplanarak minimum olanı alınır. Çerçevenin sonuna gelindiğinde Gauss fonksiyonunu minimum yapan yol bulunmuş olur. Bu bilgidен tüm giriş bitleri elde edilir.

Bu ölçekleyici değerinin elde edilmesinde işlem kolaylığı olması bakımından bir çok zaman sadece $|s_n - s'|^2$ kullanılır. Aynı zamanda sönümlemeli kanallarda kanal bilgisi elde edilmiş ise, alınan işareti karşılaştırılacak değerin seviyesine getirmek için

sönümleme değerine bölmek gerekir. Bu yöntem özellikle çok antenli sistemlerde etkilidir.

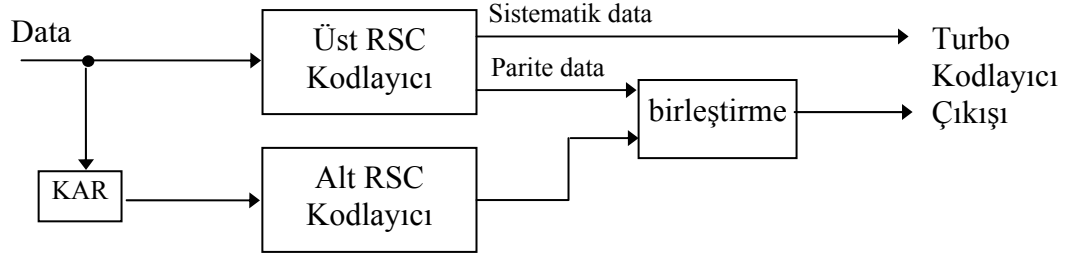
Kafes yapının ilk durumunun sıfır olduğu gözardı edilmemelidir. Aynı zamanda bir sonraki çerçevede ilk durumun sıfır olabilmesi için bir önceki çerçevenin sonuna bellek sayısı kadar anlamsız (*dummy*) bit eklenir.

2.2. Turbo Kodlama

Kablosuz haberleşmede, özellikle uydu haberleşmesinde sinyallerdeki bozulma ve bundan doğan iletişim problemleri çok önemlidir. Bu sorunu ortadan kaldırmanın en iyi yolu olarak 1993 yılında Beurro, Glavieux ve Thitimasjshima [17] Turbo kodlamayı buldular. Hata başarımının Shannon limite çok yakın olması bu kodlama yönteminin önemini arttırdı. Fakat bu yöntem ikili (binary) iletişimde geçerliydi. Bu problem de Blackert ve Wilson tarafından 1996 yılında çözülmüş ve Turbo kodlar kafesli kodlamalı modülasyona uygulanarak Turbo Kafes Kodlamalı Modülasyon (TTCM) [20] bulunmuştur.

2.2.1. Turbo Kodlayıcı

Turbo kodlayıcının genel yapısı Şekil 2.22’de verilmiştir. Turbo kodlayıcı iki tane genellikle aynı yapıdaki Geribeslemeli Sistematik Katlamalı (Recursive Systematic Convolutional, RSC) kodlayıcıdan oluşur. Her iki kodlayıcıda aynı datayı alır fakat ikinci kodlayıcı giriş datası karıştırıcıdan geçtikten sonra oluşan yeni dizilimli datayı alır. Turbo kodlarının rasgele gibi görünmesini sağlayan bu karıştırma işlemidir. Eğer karıştırıcının boyutu sabit ve her iki RSC kodlayıcının başlangıç durumları sıfırsa, bu durumdaki Turbo kodlara lineer blok kodlar denir.



Şekil 2.22. Turbo Kodlayıcı

Lineer blok kodlar için minimum uzaklık, olası kodların sıfır olmayan en küçük Hamming ağırlıdır. Karıştırıcı (KAR) ve RSC kodlayıcının birleşimi, turbo kodlayıcı tarafından oluşturulan kodların Hamming ağırlıklarının yüksek olmasını sağlar. Buna rağmen bazı giriş dizileri RSC kodlayıcının düşük ağırlıklı çıkış yaratmasına neden olabilir. Karıştırıcıdan dolayı her iki kodlayıcı giriş dizilerini aynı sıra ile almazlar. Bu nedenle, kodlayıcılardan birine düşük ağırlıklı çıkış yaratacak bir giriş olması halinde, diğer kodlayıcıya da düşük ağırlıklı çıkış yaratacak bir giriş olması pek olası değildir. Fakat bazı durumlarda her iki kodlayıcısında düşük ağırlıklı bir çıkış yaratacak giriş almaları durumunda Turbo kodların minimum uzaklıkları pek yüksek olmaz. İyi dizayn edilmiş bir Turbo kodlayıcı ile bu gibi düşük ağırlıklı kodlara az raslanır.

Turbo kodların performansı, düşük ağırlıklı kodların sayısının az olmasından dolayı, düşük sinyal gürültü oranlarında (Signal-to-Noise, SNR) iyidir. Buna rağmen, turbo kodların yüksek sinyal gürültü oranlarındaki başarımları minimum uzaklığın göreceli olarak daha az olmasından dolayı sınırlıdır. Geleneksel kodlama yöntemlerinin başarısı

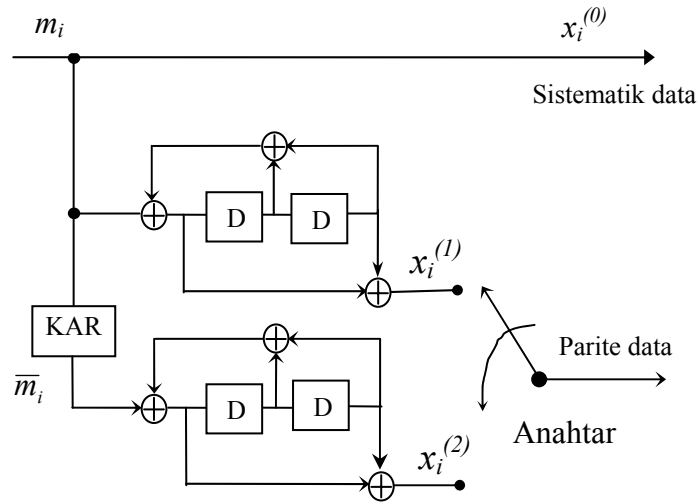
kodların minimum uzaklığını arttırmak, Turbo kodlamanın başarısı ise düşük ağırlıklı kodların sayısını azaltmaktır.

RSC kodlayıcılı bir Turbo kodlayıcı Şekil 2.23'de gösterilmektedir. Karıştırıcının i . girişe uyguladığı fonksiyonu $\alpha(i)$ dır, çıkışı $\bar{m}_i$ ile gösterilmektedir. Bu durumda giriş çıkış ilişkisi,

$$\bar{m}_{\alpha(i)} = m_i, \quad (2.23)$$

şeklinde gösterilir. Aynı şekilde düzenleyicinin giriş çıkış ilişkisi aşağıdaki gibidir.

$$m_{\alpha^{-1}(i)} = \bar{m}_i \quad (2.24)$$



Şekil 2.23 2 bellekli Turbo Kodlayıcı

Turbo kodlayıcının sistematik çıkışı $\mathbf{x}^{(0)}$ giriş datasının aynı sıralamasına sahiptir. $\mathbf{x}^{(1)}$ ve $\mathbf{x}^{(2)}$ parite dataları üst (birinci) ve alt (ikinci) RSC kolayıcılarının çıktılarıdır. $\frac{1}{2}$ oranlı Turbo kodlayıcısının çıkışı $(x_0^{(0)}, x_0^{(1)}, x_1^{(0)}, x_1^{(2)}, \dots, x_{L-2}^{(0)}, x_{L-2}^{(1)}, x_{L-1}^{(0)}, x_{L-1}^{(2)})$ şeklindedir.

2.2.2. Başarım Analizi

Beyaz Gauss gürültüsü eklenmiş (Additive White Gaussian Noise, AWGN) kanaldan geçen bir (n,k) lineer blok kodun bit hata olasılığı,

$$P_b \leq \sum_{i=1}^{2^k-1} \frac{w(\mathbf{m}_i)}{k} Q\left(\sqrt{w(\mathbf{x}_i) \frac{2rE_b}{N_0}}\right) \quad (2.25)$$

şeklindedir. Buradaki Q fonksiyonu

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^{\infty} e^{-x^2/2} dx \quad (2.26)$$

dir. E_b bit başına enerji ve N_0 tek taraflı gürültü yoğunluğudur. (2.25) denklemini 2^k-1 adet sıfır olamayan kodların ağırlıklarına ihtiyaç duyar. Aynı zamanda hem kompleks hem de küçük k değerleri için kullanımı engeller. Denklem (2.25)'deki terim sayısı aşağıdaki gibi aynı ağırlıklı mesaj kodlarını oluşturan terimleri birleştirerek oldukça azaltılabilir.

$$P_b \leq \sum_{d=d_{\min}}^n \frac{\bar{w}_d N_d}{k} Q\left(\sqrt{d \frac{2rE_b}{N_0}}\right) \quad (2.27)$$

Burada N_d , d ağırlıklı mesaj kod sayısı ve $\bar{w}_d$, N_d nin ortalama ağırlığıdır. Yüksek sinyal gürültü oranlarında, bit hata olasılığı (2.27) denkleminin ilk terimine yaklaşır,

$$P_b \approx \frac{\bar{w}_{d_{\min}} N_{d_{\min}}}{k} Q\left(\sqrt{d_{\min} \frac{2rE_b}{N_0}}\right), \quad (2.28)$$

buna da “minimum uzaklık asimptodu” denir.

Katlamalı kodlayıcıların asimptodik performansları göz önünde bulundurulduğunda, katlamalı kodlayıcıların blok uzunluğu sınırlandırılmamış olduğundan (2.27) denklemini kullanılamaz. Bunun yerine katlamalı kodlayıcıların maksimum olasılıklı (Maximum Likelihood, ML) kod çözümleri için

$$P_b = \sum_{d=d_{\text{serbest}}}^{\infty} W_d^0 Q\left(\sqrt{d \frac{2rE_b}{N_0}}\right), \quad (2.29)$$

kullanılır. Burada W_d^0 tüm mesaj kodlarının ağırlıklarının toplamıdır. Yüksek sinyal gürültü oranında, katlamalı kodların bit hata olasılığı serbest uzaklık asimptoduna yaklaşır.

$$P_b \approx W_{d_{\text{serbest}}}^0 Q\left(\sqrt{d_{\text{serbest}} \frac{2rE_b}{N_0}}\right) \quad (2.30)$$

2.2.3. Turbo Kod Çözme

Gürültülü elde edilmiş Markov sürecinin durum sırasını tahmin etmenin iki bilinen yöntemi vardır. Bunlardan birincisi *Viterbi algoritması* (VA) [3] diğeri de (sembol-sembol) *en büyük ileriye dönük* (MAP) [18] algoritmasıdır. Durum sırası tahmin edildikten sonra mesaj bitleri $\mathbf{m}$ veya ilgili mesaj kodları $\mathbf{x}$ in elde edilmesinde herhangi bir zorluk yoktur. VA ve MAP algoritmaları sağlamaya çalıştıkları kritere göre birbirlerinden ayrılırlar. Viterbi algoritması alınan $\mathbf{y}$ dizisine göre en olası durum dizisini bulur.

$$\hat{\mathbf{s}} = \arg \left\{ \max_{\mathbf{s}} P[\mathbf{s} | \mathbf{y}] \right\} \quad (2.31)$$

MAP algoritması ise, $\mathbf{y}$ dizisine göre her bir kodlama adımı için en olası durumu bulur.

$$\hat{s}_i = \arg \left\{ \max_{s_i} P[s_i | \mathbf{y}] \right\} \quad (2.32)$$

Bu iki algoritma arasındaki fark, VA ile kestirilen durumlar kafes yapı üzerinde birbirine bağlı yollar oluşturur fakat MAP algoritması ile kestirilen durumlar için böyle bir şeyden söz edilemez. Bundan dolayı VA çerçeve hata oranını (Frame Error Ratio,

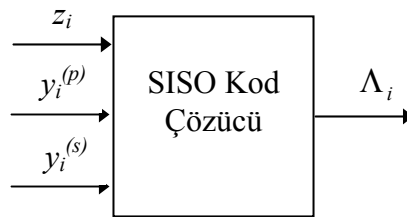
FER) minimize ederken, MAP algoritması bit hata oranını (Bit Error Ratio, BER) minimize eder

2.2.3.1. Yumuşak-Giriş Yumuşak-Çıkış (Soft-Input Soft-Output) Kod çözümü

Yumuşak bit kararı logaritmik olasılık oranı (log-likelihood ratio LLR) ile gösterilir. Kodlayıcının i . andaki girişi m_i , gürültülü olarak alınan dizi $\mathbf{y}$ olmak üzere, i . ana dair LLR kodlayıcı çıkışı Λ_i ,

$$\Lambda_i = \ln \frac{P[m_i = 1|\mathbf{y}]}{P[m_i = 0|\mathbf{y}]} \quad (2.33)$$

ile gösterilir. Ön (*a priori*) bilgiyi giriş olarak alan ve çıkışında ileriye dönük (*a posteriori*) bilgiyi üreten kod çözme algoritmasına yumuşak-girişli yumuşak-çıkışlı (Soft Input Soft Output, SISO) kod çözme algoritması denir. Şekil 2.24’de gösterildiği gibi $\frac{1}{2}$ oranlı bir RSC kodlayıcı için SISO kod çözücüsüne üç giriş gelir. Bunlar; sistematik bilgi $y_i^{(s)}$, parite data $y_i^{(p)}$ ve diğer kod çözücünden gelen ön bilgi z_i . SISO kod çözücüsünün çıkışı Λ_i (2.33) denklemi ile verilmiştir.



Şekil 2.24 $\frac{1}{2}$ oranlı RSC kodlayıcı için yumuşak-giriş yumuşak-çıkış (SISO) kod çözücü

Modülasyon için Şekil 2.25’de gösterilen İkili Faz Kaydırmalı Anahtarlama (BPSK) uygulanır. BPSK ile değeri 1 olan bilgiler 1 olarak, 0 olan bilgiler ise -1 olarak gönderilir.



Şekil 2.25 BPSK Modulator

Buna bağlı olarak elde edilen gürültülü işaret,

$$y' = \rho \sqrt{E_s} (2x - 1) + n' \quad (2.34)$$

olur. Burada ρ sönmleme katsayısı, $\sqrt{E_s} (2x - 1)$ BPSK modulator çıkışı ($x \in \{0,1\}$), E_s sembol başına enerji ve n' ortalaması sıfır varyansı $\sigma^2 = N_0 / 2E_s$ olan Gauss dağılımlı rasgele bir değişkendir. Eğer $\rho=1$ olursa kanala AWGN kanal denir. Aksi taktirde kanal sönmlemeli kanal olur ve kanalın tipi ρ nun dağılımına bağlıdır. En gerçekçi sönmlemeli kanal modelleri Rician ve Rayleigh'dır. Denklem (2.34) ün genelde kullanılan hali

$$y = \rho(2x - 1) + n \quad (2.35)$$

şeklinde gösterilir ve burada gürültünün varyansı $\sigma^2 = N_0 / 2E_s$ dir.

SISO kod çözücüsünün çıkışındaki LLR bu kanal modelini kullanarak üç terim ile gösterilir.

$$\Lambda_i = \frac{4\rho_i^{(s)}E_s}{N_0} y_i^{(s)} + z_i + l_i \quad (2.36)$$

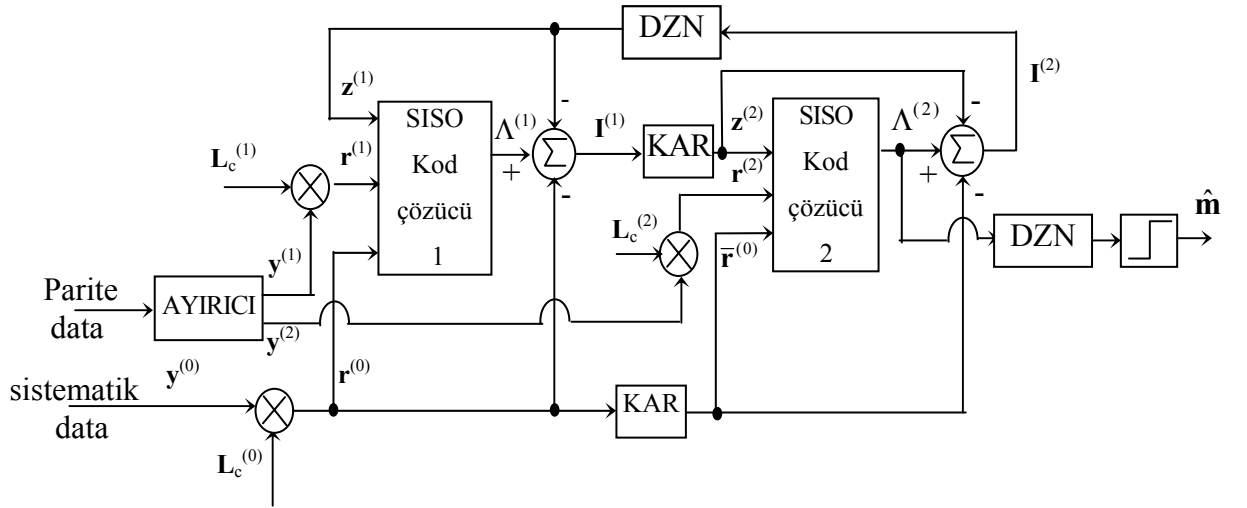
Burada l_i dışsal (*extrinsic*) bilgiyi göstermektedir. Eşitliğin sağ tarafındaki ilk iki terim alınan sistematik data ve diğer kodlayıcıdan alınan *ön* bilgi z_i ile ilgilidir, *dışsal* bilgi kod çözme işleminin o anki safhasından elde edilen yeni bilgiyi gösterir. Pozitif geribesleme problemini önlemek için bir kod çözücünden diğerine sadece *dışsal* bilginin gitmesi gerekir. Bu yüzden bir kod çözücünün girişindeki *ön* bilgisi veya diğeri tarafında hesaplanan *dışsal* bilgi aşağıdaki formülde verildiği gibi iki değer çıkarılması ile bulunur.

$$l_i = \Lambda_i - \frac{4\rho_i^{(s)}E_s}{N_0} y_i^{(s)} - z_i \quad (2.37)$$

Turbo kod çözücünün genel yapısı Şekil 2.26'da verilmiştir. İlk kod çözücüye sistematik kanal gözlemi $\mathbf{y}^{(0)}$ kanal güvenilirliği $\mathbf{L}_c^{(0)}$ ile çarpılarak gelir. $\mathbf{L}_c = \frac{4\rho E_s}{N_0}$ olarak tanımlanır. Aynı şekilde $\mathbf{y}^{(1)}$ de $\mathbf{L}_c^{(1)}$ ile çarpılarak ilk kod çözücüye girer. Bir de *ön* bilgi $\mathbf{z}^{(1)}$ de ilk kod çözücüye girer. İlk kod çözücü $\Lambda^{(1)}$ çıkışı verir. İlk kod çözücünün *dışsal* bilgisi $\mathbf{I}^{(1)}$, $\Lambda^{(1)}$ den kanal güvenilirliği ile çarpılmış sistematik bilginin ve *ön* bilginin çıkarılması ile bulunur. *Dışsal* bilgi karıştırılır ve diğer kod

çözücü için *ön* bilgi olarak kullanılır. Diğer kod çözücüye karıştırılmış ve $\mathbf{L}_c^{(0)}$ ile çarpılmış sistematik kanal gözlemi ile $\mathbf{L}_c^{(2)}$ ile çarpılmış ikinci kodlayıcının parite datası $\mathbf{r}^{(2)}$ girer. İkinci kod çözücü $\Lambda^{(2)}$ yi üretir, ağırlıklı sistematik ve *ön* bilginin çıkarılması ile $\mathbf{I}^{(2)}$ elde edilir. İkinci kod çözücü tarafından elde edilen *dışsal* bilgi düzenlenerek bir sonraki iterasyonda birinci kod çözücüde *ön* bilgi olarak kullanılır. Gerekli iterasyon sayısına ulaşıldığında ikinci kod çözücünün çıkışı düzenlenip katı-karar dan geçirilerek bilgi bitleri tahmin edilmiş olur.

$$\hat{m}_i = \begin{cases} 1 & \Lambda_{\alpha(i)}^{(2)} \geq 0 \\ 0 & \Lambda_{\alpha(i)}^{(2)} < 0 \end{cases} \quad (2.38)$$



Şekil 2.26 Turbo Kod Çözücü

2.2.3.2 En Büyük İleriye Dönük Algoritma

Sembol-sembol MAP algoritması 1974 yılında Bahl et al tarafından Viterbi algoritmasına alternatif olarak sunulmuştu [18]. MAP algoritmasının *ileri-geri* versiyonuna [29]'dan sonra tip-I MAP algoritması ve *sadece-ileri* versiyonuna da tip-II MAP algoritması denir. Tip-II tip-I den çok daha karmaşık ve daha fazla hafızaya ihtiyaç duyar. Aynı zamanda Turbo kodlama blok yönlendirmeli bir işlem olduğu için tip-I MAP algoritması kullanılır.

MAP algoritması her durumgeçişi, mesaj biti, Markov süreci ile yaratılan semboller için gözlemlenen gürültülü $\mathbf{y}$ 'lerden *ileriye dönük* (a posteriori) olasılığını (APP) hesaplar. İstenen her olası değer için APP hesaplandıktan sonra katı-karar uygulanarak en yüksek olasılıklı olan seçilir. Turbo kodlamada MAP algoritması mesaj bitlerinin APP lerini hesaplar, $P[m_i = 1 | \mathbf{y}]$ ve $P[m_i = 0 | \mathbf{y}]$. Turbo kodlamada kullanılan MAP algoritması son iterasyona kadar katı-karar uygulamaz.

Mesaj bitlerin APP leri bulmadan önce MAP algoritması gözlemlenen gürültülü $\mathbf{y}$ ile olası her durum geçişi için $P[s_i \rightarrow s_{i+1} | \mathbf{y}]$ olasılığını hesaplar. Koşullu olasılık tanımından

$$P[s_i \rightarrow s_{i+1} | \mathbf{y}] = \frac{P[s_i \rightarrow s_{i+1}, \mathbf{y}]}{P[\mathbf{y}]} . \quad (2.39)$$

Markov sürecinin özellikleri kullanılarak (2.39) denkleminin pay kısmı şu şekilde bölümlenebilir,

$$P[s_i \rightarrow s_{i+1}, \mathbf{y}] = \alpha(s_i) \gamma(s_i \rightarrow s_{i+1}) \beta(s_{i+1}). \quad (2.40)$$

Burada,

$$\alpha(s_i) = P[s_i, (y_0, \dots, y_{i-1})] \quad (2.41)$$

$$\gamma(s_i \rightarrow s_{i+1}) = P[s_{i+1}, y_i | s_i] \quad (2.42)$$

$$\beta(s_{i+1}) = P[(y_{i+1}, \dots, y_{L-1}) | s_{i+1}]. \quad (2.43)$$

$\gamma(s_i \rightarrow s_{i+1})$ terimi, s_i den s_{i+1} durumuna geçiş olasılığı dal ölçekleyicisidir ve

$$\begin{aligned} \gamma(s_i \rightarrow s_{i+1}) &= P[s_{i+1} | s_i] P[y_i | s_i \rightarrow s_{i+1}] \\ &= P[m_i] P[y_i | x_i] \end{aligned} \quad (2.44)$$

şeklindedir. Burada m_i ve x_i , s_i den s_{i+1} duruma geçerkenki mesaj ve çıkış değerleridir.

Eğer s_i den s_{i+1} 'e kafes yapıda bir geçiş yoksa (2.44) denklemini sıfıra eşittir.

Eşitlik (2.44)'ün ilk terimi, $P[m_i]$, ön bilgidir ve aşağıdaki gibi tanımlanır.

$$P[m_i] = \begin{cases} \frac{e^{z_i}}{1 + e^{z_i}} & m_i = 1 \\ \frac{1}{1 + e^{z_i}} & m_i = 0 \end{cases} \quad (2.45)$$

ve logaritmik hesaplama için

$$\ln P[m_i] = z_i m_i - \ln(1 + e^{z_i}). \quad (2.46)$$

Eşitlik (2.44)'ün ikinci terimi kanal modeli ve modülasyonun bir fonksiyonudur, aşağıdaki gibi tanımlanır.

$$P[y_i | x_i] = \frac{1}{\sqrt{\pi N_0 / E_s}} \exp \left\{ \frac{-E_s}{N_0} \sum_{q=0}^{n-1} [y_i^q - \rho_i^{(q)}(2x_i^{(q)} - 1)]^2 \right\} \quad (2.47)$$

$\alpha(s_i)$ olasılığı, ileri yineleme ile

$$\alpha(s_i) = \sum_{s_{i-1} \in A} \alpha(s_{i-1}) \gamma(s_{i-1} \rightarrow s_i), \quad (2.48)$$

burada A , s_i ' lere bağlı s_{i-1} ' lerin kümesini gösterilmektedir. Buna benzer şekilde $\beta(s_i)$ de, geri yineleme ile bulunur.

$$\beta(s_i) = \sum_{s_{i+1} \in B} \beta(s_{i+1}) \gamma(s_i \rightarrow s_{i+1}), \quad (2.49)$$

burada B, s_i lere bağılı s_{i+1} lerin kümesini göstermektedir.

Her durum geçişi için $P[s_i \rightarrow s_{i+1} | \mathbf{y}]$ APP hesaplandıktan sonra mesaj biti olasılıkları

$$P[m_i = 1 | \mathbf{y}] = \sum_{S_1} P[s_i \rightarrow s_{i+1} | \mathbf{y}], \quad (2.50)$$

$$P[m_i = 0 | \mathbf{y}] = \sum_{S_0} P[s_i \rightarrow s_{i+1} | \mathbf{y}] \quad (2.51)$$

bulunur, burada $S_1 = \{s_i \rightarrow s_{i+1} | m_i=1\}$ mesaj bitinin 1 olması durumundaki tüm durum geçişlerini ve $S_0 = \{s_i \rightarrow s_{i+1} | m_i=0\}$ mesaj bitinin 0 olması durumundaki tüm durum geçişlerini gösterir. Sonrasında LLR çıkışı verir.

$$\Lambda_i = \ln \frac{\sum_{S_1} \alpha(s_i) \gamma(s_i \rightarrow s_{i+1}) \beta(s_{i+1})}{\sum_{S_0} \alpha(s_i) \gamma(s_i \rightarrow s_{i+1}) \beta(s_{i+1})} \quad (2.52)$$

Eşitlik (2.39)'daki $P[\mathbf{y}]$ terimi, LLR ile oranlama yapıldığında birbirini götürceğinden ayrıca hesaplanmaz.

2.2.3.3. Max-log-MAP ve log-MAP Algoritmaları

MAP algoritması her bitin APP'sini kusursuz olarak hesaplayabilmesine rağmen bazı pratik uygulamalarda önemli problemleri vardır. Bunlardan birincisi, yoğun bir işlem çokluğuna gereksinmesi, ikincisi ise hesaplama sonuçlarının sonlu hassasiyette gösterilmeye çalışıldığından taşma yapması veya işlem sonucunun sonsuza gitmesidir. Bu iki sebep işlemlerin logaritmik domende yapılmasını gerektirmektedir. Bu sayede çarpma işlemleri toplamaya dönüşmektedir. Buna rağmen logaritmik domende toplama işlemine pek başvurulmaz. Logaritmik domende toplama işleminin nasıl yapıldığını göstermek için öncelikle *Jacobian logaritması* göz önüne alınır.

$$\begin{aligned}\ln(e^x + e^y) &= \max(x, y) + \ln(1 + \exp\{-|y - x|\}) \\ &= \max(x, y) + f_c(|y - x|)\end{aligned}\quad (2.53)$$

Bu formül ile logaritmik domende toplama işlemi maksimum seçme ve bir düzeltme fonksiyonu $f_c(\cdot)$ ile toplama halini alır. Eğer x ve y değerleri birbirlerinden çok farklı ise düzeltme fonksiyonu sıfıra çok yakın bir değer alır, bu durumda

$$\ln(e^x + e^y) \approx \max(x, y) \quad (2.54)$$

değerini alır.

MAP algoritmasının logaritmik domende çalışan iki türü vardır. Bunlar *max-log-MAP* ve *log-MAP* dir [30]. Aralarındaki fark logaritmik domende toplama işleminin nasıl yapıldığıdır. Max-log-MAP algoritmasında, toplama işlemi (2.54) deki maksimum

seçme ile yapılır. Log-Map algoritmasında ise (2.53) kullanılarak maksimum seçmenin ardında düzeltme fonksiyonu hesaplanır. Bu düzeltme fonksiyonu iki sayının farkına bağlı olduğu için bir tablodan okunarak kullanılabilir.

$\bar{\gamma}(s_i \rightarrow s_{i+1})$, $\gamma(s_i \rightarrow s_{i+1})$ nin doğal logaritması olsun

$$\bar{\gamma}(s_i \rightarrow s_{i+1}) = \ln \gamma(s_i \rightarrow s_{i+1}) \quad (2.55)$$

$$= P[m_i] + P[y_i | x_i] \quad (2.56)$$

ve bu durumda durum geçiş ölçekleyicisinin logaritması

$$\bar{\gamma}(s_i \rightarrow s_{i+1}) = \ln P[m_i] - \frac{1}{2} \ln(\pi N_0 / E_s) - \frac{E_s}{N_0} \sum_{q=0}^{n-1} [y_i^{(q)} - (2x_i^{(q)} - 1)]^2. \quad (2.57)$$

Şimdi $\bar{\alpha}(s_i)$, $\alpha(s_i)$ nin doğal logaritması olsun

$$\bar{\alpha}(s_i) = \ln \alpha(s_i) \quad (2.58)$$

$$\bar{\alpha}(s_i) = \ln \left\{ \sum_{s_{i-1} \in A} \exp[\bar{\alpha}(s_{i-1}) + \bar{\gamma}(s_{i-1} \rightarrow s_i)] \right\} \quad (2.59)$$

$$= \max_{s_{i-1} \in A}^* [\bar{\alpha}(s_{i-1}) + \bar{\gamma}(s_{i-1} \rightarrow s_i)] \quad (2.60)$$

burada A, s_i' lere bağılı s_{i-1}' lerin kümesini gösterilmektedir. Max-log-MAP algoritması için $\max^*(x,y)=\max(x,y)$, log-MAP algoritması içinse $\max^*(x,y)=\max(x,y)+f_c(|y-x|)$ dir.

İleri yineleme işlemine benzer olarak $\bar{\beta}(s_i)$, $\beta(s_i)$ 'in doğal logaritması olsun,

$$\bar{\beta}(s_i) = \ln \beta(s_i) \quad (2.61)$$

$$\bar{\beta}(s_i) = \ln \left\{ \sum_{s_{i+1} \in B} \exp[\bar{\beta}(s_{i+1}) + \bar{\gamma}(s_i \rightarrow s_{i+1})] \right\} \quad (2.62)$$

$$= \max_{s_{i+1} \in B}^* [\bar{\beta}(s_{i+1}) + \bar{\gamma}(s_i \rightarrow s_{i+1})] \quad (2.63)$$

burada B, s_i' lere bağılı s_{i+1}' lerin kümesini göstermektedir.

Kafes yapıdaki tüm durumlar için $\bar{\alpha}(s_i)$ ve $\bar{\beta}(s_i)$ hesaplandıktan sonra LLR bulunur.

$$\Lambda_i = \ln \left\{ \sum_{s_1} \exp[\bar{\alpha}(s_i) + \bar{\gamma}(s_i \rightarrow s_{i+1}) + \bar{\beta}(s_{i+1})] \right\} \\ - \ln \left\{ \sum_{s_0} \exp[\bar{\alpha}(s_i) + \bar{\gamma}(s_i \rightarrow s_{i+1}) + \bar{\beta}(s_{i+1})] \right\} \quad (2.64)$$

Bu denklemde logaritma işlemlerini maksimum seçme işlemi olarak düşünebileceğimizden (2.64) denklemi aşağıdaki şekle dönüşür.

$$\begin{aligned} \Lambda_i = & \max_{S_1}^* [\bar{\alpha}(s_i) + \bar{\gamma}(s_i \rightarrow s_{i+1}) + \bar{\beta}(s_{i+1})] \\ & - \max_{S_0}^* [\bar{\alpha}(s_i) + \bar{\gamma}(s_i \rightarrow s_{i+1}) + \bar{\beta}(s_{i+1})] \end{aligned} \quad (2.65)$$

2.3. Turbo Kafes Kodlamalı Modülasyon

Turbo kafes kodlamalı modülasyon (Turbo Trellis Coded Modulation, TTCM) ilk olarak Blackert ve Wilson tarafından [20]'de sunulmuştur. Daha sonra 1996'da kafes kodlama için kullanılan MAP algoritması Robertson ve Worz tarafından [21]'de ayrıntılı olarak verilmiştir.

2.3.1. Turbo Kafes Kodlayıcı

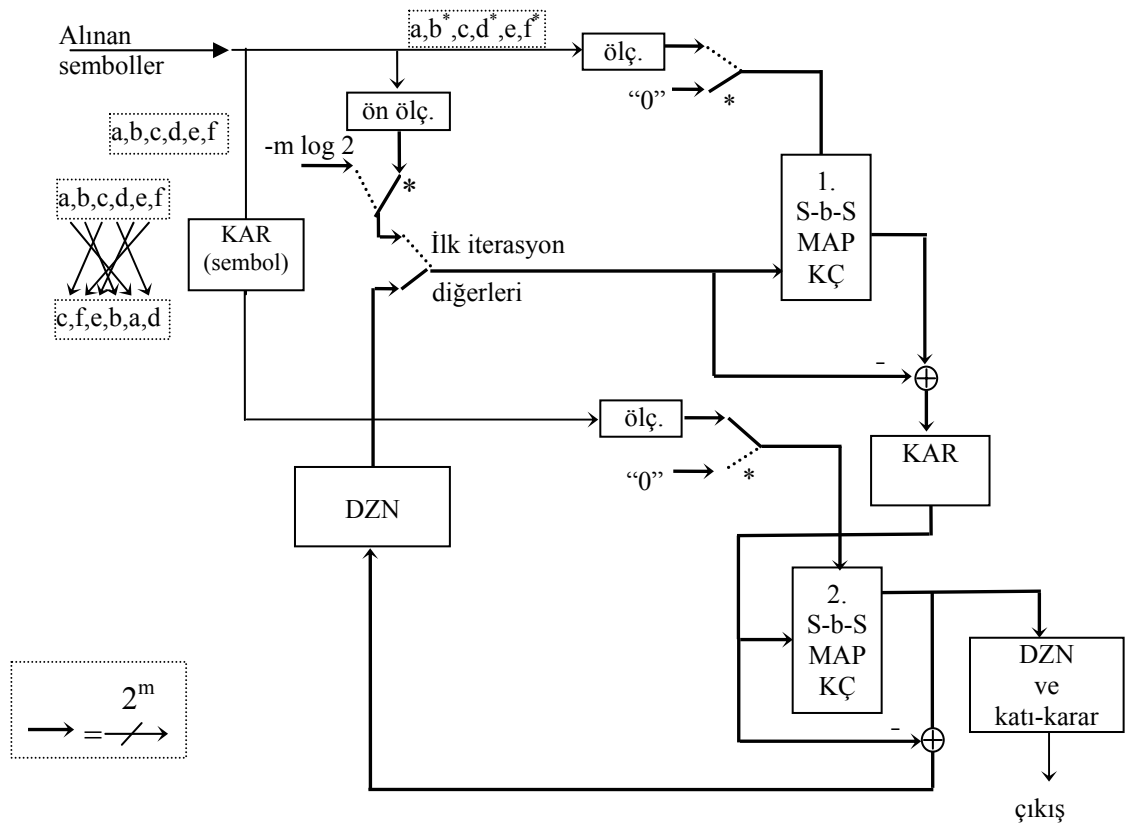
Turbo Kafes Kodlamalı Modülasyonda kullanılan kodlayıcının (Şekil 2.27) yapısı Turbo kodlayıcıdan biraz farklıdır. Bunlar; TTCM'de girişin en az 2 olması, girişlerin kodlanıp bir sinyale eşlenmesi, kullanılan karıştırıcı ve düzenleyicilerin tekleri teke, çiftleri de çifte koyması gerekmesidir. Tabi ki bu durum kod çözme işleminde birtakım farklılıklar yaratacaktır ama yine de kod çözme işlemi Turbo Kodlama ile benzer sayılabilir.

2.3.2. Faz Kaydırmalı Modülasyon

Şekil 2.27'de verilen kodlayıcı yapısında iki giriş ($m=2$) biti ve üç çıkış biti olduğu görülmektedir. Kodlayıcının iki bitlik bir " d " girişine karşılık, çıkışındaki " $b_0 b_1 b_2$ " ile gösterilen bu üç bit Şekil 2.28'de gösterildiği gibi kompleks düzlemede eşlenir.

2.3.3. Turbo Kafes Kod Çözücü

Turbo Kafes Kod çözücü Şekil 2.29’da gösterilmiştir. Burada görüldüğü gibi alınan işaretler 1. ve 2. S-b-S (Symbol by Symbol) MAP kod çözümüsüne (KÇ) gönderilir. Turbo kodlamadan farklı olarak ilk iterasyon sırasında herhangi bir *ön* bilgi olmadığı için ön ölçekleyici ile çift indisli işaretler içinde bir olasılık hesabı yapılır ve bu 1. S-b-S MAP kod çözümüsünde kullanılır. Diğer iterasyonlarda 2. kod çözümü bilgi üretmiş olduğu için ön ölçekleyici kullanılmaz.



Şekil 2.29 TTCM Kod Çözücü

Bir diğerk fark ise bir kod çözücünden diğerk kod çözücüye her bir işaret için giriş bitlerinin ihtimali kadar bilginin gitmesidir. Yani 2 girişli bir kodlayıcı kullanılırsa bir kod çözücünden diğerkine 4 bilgi gidecektir. Bunun sebebi de girişin tek olmamasıdır. Bu nedenle çıkışta karar mekanizmasında bir logaritmik olasılık oranı (LLR) kullanılamaz. Sadece bu dört durumdan en yüksek ihtimalli olanı seçebilebilir.

İteratif kod çözücü ikili kod çözücüye benzer fakat bazı farklılıkları vardır. Bunlar bir kod çözücünden diğerk kod çözücüye giden bilginin biçiminin farklı olması ve ilk kod çözme adımındaki davranışın farklılığından kaynaklanır. Bilgi bitleri, diğerk bir değışle sistematik bitler, kodlanan parite biti ile birlikte eşleyici tarafından bir işarete karşı düşürölürler. Her kod çözücü kendi kodlayıcısından gelen bilgiyi çözer yani tek indisli olanlar birinci kod çözücüde, çift indisli olanlar ise ikinci kod çözücü tarafından çöölürler. Her kod çözücü için diğerk kodlayıcıdan gelen bilgi “*” asterisk işareti ile gösterilir ve bu sembollere “delikli” (punctured) denir.

İkili Turbo kodlayıcılarında, her k bilgi biti için, kod çözücülerin çıkışları toplanmış üç parçadan oluşmuştur. Bunlar; sistematik bileşen, ön bileşen ve dışsal bileşendir. Sadece dışsal bileşen diğerk kod çözücüye verilmelidir, aksi takdirde, bilgi diğerk kod çözücünden birden fazla defa kullanılacaktır bu da ileri yığılmaya neden olacaktır.

Turbo Kafes Kodlamalı Modülasyonda ise durum daha karmaşıktır, çünkü sistematik bileşen ikili Turbo kodlamadan farklı olarak *dışsal* bileşenden ayrılamaz. Sistematik bilgi parite bilgi ile aynı sembol üzerinden gönderilir. Buna rağmen çıkışı iki parçaya bölebiliriz; 1-*ön* 2-*dışsal* ve sistematik. Fakat *dışsal* ve sistematik bileşenleri ayırmak mümkün değildir.

2.3.3.1. Dışsal, Ön ve Sistematik Bileşenler

Bu kısımda iteratif kod çözücü anlatılacaktır. Burada olasılıkların logaritmik ifadeleri ($\log \Pr(\cdot)$) kullanılacak ve bu da $L(\cdot)$ ile gösterilecektir. Sistematik ve *dışsal* bileşenler bir kod çözücünden çıkar ve diğerine gönderilir ve o kod çözücünde *ön* bilgi olarak kullanılırlar. MAP algoritmasının çıkışını sistematik ve *dışsal* bileşen olarak *ön* bilgi ($\Pr\{d_k = i\}$)'den bağımsız tanımlayabiliriz. *Ön* bilgi,

$$L_p(d_k = i) = \log \Pr\{d_k = i\} \quad (2.66)$$

olarak tanımlanır ve bu MAP algoritmasının sonucundan çıkartılarak,

$$L_{e\&s}(d_k = i) = \log \Pr\{d_k = i | \underline{\mathbf{y}}\} - \log \Pr\{d_k = i\} \quad \forall i \in \{0 \dots 2^m - 1\} \quad (2.67)$$

sistematik ve *dışsal* bileşen elde edilmiş olur. Bu bilgi bir sonraki kodçözücünde *ön* bilgi olarak kullanılır.

2.3.3.2. Ön Ölçekleyici Hesaplaması

Yöntemi kullanabilmek için *ön* bilgiyi çıkartmamız gerekmektedir. Fakat ilk iterasyonun ilk adımında *ön* bilgidan bahsetmek mümkün değildir. Bu yüzden birinci kod çözücü işleme başlamadan önce *ön* bilgiyi elde etmemiz gerekmektedir. Bu şekilde *dışsal* bilginin birinci kod çözücünden ikinci kod çözücüye gönderilirken * geçişlerinde sistematik bilgi içerebilmesi sağlanır. Burada gönderilen sembol bilgi grubu d_k ile kısmen, bunun yanısıra diğer kodlayıcı tarafından oluşturulan bilinmeyen parite bit $b^{0,*}$ $\in \{0,1\}$ tarafından saptanır. Bu yüzden *ön* bilgi birleşik Bayes kuralı ile saptanır,

$$\begin{aligned} \Pr\{d_k = i \mid y_k\} &= \text{const} \cdot p(\mathbf{y}_k \mid d_k = i) \\ &= \text{const} \cdot \sum_{j \in \{0,1\}} p(\mathbf{y}_k, b_k^{0,*} = j \mid d_k = i) \\ &= \frac{\text{const}}{2} \cdot \sum_{j \in \{0,1\}} p(\mathbf{y}_k \mid d_k = i, b_k^{0,*} = j) \end{aligned} \quad (2.68)$$

Burada $\Pr\{b_k^{0,*} = j \mid d_k\} = \Pr\{b_k^{0,*} = j\} = 1/2$ olduğu varsayılır. Başka bir deyişle, x_k içindeki parite bitin bilgi bit grubundan istatistiksel olarak bağımsızdır ve bu durumda parite bitin 0 ya da 1 olma olasılığı eşittir. Ayrıca d_k 'nın ilk *ön* olasılığı tüm i için sabittir. Yukarıdaki sabit değeri hesaplamaya gerek yoktur, tüm i ler üzerinden toplanıp normalize edilebilir. Eğer üst kod çözücü * geçişinde değilse $\Pr\{d_k = i\}$ $1/2^m$ değerine eşitlenir.

2.3.3.3. TCM için MAP Algoritması

Bu kısımda, ikili olmayan kafes yapıli kodlayıcılar için sembol-sembol MAP algoritması açıklanacaktır. Burada klasik Kafes Kodlamalı Modülasyon yapıları düşünölmüştür. Kodlayıcıda v tane bellek olduđu düşünölürse, 2^v tane durum sözkonusudur. Kodlayıcının k anındaki durumu, $S_k \in \{0,1,\dots,2^v-1\}$ şeklinde gösterilir. Kodlayıcıya giren bilgi bitlerinin oluşturduđu m tane bilgi biti, $(0\dots 2^m-1)$ arasındaki tam sayılarla gösterilir ve giriş bitlerine göre $k-1$ 'inci adımdan k 'ıncı adıma geçişı belirler. Karıştırıcı boyutunun N olduđu düşünölürse, blok başına modüle edilen sembol sayısı $N \times n$ dir. Burada n , gönderilen sinyalin boyutunun yarısıdır. Bu durumda blok başına gönderilen bilgi biti sayısı $N \times m$ dir. Alıcı N set n güröltölü sembol alır ve $\mathbf{y}_k = (y_k^0, \dots, y_k^{(n-1)})$ şeklinde gösterilir. Toplam alınan dizi $\underline{\mathbf{y}} = \underline{\mathbf{y}}_1^N = (\mathbf{y}_1, \dots, \mathbf{y}_N)$ olur. Bu, Kafes Kodlamalı Modülasyon çıkışına $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ tek taraflı güç spektral yoğunluğu $N_0/2$ olan beyaz Gauss güröltüsü eklenmiş halidir. Her $\mathbf{x}_k = (x_k^0, \dots, x_k^{(n-1)})$ k anında eşleyiciden çıkan n sembollü bir gruptur.

Kod çözücünün yapması gereken, her d_k ve k için $\Pr\{d_k | \underline{\mathbf{y}}_1^N\}$ 'yı hesaplamaktır. İleri ve geri olasılıklar şu şekilde verilir.

$$\alpha_{k-1}(M') = \frac{p(S_{k-1} = M', \underline{\mathbf{y}}_1^{k-1})}{p(\underline{\mathbf{y}}_1^{k-1})} \quad (2.69)$$

$$\beta_k(M) = \frac{p(\underline{\mathbf{y}}_{k+1}^N | S_k = M)}{p(\underline{\mathbf{y}}_{k+1}^N | \underline{\mathbf{y}}_1^k)} \quad (2.70)$$

k adımı için dal geçiş olasılığı $p(d_k=i, y_k, S_k=M \mid S_{k-1}=M')$ dir ve aşağıdaki şekilde hesaplanır.

$$\begin{aligned} \gamma_i(y_k, M', M) &= p(y_k \mid d_k = i, S_k = M, S_{k-1} = M') \\ &\cdot q(d_k = i \mid S_k = M, S_{k-1} = M') \\ &\cdot \Pr\{S_k = M \mid S_{k-1} = M'\} \end{aligned} \quad (2.71)$$

$q(d_k = i \mid S_k = M, S_{k-1} = M')$ 0 ya da 1 olabilir ve bu kodlayıcı girişi $i \in \{0, 1, \dots, 2^m-1\}$ nin $S_{k-1}=M'$ den $S_k=M$ 'ye geçişi ile ilgili olup olmadığına bağlıdır. (2.71) formülünün üçüncü çarpanını hesaplamak için *ön* bilgi kullanılır.

$$\begin{aligned} &\Pr\{S_k = M \mid S_{k-1} = M'\} \\ &= \begin{cases} \Pr\{d_k = 0\}, & q(d_k = 0 \mid S_k = M, S_{k-1} = M') = 1 \\ \Pr\{d_k = 1\}, & q(d_k = 1 \mid S_k = M, S_{k-1} = M') = 1 \\ \Pr\{d_k = 2^m - 1\}, & q(d_k = 2^m - 1 \mid S_k = M, S_{k-1} = M') = 1 \end{cases} \\ &= \Pr\{d_k = j\} \end{aligned} \quad (2.72)$$

Burada j : $q(d_k=j \mid S_k=M, S_{k-1}=M')=1$. Eğer $q(d_k=j \mid S_k=M, S_{k-1}=M')=1$ yapacak bir j yoksa $\Pr\{S_k=M \mid S_{k-1}=M'\}$ değeri 0 olur.

$(d_k=i, y_k, S_{k-1}=M')$ olayı için, S_k nın bilinmesi durumunda, $\underline{\mathbf{y}}_{k+1}^N$ in hiç bir etkisi yoktur.

Bundan dolayı k anı ve önceki durumların bilindiği düşünülerek bir sonraki girişe ilişkin geçiş olasılığı,

$$p(\underline{\mathbf{y}}_{k+1}^N | S_k = M) = p(\underline{\mathbf{y}}_{k+1}^N | d_k = i, \mathbf{y}_k, S_k = M, S_{k-1} = M') \quad (2.73)$$

ile, k anı ve sonraki durumların bilindiği düşünülerek bir önceki girişin olasılığı,

$$p(\underline{\mathbf{y}}_1^{k-1}) = \frac{p(\underline{\mathbf{y}}_1^k)}{p(\mathbf{y}_k | \underline{\mathbf{y}}_1^{k-1})} \quad (2.74)$$

kullanılarak ileri, geri ve dal geçiş olasılıklarının çarpımı aşağıdaki gibi olur.

$$\begin{aligned} \alpha_{k-1}(M') \cdot \beta_k(M) \cdot \gamma(\mathbf{y}_k, M', M) &= p(S_{k-1} = M', \underline{\mathbf{y}}_1^{k-1}) \\ &\cdot p(\underline{\mathbf{y}}_{k+1}^N, d_k = i, \mathbf{y}_k, S_k = M | S_{k-1} = M') \quad (2.75) \\ &\cdot \frac{p(\mathbf{y}_k | \underline{\mathbf{y}}_1^{k-1})}{p(\underline{\mathbf{y}}_1^N)} \end{aligned}$$

Açık olarak yazılırsa, (2.75)'in ikinci çarpanı $p(\underline{\mathbf{y}}_{k+1}^N, d_k, \mathbf{y}_k, S_k = M | S_{k-1} = M') =$

$p(\underline{\mathbf{y}}_{k+1}^N, d_k, \mathbf{y}_k, S_k = M | S_{k-1} = M', \underline{\mathbf{y}}_1^{k-1})$ dir ve bu durumdan dolayı ileri, geri ve dal geçiş

olasılıklarının çarpımı aşağıdaki şekilde yazılabilir.

$$\begin{aligned} \alpha_{k-1}(M') \cdot \beta_k(M) \cdot \gamma(\mathbf{y}_k, M', M) \cdot \frac{1}{p(\mathbf{y}_k | \underline{\mathbf{y}}_1^{k-1})} &= p(S_{k-1} = M', S_k = M, \underline{\mathbf{y}}_1^N) \cdot \frac{1}{p(\underline{\mathbf{y}}_1^N)} \quad (2.76) \\ &= p(S_{k-1} = M', S_k = M | \underline{\mathbf{y}}_1^N) \end{aligned}$$

Buna göre MAP algoritmasının çıkışı aşağıdaki gibi yazılabilir.

$$P_r\{d_k = i | \underline{\mathbf{y}}\} = \text{sabit} \cdot \sum_M \sum_{M'} [\gamma_i(y_k, M', M) \cdot \alpha_{k-1}(M') \cdot \beta_k(M)] \quad \forall i \in \{0 \dots 2^m - 1\} \quad (2.77)$$

Buradaki sabit, denklemi tüm i üzerinden toplamını alıp normalize ederek ortadan kaldırılabılır. $\Pr\{d_k | \underline{\mathbf{y}}\}$ ön, sistematik ve dışsal bigisini taşır, çünkü bu d_k nın ön olasılığında olduğu gibi alınan tüm diziye bağlıdır.

Şimdi geriye $\alpha_{k-1}(M')$ ve $\beta_k(M)$ in yenilenerek bulunması kaldı. İlk olarak,

$$\Pr\{S_k = M | \underline{\mathbf{y}}_1^{k-1}, \mathbf{y}_k\} \cdot p(\mathbf{y}_k | \underline{\mathbf{y}}_1^{k-1}) = p(\mathbf{y}_k, S_k = M | \underline{\mathbf{y}}_1^{k-1}) \quad (2.78)$$

ele alalım, bu denklemin her iki tarafını $p(\mathbf{y}_k | \underline{\mathbf{y}}_1^{k-1})$ 'e bölersek aşağıdaki şekle açabiliriz.

$$\Pr\{S_k = M | \underline{\mathbf{y}}_1^k\} = \alpha_k(M) = \frac{\sum_{M'} p(\mathbf{y}_k, S_k = M, S_{k-1} = M' | \underline{\mathbf{y}}_1^{k-1})}{\sum_M \sum_{M'} p(\mathbf{y}_k, S_k = M, S_{k-1} = M' | \underline{\mathbf{y}}_1^{k-1})} \quad (2.79)$$

ve denklem (2.77)'den dolayı

$$\alpha_k(M) = \frac{\sum_{M'} \Pr\{\mathbf{y}_k, S_k = M | S_{k-1} = M'\} \cdot p(S_{k-1} = M' | \underline{\mathbf{y}}_1^{k-1})}{\sum_M \sum_{M'} p(\mathbf{y}_k, S_k = M | S_{k-1} = M') \cdot \Pr\{S_{k-1} = M' | \underline{\mathbf{y}}_1^{k-1}\}} \quad (2.80)$$

şeklinde yazılabilir. Tüm olası giriş bitleri için dal geçiş olasılığı

$$\gamma_T(\mathbf{y}_k, M', M) = \sum_{i=0}^{2^m-1} \gamma_i(\mathbf{y}_k, M', M) \quad (2.81)$$

şeklinde tanımlanır. Bu durumda ileri ve geri olasılıklar,

$$\alpha_k(M) = \frac{\sum_{M'} \gamma_T(\mathbf{y}_k, M', M) \cdot \alpha_{k-1}(M')}{\sum_M \sum_{M'} \gamma_T(\mathbf{y}_k, M', M) \cdot \alpha_{k-1}(M')} \quad (2.82)$$

$$\beta_k(M) = \frac{\sum_{M''} p(S_{k+1} = M'', \underline{\mathbf{y}}_{k+1}^N | S_k = M)}{p(\underline{\mathbf{y}}_{k+1}^N | \underline{\mathbf{y}}_1^k)} \quad (2.83)$$

$$= \frac{\sum_{M''} p(S_{k+1} = M'', \mathbf{y}_{k+1} | S_k = M) \cdot p(\underline{\mathbf{y}}_{k+2}^N | S_{k+1} = M'')}{p(\underline{\mathbf{y}}_{k+1}^N | \underline{\mathbf{y}}_1^k)}$$

burada $p(\underline{\mathbf{y}}_{k+2}^N | S_{k+1} = M'') = p(\underline{\mathbf{y}}_{k+2}^N | S_{k+1} = M'', \mathbf{y}_{k+1}, S_k = M)$ dir. Bu durumda

$\beta_k(M)$ için yeni oluşacak denklem aşağıdaki gibi gösterilir.

$$\beta_k(M) = \frac{\sum_{M''} p(S_{k+1} = M'', \mathbf{y}_{k+1} | S_k = M) \cdot \frac{p(\underline{\mathbf{y}}_{k+2}^N | S_{k+1} = M'')}{p(\underline{\mathbf{y}}_{k+2}^N | \underline{\mathbf{y}}_1^{k+1})}}{\sum_{M''} \sum_M p(S_{k+1} = M'', S_k = M, \mathbf{y}_{k+1} | \underline{\mathbf{y}}_1^k)} \quad (2.84)$$

$$= \frac{\sum_{M''} \gamma_T(\mathbf{y}_{k+1}, M, M'') \cdot \beta_{k+1}(M'')}{\sum_{M''} \sum_M \gamma_T(\mathbf{y}_{k+1}, M, M'') \cdot \alpha_k(M)}$$

İleri olasılıklar ($\alpha_k(M)$), kodlayıcının başlangıç durumunun 0 olduğu bilindiğinden dolayı $\alpha_0(0)=1$ alınarak ve dal geçiş olasılıkları γ 'nın da önceden hesaplanmış olduğundan dolayı (2.82) formülüne göre hesaplanır. Aynı şekilde $\beta_k(M)$, anlamsız bitlerden dolayı (*dummy bits*) kodlayıcının bir çerçeve gönderildikten sonraki son durumunun 0 olduğu bilindiğinden $\beta_N(0)=1$ alınarak ve dal geçiş olasılıkları γ 'nın da önceden hesaplanmış olduğundan dolayı (2.84) formülüne göre hesaplanabilir. Bu işlemlerden sonra MAP algoritmasının çıkışı (2.77) formülüne göre elde edilir.

3. BULGULAR

3.1. Sönümlmeli Kanalların ve Faz Bozulmasının Turbo Kafes Kodlara Etkisi

Kablosuz haberleşmede kanala verilen analog sinyaller, kanalda çeşitli nedenlerden dolayı bozulurlar. Bu bozulma genel olarak iki şekilde olur; bunlardan birincisi, sinyalin genliğindeki bozulma diğeri de fazındaki bozulmadır. Bu konular ile ilgili olarak, Turbo kodların 1+D gibi bir önceki sinyali o anki sinyalin üzerine bindiren bellekli kanallardaki sönümlmeli ve faz bozulması etkisi altındaki başarımı [31-33]'de, Turbo kafes kodların sönümlmeli ve faz bozulmalı kanallardaki performansları ise [34]'de incelenmiştir.

3.1.1. Sönümleme Etkisi

Bu etki genellikle sinyalin genliğini azaltan şekildedir. Bundan dolayı gönderilen sinyallerin Öklit uzaklıkları birbirlerine yaklaşır diğeri bir değişle sinyalin gücü azalır. Bu sönümleme iki nedenden dolayı oluşur. Birincisi sinyalin herhangi bir malzemenin içinden geçerken gücünün azalması şeklinde olduğudur. Diğeri ise, kapalı bir yapı içindeki (*wireless local loop*) gönderilen sinyalin oda içerisinde birçok yerden yansyarak farklı zamanlarda alıcıya gitmesi ve bu yansımaların daha sonraki sinyallerin üzerine eklenmesi şeklinde olur. Buna da çoklu-yol (*multipath*) etkisi denir. Bu kanal şekli kanalın sönümleme etkisinin çok ya da az olmasına bağlı olarak (3.1)'de verilmiş

olan Rician olasılık yoğunluk fonksiyonu ile modellenir. Burada K Rician fonksiyonunun parametresidir ve dB ile ifade edilir. K sonsuz olduğunda sönümleme etkisi tamamen yok olurken, K azalırken sönümleme etkisi de artar. Alınan sinyaller sadece yansılardan ibaret olduğu zamanki durum en çok sönümlemenin olduğu durumdur ve bu durumda kanala Rayleigh kanal denir. Rayleigh kanal modeli (3.2)'de verildiği gibi Rician kanal parametresi K 'nın 0 dB olarak alınmasıyla elde edilebilir.

$$P(\rho) = 2\rho(1+K) e^{(-\rho^2(1+K)-K)} I_0[2\rho\sqrt{K(1+K)}] \quad (3.1)$$

$$P(\rho) = 2\rho e^{(-\rho^2)} \quad (3.2)$$

3.1.2. Faz Bozulması Etkisi

Diğer etki ise fazındaki bozulmadır. Gönderilen sinyal herhangi bir malzemenin içinden geçerken veya yansırken hızında bir düşme olur. Bilindiği gibi tüm sinyaller ışık hızında hareket ederler, fakat yukarıda belirtilen dış etkilere dolayı bir yavaşlama söz konusu olabilir. Bu da o sinyalin gecikmesine neden olur, buna faz bozulması ya da seyirme (jitter) etkisi denir. Faz bozulması sinyalin gücünde herhangi bir azalma yaratmaz, fakat fazını değiştirdiği için başka bir sinyale yaklaşmasına ya da benzemesine neden olur, bu da alıcıda başka bir girişe ait sinyal olarak alınmasına neden olabilir. Bu faz bozulması da (3.3)'de verildiği gibi Tikhonov olasılık yoğunluk fonksiyonu ile modellenir.

$$p(\theta) = \frac{e^{\alpha \cos(\theta)}}{2\pi I_0(\alpha)} \quad \left| \theta \right| \leq \pi \quad (3.3)$$

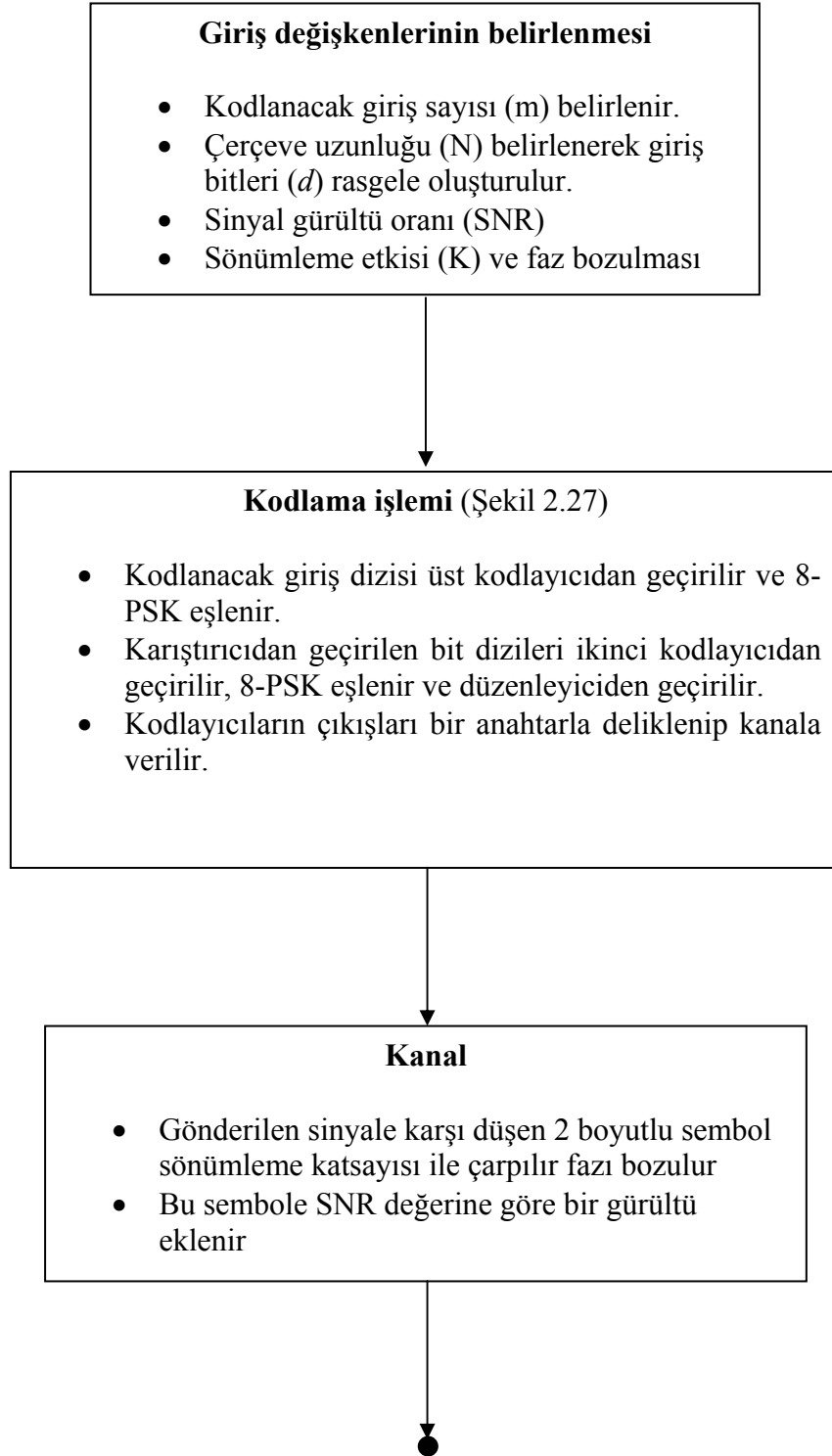
Faz bozulmasının bunun dışında başka bir nedeni de vardır. Bu, özellikle mobil haberleşme ile özdeşleştirilmiş olan Doppler kaymasıdır. Yani haberleşme sırasında alıcı ve verici arasındaki mesafede bir değişiklik oluyorsa yani bir hareket var ise Doppler kayması oluşur bu da fazda bir kayma olmasına neden olur.

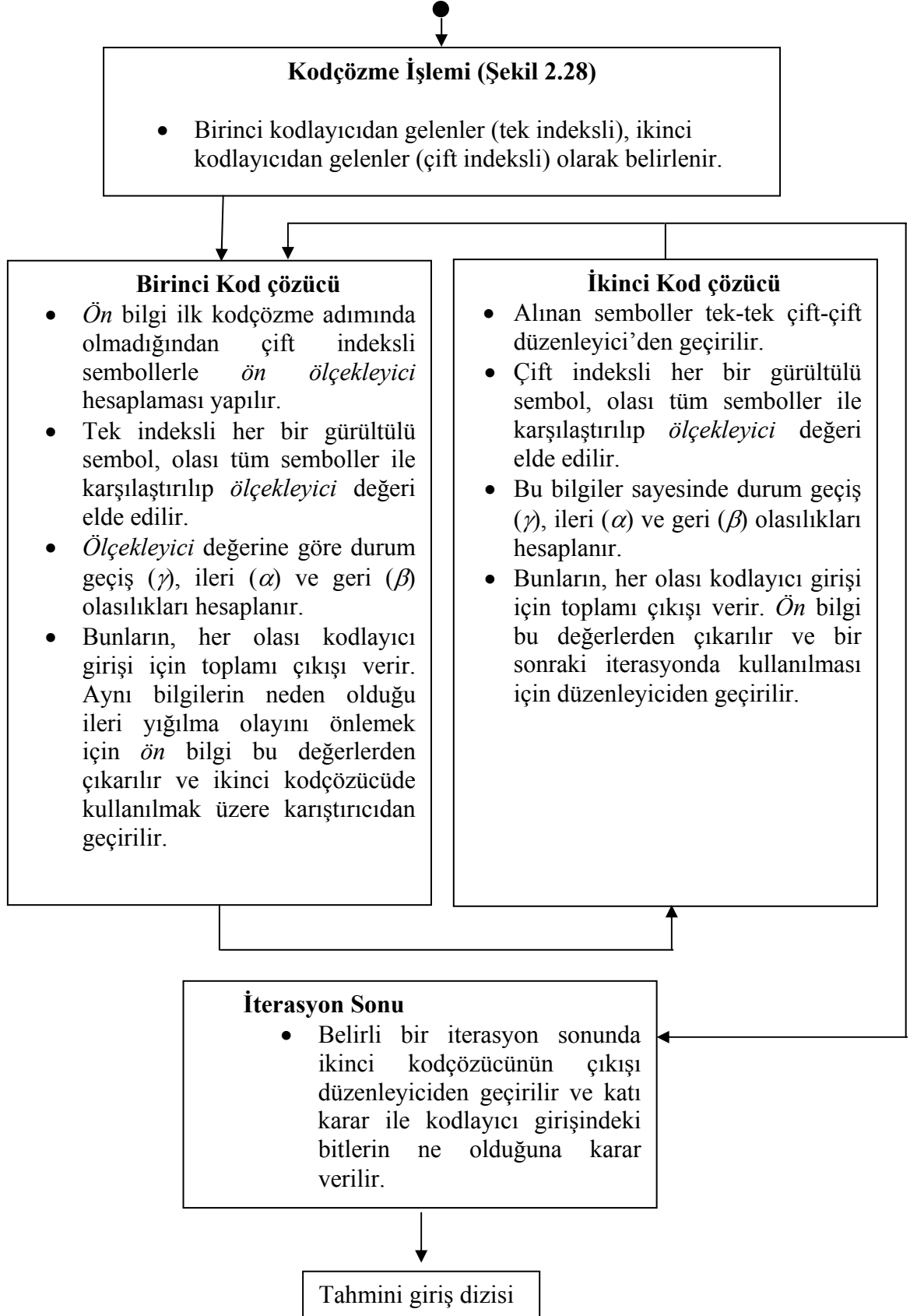
Turbo Kafes Kodlamalı Modülasyon yönteminin her ortamdaki bit hata performansını görebilmek için sönmülemeli kanalları ve faz bozulma etkisini göz önünde bulundurmak gerekir. Vericiden k anında gönderilen işaret x_k , sönmüleme etkisi ρ_k , faz bozulması θ_k ve Gauss gürültüsü n_k olmak üzere alınan demodüle edilmiş işaret aşağıdaki şekildedir.

$$y_k = \rho_k x_k e^{j\theta_k} + n_k \quad (3.4)$$

Simülasyonda sistemin verici, kanal ve alıcı blok diyagramlarının nasıl gerçekleştirildiğini ve hangi değişkenlerin nasıl ve nerede belirlendiğini göstermek amacı ile aşağıdaki gibi bir akış diyagramı verilmiştir.

Tablo 3.1 Sönümlemeli kanallar için Turbo kodlama akış şeması





3.1.3. Sönümlemeli Kanallarda Faz Bozulması Etkisi Altında Turbo Kafes Kodlamalı Modülasyon'un Başarımı

Simülasyon, farklı kanallar ve farklı faz bozulma durumları için çeşitli sinyal gürültü oranlarında, çerçeve boyutunun 256 ve 1024 olması halinde iki girişli 8 durumlu TCM sistemi için yapılmıştır. Modülasyon olarak 8-PSK uygulanmış, kod çözme işlemi sırasında kanal parametrelerinin tam olarak bilindiği ve hiç bilinmediği durumlar göz önüne alınmıştır. Ayrıca kod çözme işleminde 4 iterasyon yapılmıştır.

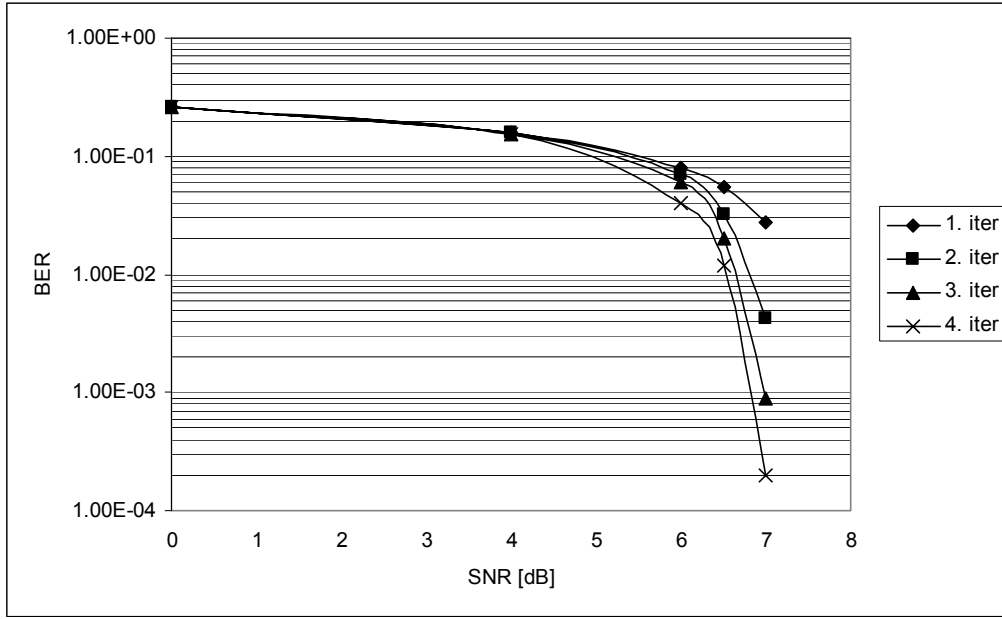
Simülasyon MATLAB programı kullanılarak hazırlanmıştır. PIV 1.5 GHz bir bilgisayarda 4 iterasyonda, başlangıçta 256 bit uzunluklu bir çerçeve için 4.8 sn ve 1024 bit uzunluklu bir çerçeve için 14 sn sürmektedir. Çerçeveler gönderilirken sistemin bellek özelliklerine ve işletim sistemine göre bu süre uzayabilmektedir. Belirli bir bit hata olasılığını istatistiksel olarak doğru yakalamak için gönderilmesi gereken bit sayısı hesabı: Örneğin belirli bir SNR değerinde 10^{-x} bit hata olasılığı yakalanıyor ise gönderilmiş bit sayısının minimum değerinin 10^{x+1} olması gerekmektedir. Bu da yaklaşık olarak o durumda en az 10 bitte hata olması gerektiğini söyler. Bu nedenle gönderilen bit sayısı arttıkça bit hata olasılığı daha doğru olarak elde edilmiş olur.

Kod çözme işlemi sırasında faz kayması ile ilgili bilgi tam olarak biliniyorsa, (3.4)'deki denklemi $e^{-j\theta_k'}$ ile çarparak kullanırsak faz bozulma etkisini tamamen ortadan kaldırmış oluruz. Bu yüzden simülasyonlar θ_k 'nin bilinmediği düşünülerek yapılmıştır. Sönümleme katsayısını bilmek ise ρ 'nun etkisini tam olarak yok etmemekte sadece azaltmaktadır.

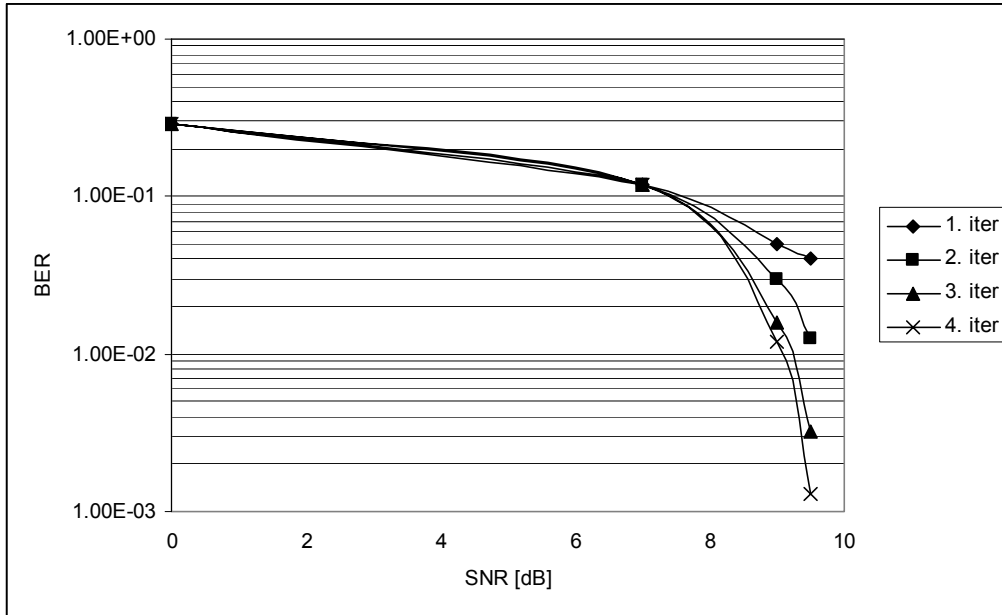
Şekil 3.1’den 3.6’ya kadar olan simülasyon sonuçları AWGN kanal ve 4 iterasyon için alınmıştır. Şekil 3.1, 3.2 ve 3.3’de giriş 1024 bitlik iki dizi halinde olduğu düşünülmüştür. Şekil 3.1’de hiç bir bozucu etki yoktur, Şekil 3.2’de $\alpha=20\text{dB}$ olacak şekilde faz kayması vardır, Şekil 3.3’de ise $\alpha=10\text{dB}$ bir faz kayması vardır. Simülasyon sırasında (3.4) denklemindeki faz kaymasına neden olan θ değerlerinin bilinmediği farz edilmiştir yani kanal bilgisi olmadan çözülmüştür.

Şekil 3.4, 3.5 ve 3.6’da giriş 256 bitlik iki dizi halinde olduğu düşünülmüştür. Şekil 3.4’de hiç bir bozucu etki yoktur, Şekil 3.5’de $\alpha=20\text{dB}$ olacak şekilde faz kayması vardır, Şekil 3.6’de ise $\alpha=10\text{dB}$ ’lık bir faz kayması vardır.

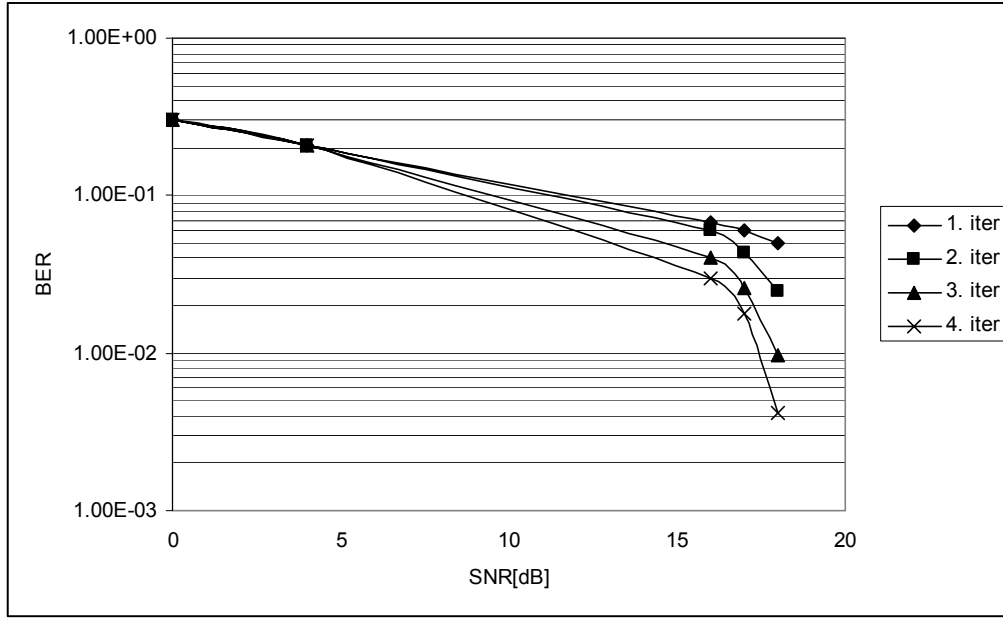
Simülasyon sırasında faz kaymasına neden olan θ değerlerinin bilinmediği farz edilmiştir yani kanal bilgisi olmadan çözülmüştür. Simülasyon sonuçlarına bakıldığında faz bozulması arttıkça hata başarımının azaldığı görülmektedir. Aynı zamanda, çerçeve uzunluğu arttıkça gönderilen bitler arasındaki ilişki de artacağından hata başarımı iyileşir, bu nedenle düşük çerçeve uzunluğuna sahip bit dizilerinde başarımlar daha kötüdür.



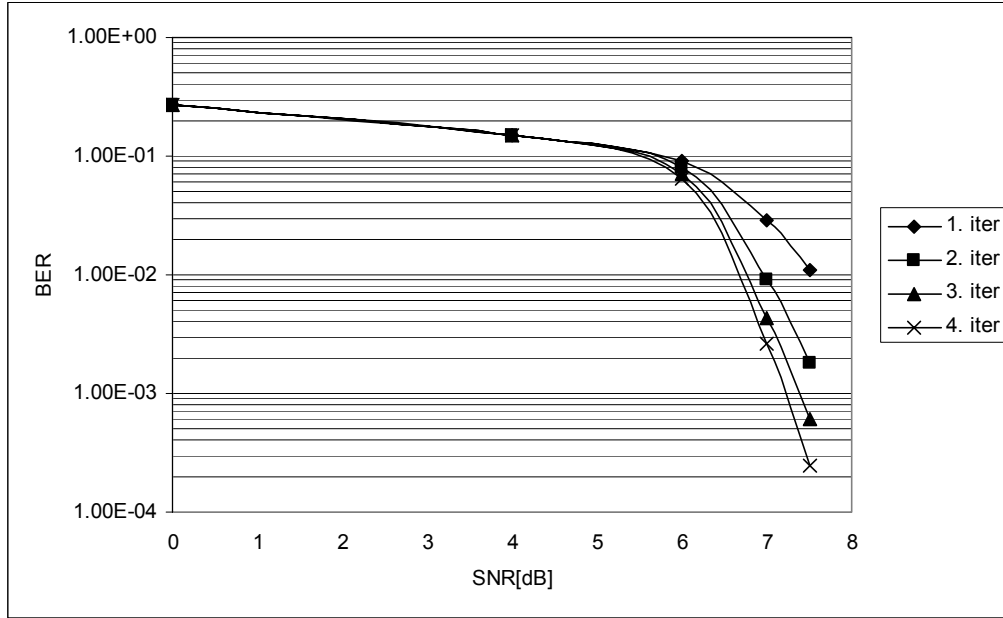
Şekil 3.1 AWGN kanal için TTCM hata başarımı, $N=1024$.



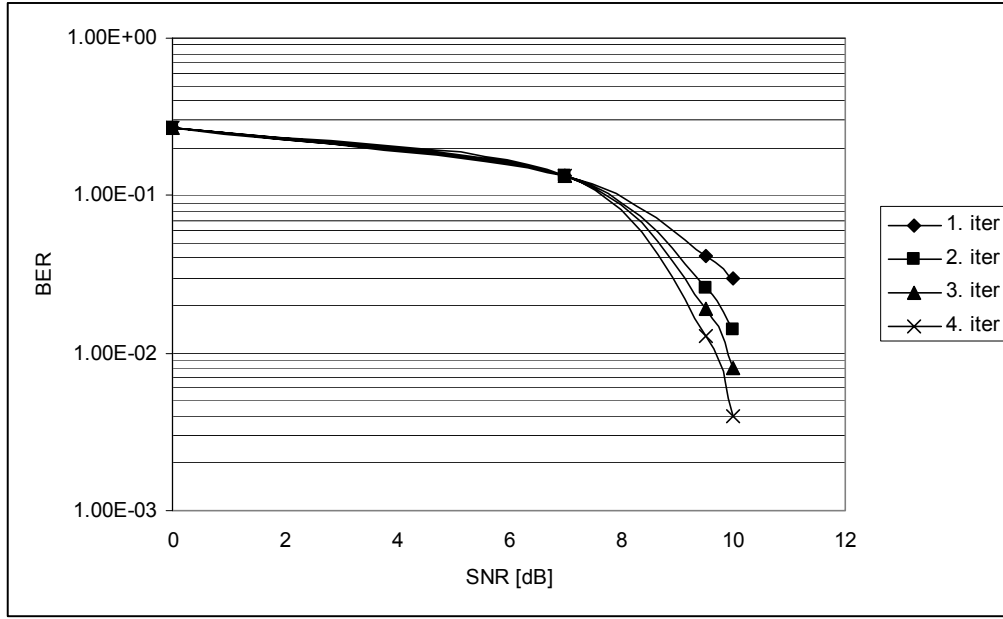
Şekil 3.2 AWGN kanal, $\alpha=20\text{dB}$ 'lik faz bozulması ve kanal bilgisinin olmadığı durumda TTCM hata başarımı, $N=1024$.



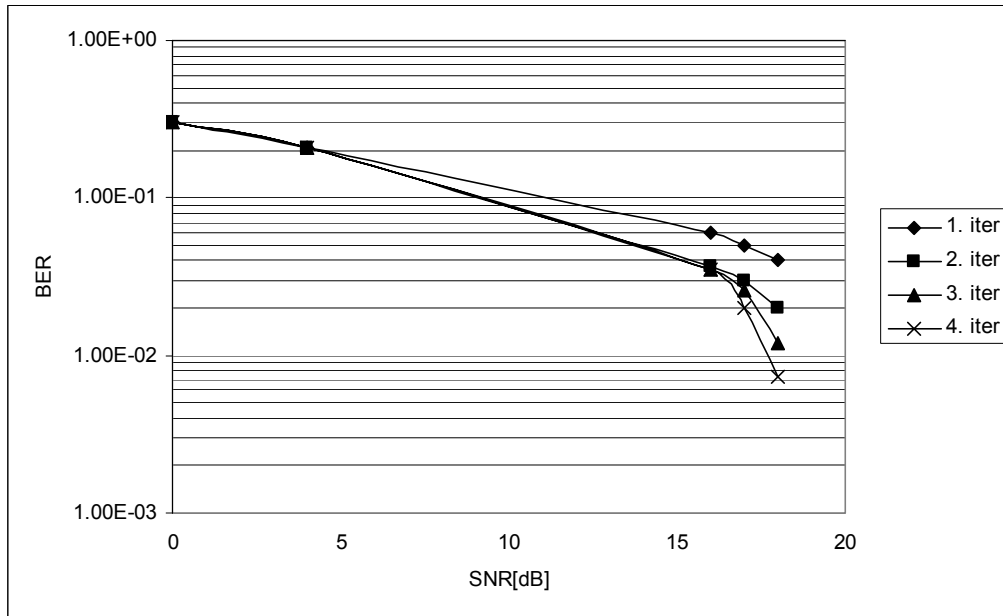
Şekil 3.3 AWGN kanal, $\alpha=10\text{dB}$ 'lik faz bozulması ve kanal bilgisinin olmadığı durumda TTCM hata başarımı, $N=1024$.



Şekil 3.4 AWGN kanal için TTCM hata başarımı, $N=256$.



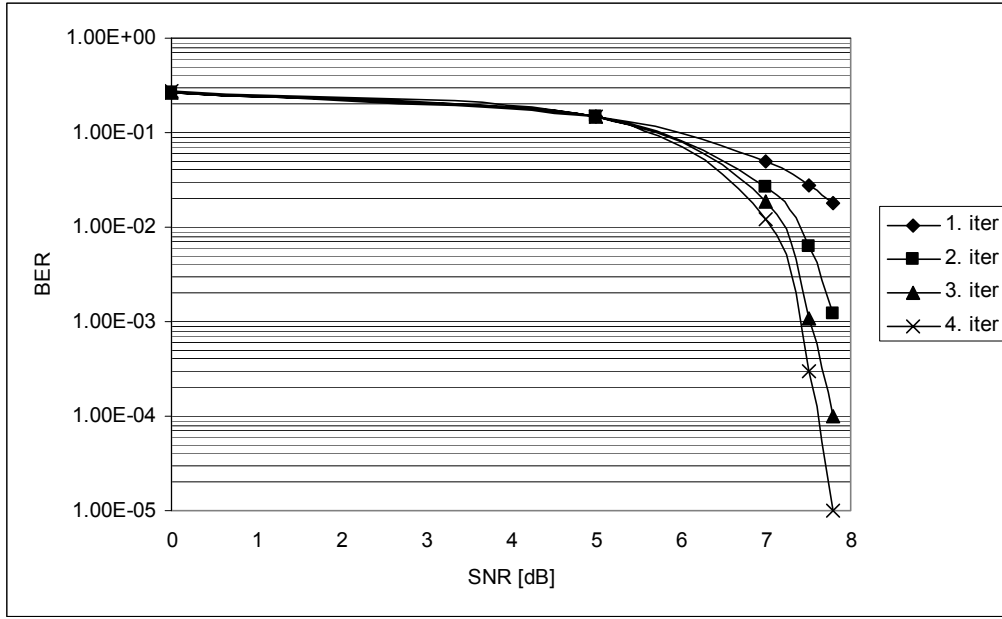
Şekil 3.5 AWGN kanal, $\alpha=20\text{dB}$ 'lik faz bozulması ve kanal bilgisinin olmadığı durumda TTCM hata başarımı, $N=256$.



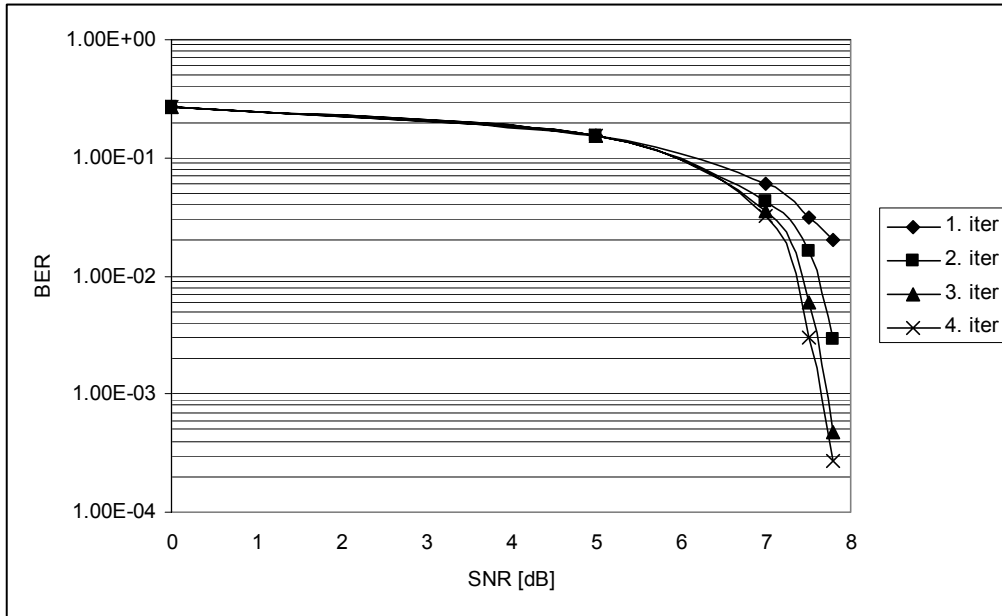
Şekil 3.6 AWGN kanal, $\alpha=10\text{dB}$ 'lik faz bozulması ve kanal bilgisinin olmadığı durumda TTCM hata başarımı, $N=256$.

Şekil 3.7’den 3.14’e kadar olan simülasyonlar, $K=10\text{dB}$ Rician kanal ve 4 iterasyon için yapılmıştır. Şekil 3.7 - 3.10’da giriş 1024 bitlik iki dizi halinde olduğu düşünülmüştür. Şekil 3.7’de kanal bilgisinin tam olarak bilindiği varsayılmıştır. Şekil 3.8’de ise kanal bilgisinin olmadığı varsayılmıştır. Şekil 3.9 ve 3.10’da ise $\alpha=20\text{dB}$ ve 10dB ’lik bir faz bozulması söz konusudur. Bu simülasyon sonuçları da kanal bilgisinin olmadığı varsayılarak yapılmıştır.

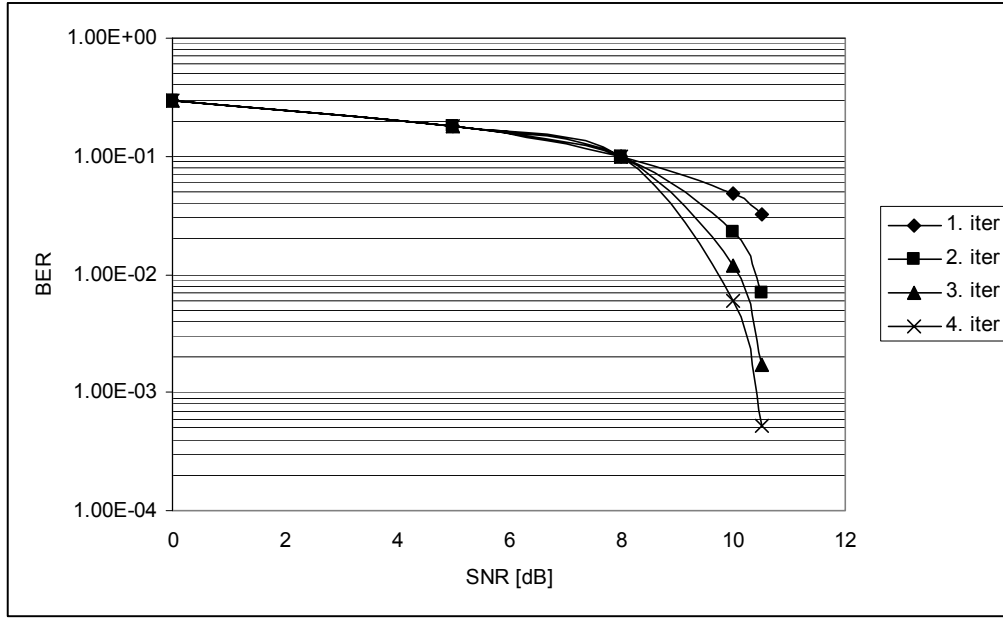
Şekil 3.11’de girişin 256 bitlik olduğu varsayılmış, sinyalin sadece sönümleme etkisine maruz kaldığı düşünülmüş ve kanal bilgisinin tam olarak bilindiği varsayılmıştır. Şekil 3.12’de ise aynı durum için kanal bilgisinin olmadığı düşünülmüştür. Şekil 3.13 ve 3.14’de ise $\alpha=20\text{dB}$ ve 10dB ’lik bir faz bozulması söz konusudur. Simülasyon sırasında faz kaymasına neden olan θ değerlerinin bilinmediği farz edilmiş, yani kanal bilgisi olmadan çözülmüştür. Simülasyon sonuçlarına bakıldığında faz bozulması arttıkça hata başarımının azaldığı, çerçeve uzunluğu arttıkça da hata başarımının arttığı görülmektedir. Ayrıca sönümleme etkisinin hata başarımını oldukça etkilediği, kanal bilgisinin elde edilmiş olması durumunda ise hata başarımının oldukça iyileştiği görülmektedir.



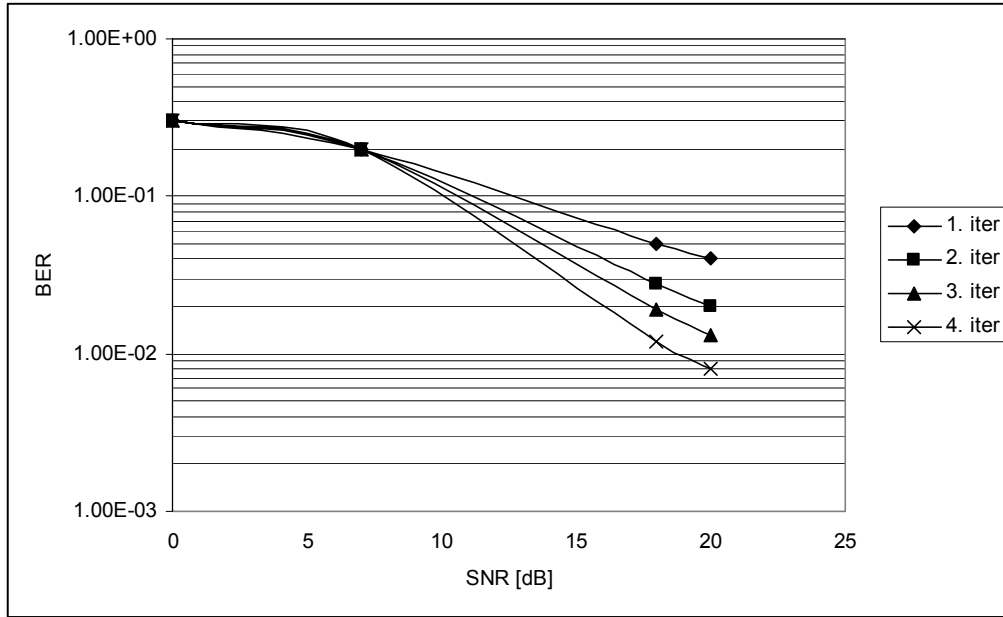
Şekil 3.7 $K=10$ dB'lik Rician kanal için kanal bilgisinin tam olarak bilinmesi durumunda TTCM hata başarımı, $N=1024$.



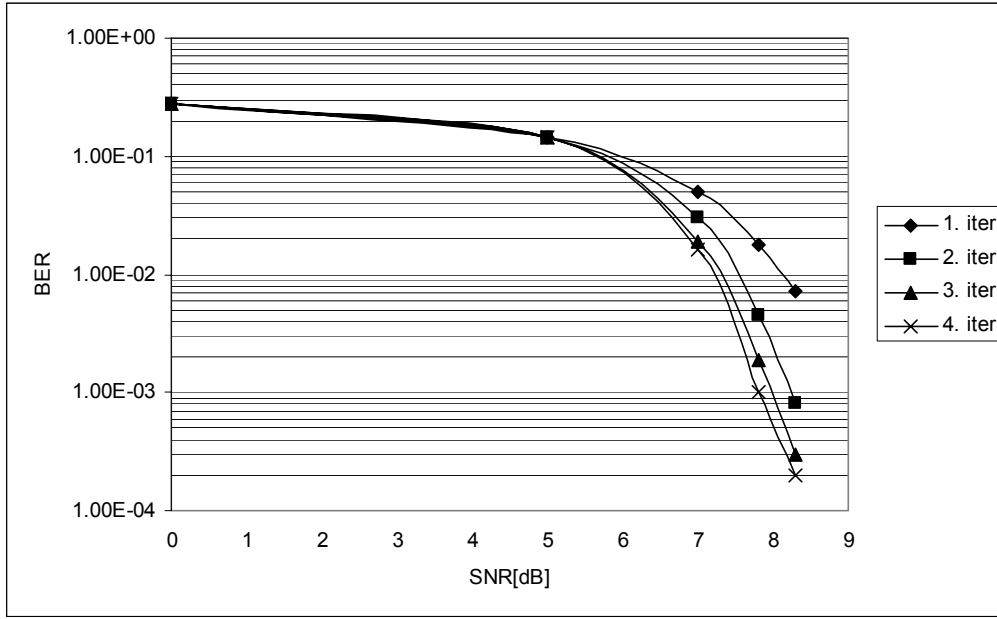
Şekil 3.8 $K=10$ dB'lik Rician kanal için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$



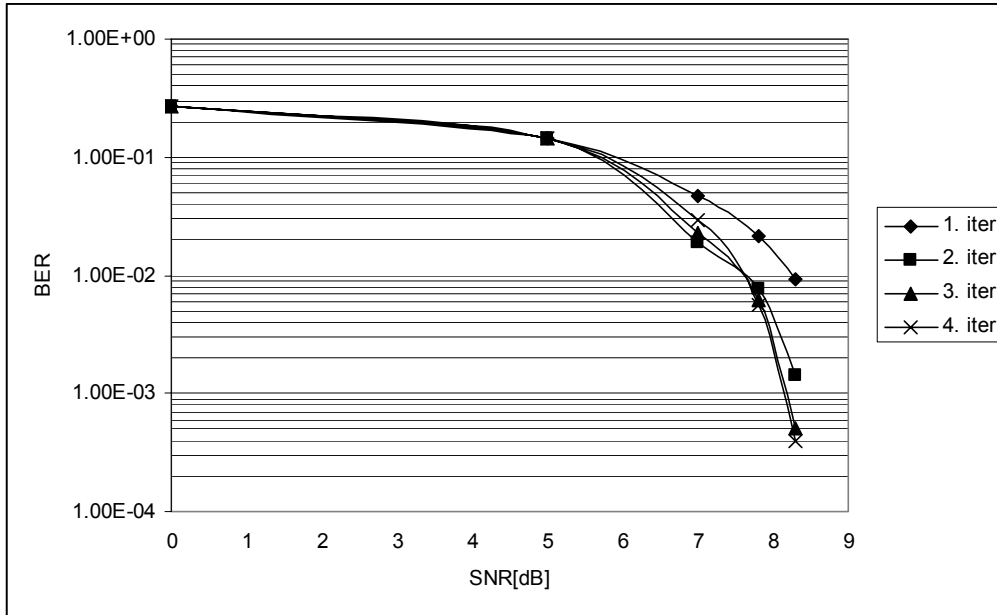
Şekil 3.9 $K=10\text{dB}$ 'lık Rician kanal ve $\alpha=20\text{dB}$ 'lık faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$



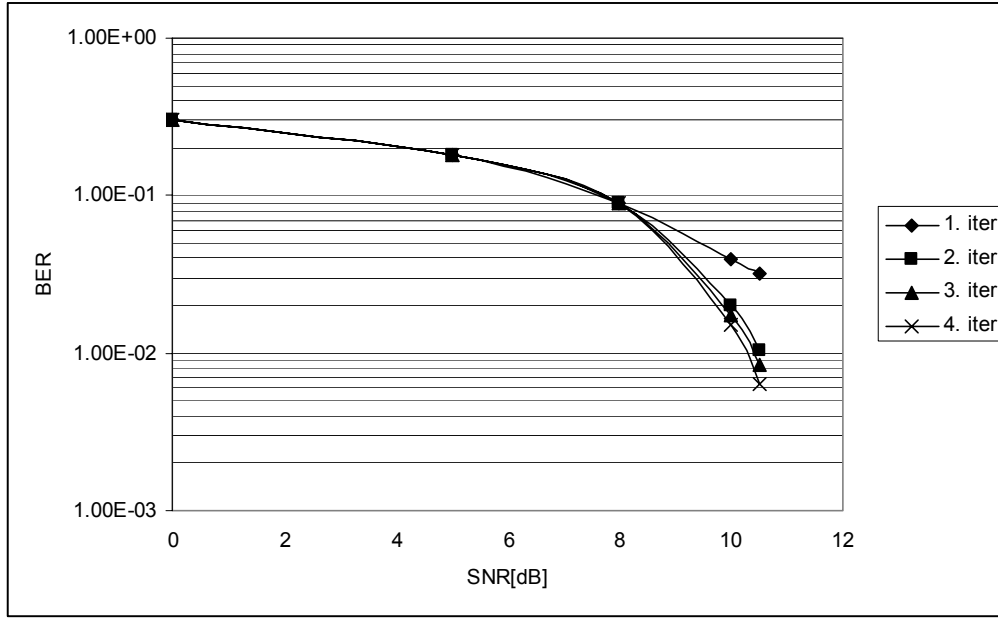
Şekil 3.10 $K=10\text{dB}$ 'lık Rician kanal ve $\alpha=10\text{dB}$ 'lık faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$



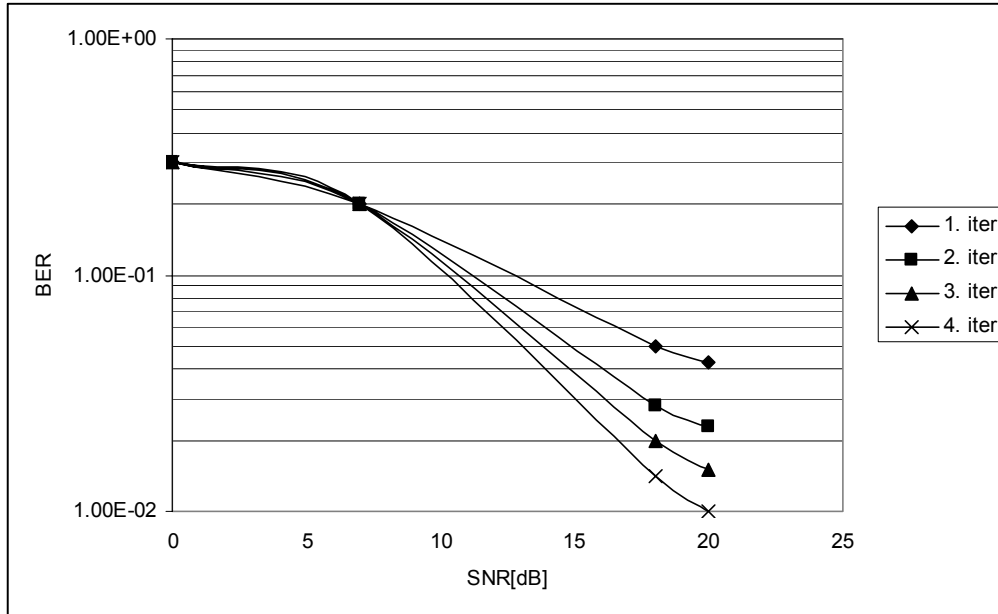
Şekil 3.11 $K=10$ dB’lik Rician kanal için kanal bilgisinin tam olarak bilinmesi durumunda TCM hata başarımı, $N=256$.



Şekil 3.12 $K=10$ dB’lik Rician kanal için kanal bilgisinin bilinmemesi durumunda TCM hata başarımı, $N=256$



Şekil 3.13 $K=10\text{dB}$ 'lik Rician kanal ve $\alpha=20\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=256$

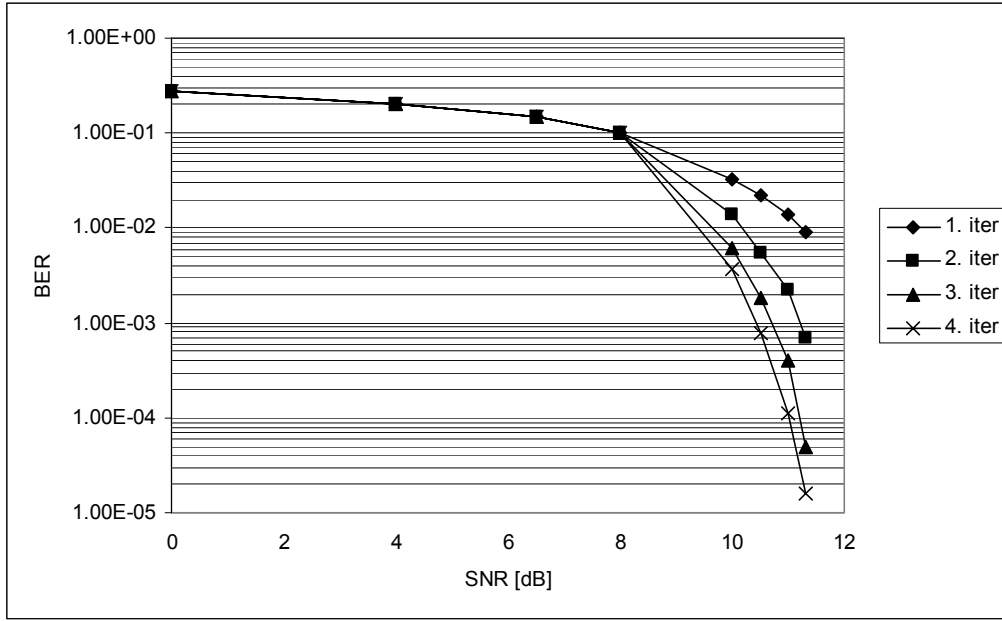


Şekil 3.14 $K=10\text{dB}$ 'lik Rician kanal ve $\alpha=10\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=256$

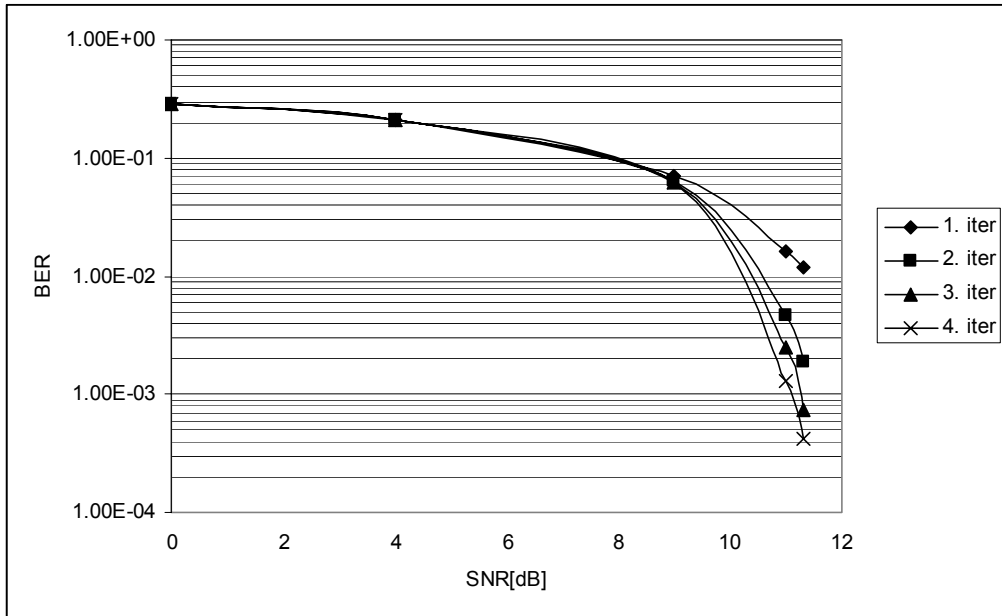
Şekil 3.15'den 3.22'ye kadar olan simülasyonlar $K=0$ dB Rician kanalda, başka bir deyişle Rayleigh kanalda, 4 iterasyon için yapılmıştır. Şekil 3.15 - 3.18'de giriş 1024 bitlik iki dizi halinde olduğu düşünülmüştür. Şekil 3.15'de kanal bilgisinin tam olarak bilindiği varsayılmıştır. Şekil 3.16'de ise kanal bilgisinin olmadığı varsayılmıştır. Şekil 3.17 ve 3.18'de ise $\alpha=20$ dB ve 10dB'lik bir faz bozulması söz konusudur. Bu simülasyon sonuçları da kanal bilgisinin olmadığı varsayılarak yapılmıştır.

Şekil 3.19'da girişin 256 bitlik olduğu varsayılmış, sinyalin sadece sönmüleme etkisine maruz kaldığı düşünülmüş ve kanal bilgisinin tam olarak bilindiği varsayılmıştır. Şekil 3.20'de ise aynı durum için kanal bilgisinin olmadığı düşünülmüştür. Şekil 3.21 ve 3.22'de ise $\alpha=20$ dB ve 10dB'lik bir faz bozulması söz konusudur. Simülasyon sırasında faz kaymasına neden olan θ değerlerinin bilinmediği farz edilmiş, yani kanal bilgisi olmadan çözülmüştür.

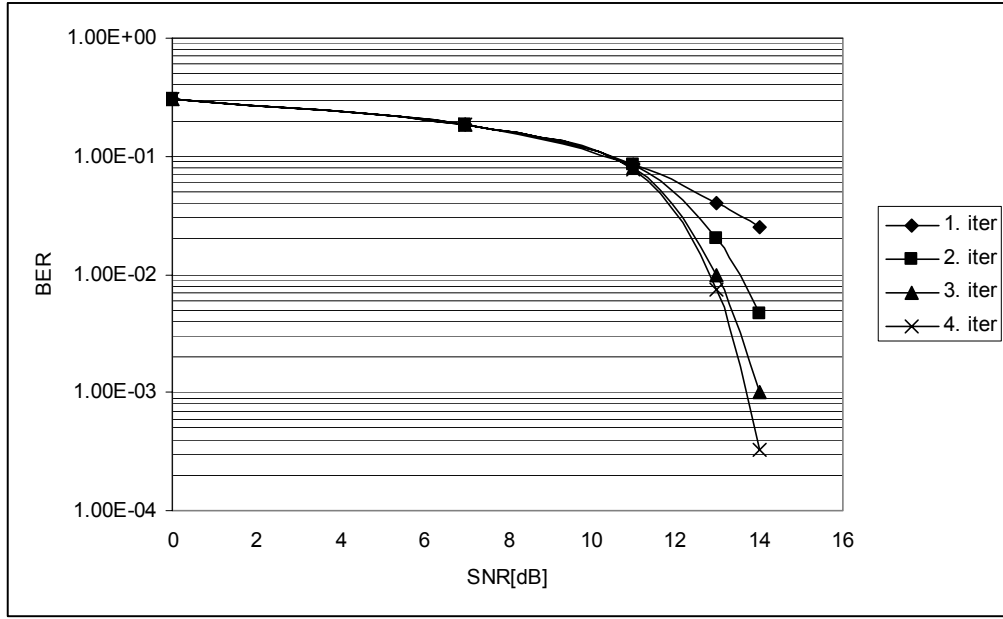
Simülasyon sonuçlarına bakıldığında faz bozulması arttıkça hata başarımının azaldığı, çerçeve uzunluğu arttıkça da hata başarımının arttığı görülmektedir. Ayrıca sönmüleme etkisinin hata başarımını oldukça etkilediği, kanal bilgisinin elde edilmiş olması durumunda ise hata başarımının oldukça iyileştiği görülmektedir.



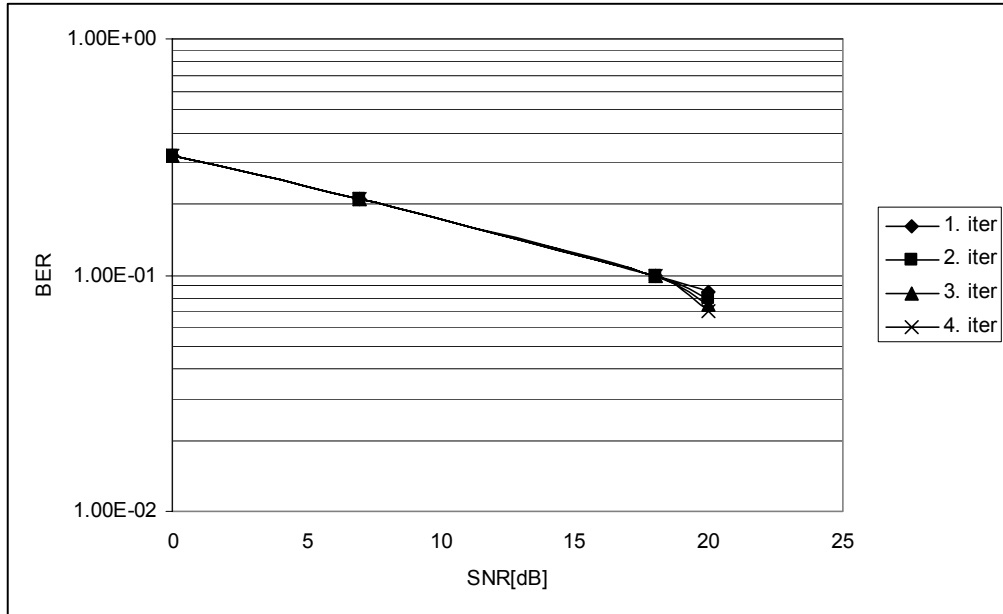
Şekil 3.15 Rayleigh kanal için kanal bilgisinin tam olarak bilinmesi durumunda TTCM hata başarımı, $N=1024$.



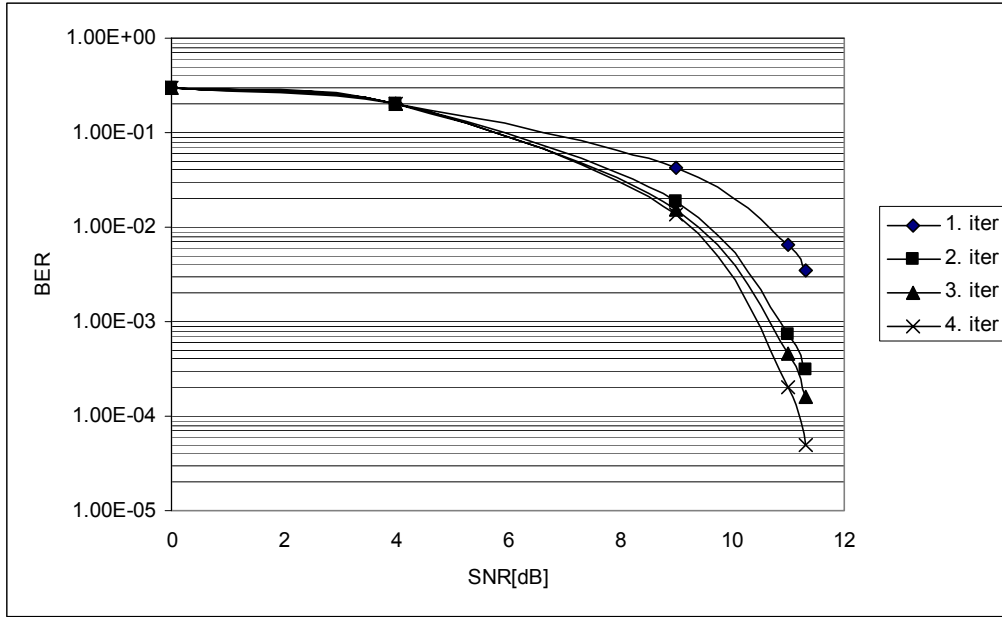
Şekil 3.16 Rayleigh kanal için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$



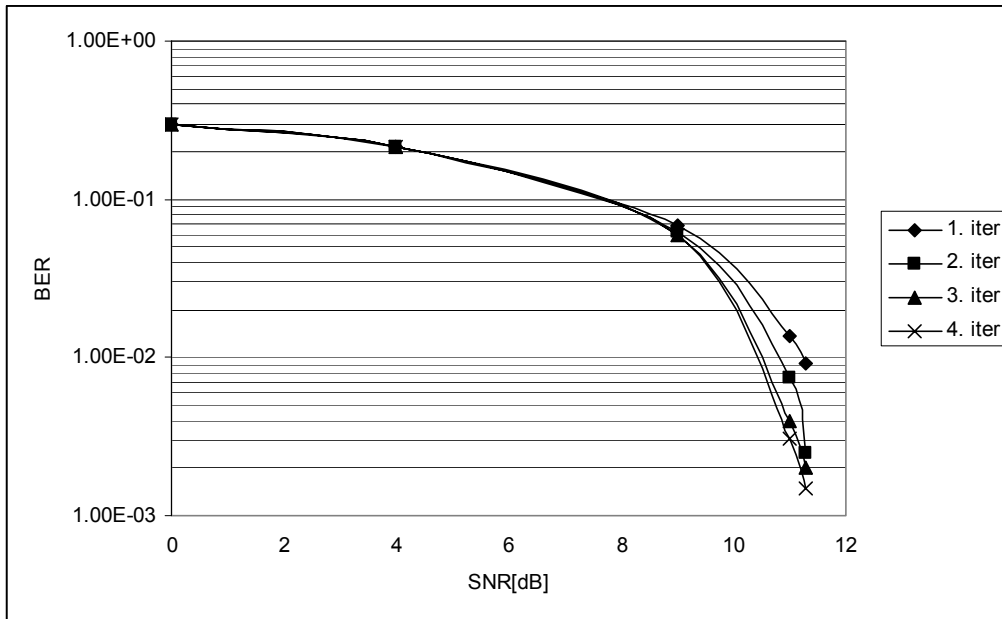
Şekil 3.17 Rayleigh kanal ve $\alpha=20\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$



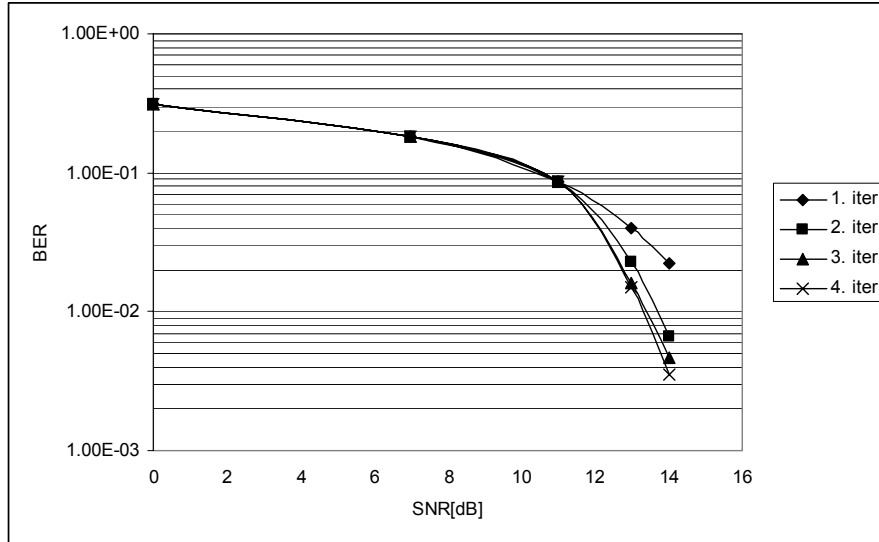
Şekil 3.18 Rayleigh kanal ve $\alpha=10\text{dB}$ 'lik faz bozulması için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=1024$



Şekil 3.19 Rayleigh kanal için kanal bilgisinin tam olarak bilinmesi durumunda TTCM hata başarımı, $N=256$.

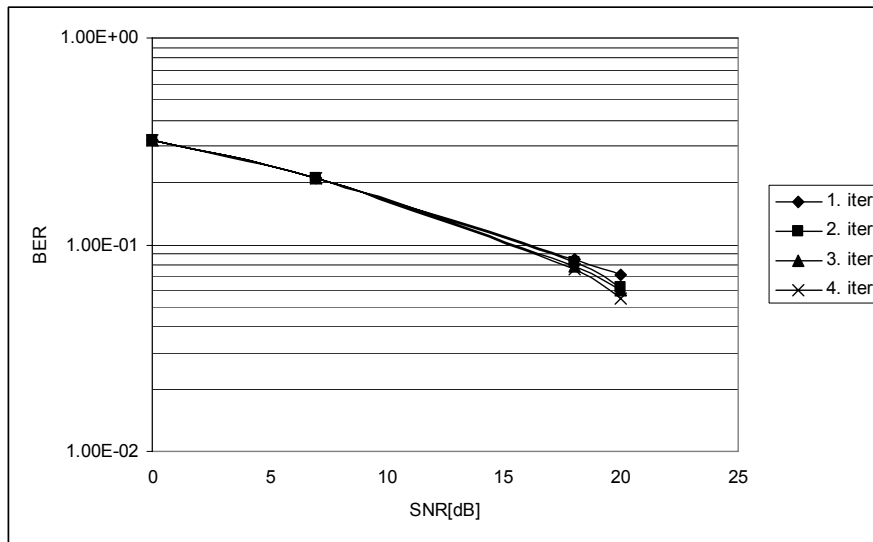


Şekil 3.20 Rayleigh kanal için kanal bilgisinin bilinmemesi durumunda TTCM hata başarımı, $N=256$



Şekil 3.21 Rayleigh kanal ve $\alpha=20\text{dB}$ 'lik faz bozulması için kanal bilgisinin

bilinmemesi durumunda TTCM hata başarımı, $N=256$



Şekil 3.22 Rayleigh kanal ve $\alpha=10\text{dB}$ 'lik faz bozulması için kanal bilgisinin

bilinmemesi durumunda TTCM hata başarımı, $N=256$

Tüm simülasyon sonuçlarına genel olarak bakıldığında Rayleigh kanalında elde edilen sonuçların hepsi $K=10\text{dB}$ 'lik Rician kanal ve AWGN kanalından daha kötü olduğu görülmektedir. Sönümleme etkisinin bilinmesi durumunun ve çerçeve uzunluğunun hata başarımını oldukça etkilediği, kanal bilgisinin tam olarak elde edilmiş olması ve uzun çerçeveli bit dizileri için hata başarımının oldukça iyileştiği anlaşılmaktadır.

3.2. Gözü Kapalı Kanal Parametre Kestirimi

Dijital iletişimde, sinyallerin yansımalarından doğan sönümlenme ve Doppler kayması büyük önem taşır. Mobil uydu haberleşmesinde kanallar çoklu-yol ortamının Rician dağılımlı, faz bozulmasının da Tikhonov dağılımlı olduğu düşünülerek modellenir.

Burada TTCM tarafından oluşturulan sinyallerdeki genlik ve faz bozulmalarını kestirmek için modifiye edilmiş gözü kapalı maksimum olasılık (maximum likelihood, ML) metodu açıklanacak ve simülasyon sonuçları verilecektir. Bu yöntemin kullanılmasının sebebi, mevcut data sayısının az olması durumunda bile oldukça iyi sonuç vermesidir [35]. Fakat gözü kapalı problemlerde ML yöntemini uygulamak işlem sayısını arttırdığı için pek pratik olmamaktadır. Bu yüzden, olasılığı enbüyüklemek için iteratif yaklaşımı uygulamak daha iyi olacaktır. TTCM'in sonlu sayıda durumu olması, girişin bağımsız aynı dağılımlı ve alfabetinin sonlu sayıda olması sistem cevabını Markov zinciri olarak modellememize imkan tanır [35-39]. Alınan sinyalin gürültülü olması Markov zincirinin saklı olmasına neden olur. Bundan dolayı problemin Saklı Markov Model formülleri ve ML'i maksimum yapan işlem sayısı bakımından az olan Baum-Welch algoritması uygulanır.

Baum-Welch algoritması, ML ile Markov zincirine bağlı parametrelerin kestirimi için geliştirilmiş ve ses işlemedeki Saklı Markov Modellerde parametrelerin

uygunlaştırılmasında oldukça geniş bir alan bulmuş bir yöntemdir. Haberleşmede de özellikle kanal parametrelerinin kestiriminde kullanılır [35-39].

Kanal parametrelerini kestirirken TTCM blok diyagramı ve buradaki geri beslemeli sistematik katlamalı kodlayıcıların kafes yapıları göz önünde bulundurulmalıdır. i . adımda m -bitlik bilgi vektörü $\mathbf{d}(i) = (d_1(i), d_2(i), \dots, d_m(i))^T$ nin TTCM sistemine girer. Bu sistemin çıkışı da $g(\mathbf{d}(i))$ şeklinde gösterilir. Sönümlemeli ve faz bozulmasının olduğu bir ortamdan geçen TTCM sinyalinin alıcıdaki formu

$$y(i) = \rho(i)g(\mathbf{d}(i))e^{j\theta(i)} + n(i) \quad (3.5)$$

şeklindedir. Burada $g(\mathbf{d}(i))$ modülatöre bağlı, eşlenmiş simgedir. Alınan sinyal için aşağıdaki varsayımlar kabul edilmiştir.

- 1- $n(i)$, sıfır ortalamalı bağımsız aynı dağılımlı (i.i.d.), σ^2 varyanslı kompleks AWGN dizisidir.
- 2- $\mathbf{d}(i)$, m bitlik eşit olasılıklı TTCM bilgi giriş dizisidir.
- 3- $\rho(i)$, Rician olasılık yoğunluk fonksiyonlu sönümleme genlik değeri ve $\theta(i)$ ise Doppler kaymasından doğan faz bozulmasıdır.
- 4- Kanalin yavaş değişen kanal olduğu farz edilmektedir.

Buradaki temel problem, belirli bir alınan sinyal kümesinden sönümleme ve faz bozulma faktörlerini kestirmektir. Bu problem için, asimptotik kararlılığından dolayı

ML yaklaşımından yararlanılacaktır. 1. varsayımdan yola çıkarak, $\mathbf{d}$ 'nin ML ölçekleyicisi, uzunluğu N olan alınan data dizisinin koşullu olasılık yoğunluk fonksiyonu ile orantılıdır. Bu da aşağıdaki gibi verilir.

$$m_l(y, \mathbf{d}) = \sum_{i=1}^N \left| y(i) - \rho_l g(\mathbf{d}(i)) e^{j\theta_l} \right|^2 \quad (3.6)$$

Burada $\mathbf{d} = (d(1), d(2), \dots, d(N))$ TTCM sistem girişidir. Maksimize edilecek ML ölçekleyicisi tüm olası iletilen $\mathbf{d}$ dizisi üzerinden $m_l(y, \mathbf{d})$ nin ortalaması alındıktan sonra elde edilebilir. Bu toplam da $m_l(y) = \sum_{\mathbf{d}} m_l(y, \mathbf{d})$ şeklinde gösterilir. Daha sonra ortalama ML ölçekleyicisi $\hat{\lambda} = \arg \max_{\lambda} [m_l(y)]$ 'yi maksimum yapan bir $\lambda = [\rho_l, \theta_l, \sigma_l^2]$ kestirimi bulunur. Bu süreçte ρ ve θ 'nin 4. varsayıma bağlı olarak l kodlama adımı boyunca sabit kaldığı düşünülür. $\mathbf{d}$ 'nin ölçekleyicisi, denklem (3.6)'nin $\mathbf{d}(i)$ 'nin olası tüm durumları için ortalaması alınarak bulunabilir.

$$m_l(y) = \sum_d \sum_{i=1}^N \left| y(i) - \rho_l g(\mathbf{d}(i)) e^{j\theta_l} \right|^2 \quad (3.7)$$

Denklem (3.7)'nin direkt minimizasyonu, (3.6)'nın olası tüm iletilen diziler üzerinden değerlendirilmesini gerektirir, bu da işlem karmaşıklığını data uzunluğu N ile üstel olarak artacak şekilde arttırır. Neyseki, bu ayrıntılı araştırmanın ağır işlem yükü, ML kestirimi için genel bir iteratif yaklaşım olan Baum-Welch (BW) algoritması [35-39] ile oldukça azaltılabilmektedir.

3.2.1. Baum-Welch Algoritması

Baum-Welch algoritması, bir Markov zincirinin olasılık fonksiyonun parametre kestirimi için geliştirilmiş iteratif bir ML kestirim algoritmasıdır [36]. Bu yöntem likelihood fonksiyonu yerine Kullback-Leibler enformasyon ölçüsü ile ilgili bir yardımcı fonksiyonun maksimizasyonu temeline dayanır. Bu yöntem özellikle çok parametrenin ve lineer olmayan likelihood fonksiyonlarının olduğu parametre kestirim uygulamalarında kullanılmaktadır. Kullback-Leibler enformasyon ölçüsü aşağıdaki gibi tanımlanır.

$$KL(\lambda_1, \lambda_2) = E_{\lambda_1} \log \left\{ \frac{f_{\lambda_1}(y)}{f_{\lambda_2}(y)} \right\} = E_{\lambda_1} \{ \log f_{\lambda_1}(y) \} - E_{\lambda_1} \{ \log f_{\lambda_2}(y) \} \quad (3.8)$$

Burada $f_{\lambda}(y)$, mevcut datanın bilinmeyen parametrelili dağılımıdır. Kullback-Leibler, $f_{\lambda_1}(y)$ ve $f_{\lambda_2}(y)$ arasındaki farklılığın ölçüsünü verir. λ_2 'ye bağlı $KL(\lambda_1, \lambda_2)$ 'nin maksimizasyonu (3.8)'in ikinci teriminin maksimize olmasıyla eşdeğerdir, çünkü $E_{\lambda_1} \{ \log f_{\lambda_1}(y) \}$ terimi sabittir. Baum-Welch algoritmasında kullanılan yardımcı fonksiyon, (3.8) denkleminin ikinci teriminin bir yaklaşımı olarak tanımlanır ve

$$\Theta(\lambda_1, \lambda_2) \equiv \sum_{\phi} f_{\lambda_1}(y, \phi) \log(f_{\lambda_2}(y, \phi)) \quad (3.9)$$

olarak verilir. Burada $\phi = \{\phi(i)\}_{i=1}^N$ durum dizisini gösterir. Aşağıdaki teorem Baum-Welch algoritması için temel oluşturur ve denklem (3.9)'daki $\Theta(\lambda_1, \lambda_2)$ 'nin seçilme nedenini açıklar.

Teorem: $\Theta(\lambda, \lambda') \geq \Theta(\lambda, \lambda)$ olması $f_{\lambda'}(y) \geq f_{\lambda}(y)$ anlamına gelir.

Teoremin kanıtı [36]'da Baum et al tarafından verilmiştir. Baum-Welch algoritması kestirilecek parametrelerin başlangıçta herhangi bir değer $\lambda^{(0)}$ atanmasıyla başlar. (t+1)'inci iterasyonda t . iterasyondaki $\hat{\lambda}^{(t)}$ değeri ilk değer olarak alır ve λ' üzerinden $\Theta(\hat{\lambda}^{(t)}, \lambda')$ 'yi maksimize ederek $\hat{\lambda}^{(t+1)}$ değerini hesaplar.

$$\hat{\lambda}^{(t+1)} = \arg \max_{\lambda'} \Theta(\hat{\lambda}^{(t)}, \lambda'). \quad (3.10)$$

Bu işlem parametreler kararlı bir noktaya oturana kadar tekrarlanır. Yardımcı fonksiyonun i . gözlemdeki açık hali aşağıdaki gibi gösterilebilir.

$$\Theta(\hat{\lambda}^{(t)}, \lambda') = C + \sum_{k=1}^N \sum_{m=1}^P \gamma_m^{(t)}(i) \left\{ \frac{1}{\sigma_l'^2} \left| y(i) - \rho_l' g(\phi_m) e^{j\theta_l'} \right|^2 \right\} + \log(\sigma_l'^2) \quad (3.11)$$

Burada $\gamma_m^{(t)}(i) = f_{\hat{\lambda}^{(t)}}(y, \phi(i) = \phi_k)$, y datası ile durumun i anında ϕ_k olmasının ortak likelihood'u dur. Aşağıda verilmiş olan ileri $\alpha_k(i)$ ve geri $\beta_k(i)$ olasılık tanımlamaları kullanılarak $\gamma_k(i)$ yinelenerek bulunabilir [35-39].

$$\beta_j^{(t)}(i) = \sum_{j=1}^P a_{m,j} \beta_j^{(t)}(i+1) f_j(y(i+1)) \quad i = N-2, \dots, 0 \quad (3.12)$$

$$\alpha_m^{(t)}(i) = \sum_{m=1}^P a_{m,j} \alpha_m^{(t)}(i-1) f_j(y(i)) \quad i = 1, \dots, N-1 \quad (3.13)$$

Burada a_{mj} Markov zincirinin geiş olasılığıdır ve TTCM için v bellek sayısı olmak üzere $1/2^v$ dir. $f_j(y)$ işaretlerin olasılık yoğunluk foksiyonudur ve Gauss dağılımına uyar.

$$f_k(y(i)) = \frac{1}{\sigma_l \sqrt{2\pi}} \exp \left\{ -\frac{1}{2\sigma_l^2} \|y(i) - \eta_k\|^2 \right\} \quad (3.14)$$

$\eta_k = \rho_l g_z(\phi_k) e^{j\theta_l}$ alınan sinyal ve σ_l^2 Gauss gürültüsünün varyansıdır. β ve α 'nın başlangıç durumları

$$\beta_j^{(t)}(N-1) = 1 \quad 1 \leq j \leq P \quad (3.15)$$

$$\alpha_m^{(t)}(0) = p_m(y(0)) \quad 1 \leq m \leq P \quad (3.16)$$

p_m sinyallerin başlangıç olasılıklarıdır ve TTCM için her sinyalin olasılığı aynıdır ve $1/2^v$ dir. i anında ϕ_k durumunda $\gamma_k(i)$, yineleme ile bulunur.

$$\gamma_k^{(t)}(i) = \alpha_k^{(t)}(i) \beta_k^{(t)}(i) \quad (3.17)$$

İlk defa Turbo kafes kodlamaya uygulamak amacıyla, her üç değışken (σ^2 , ρ ve θ) için elde edilmiş iteratif kestirim formülleri (3.11) yardımcı fonksiyonun gradyanını sıfıra eşitleyerek $\nabla_{\rho_i} \Theta = 0$, $\nabla_{\theta_i} \Theta = 0$, $\nabla_{\sigma_i^2} \Theta = 0$ elde edilebilir.

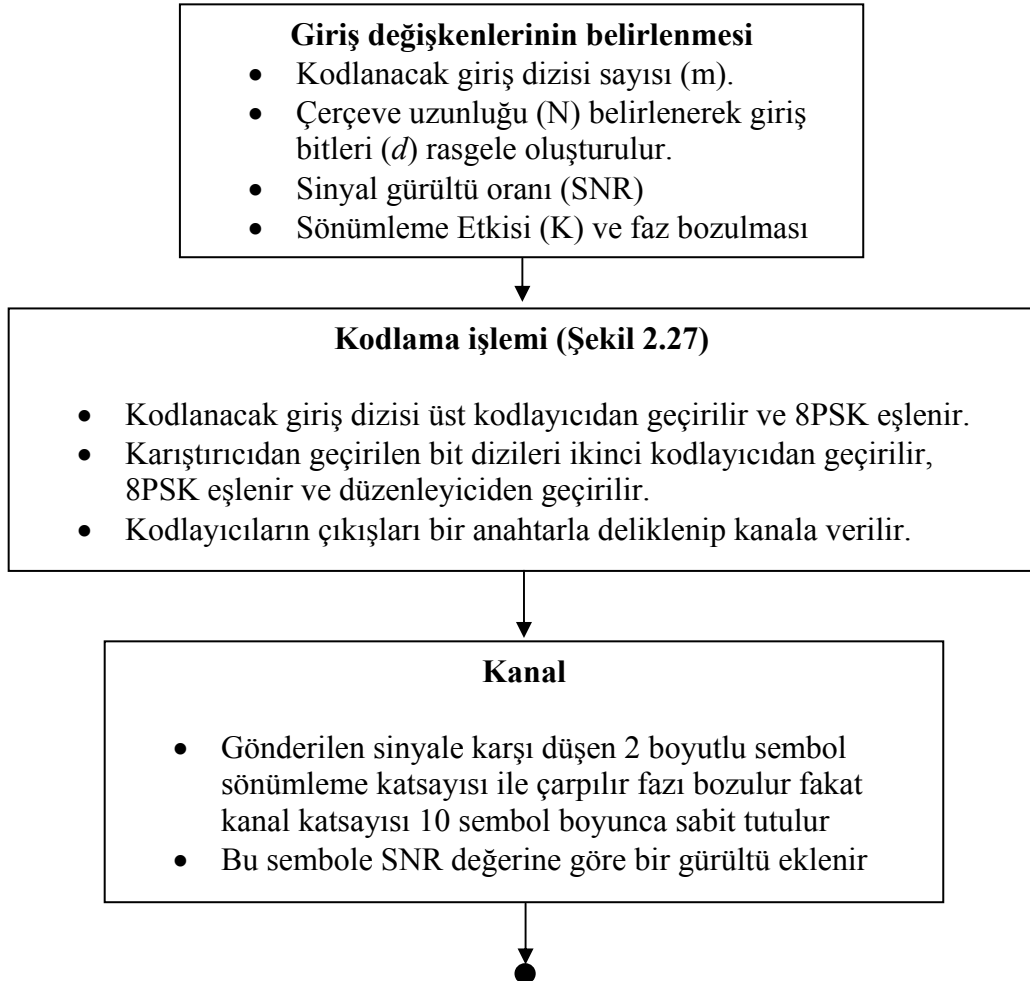
$$\nabla_{\rho_i} \Theta = \frac{2}{\sigma_l^2} \sum_{i=1}^N \sum_{m=1}^P \gamma_m^{(t)}(i) \Re \left\{ \left[y(i) - \rho_i' g(\phi_m) e^{j\theta_i'} \right] (g(\phi_m) e^{j\theta_i'})^* \right\} \quad (3.18 a)$$

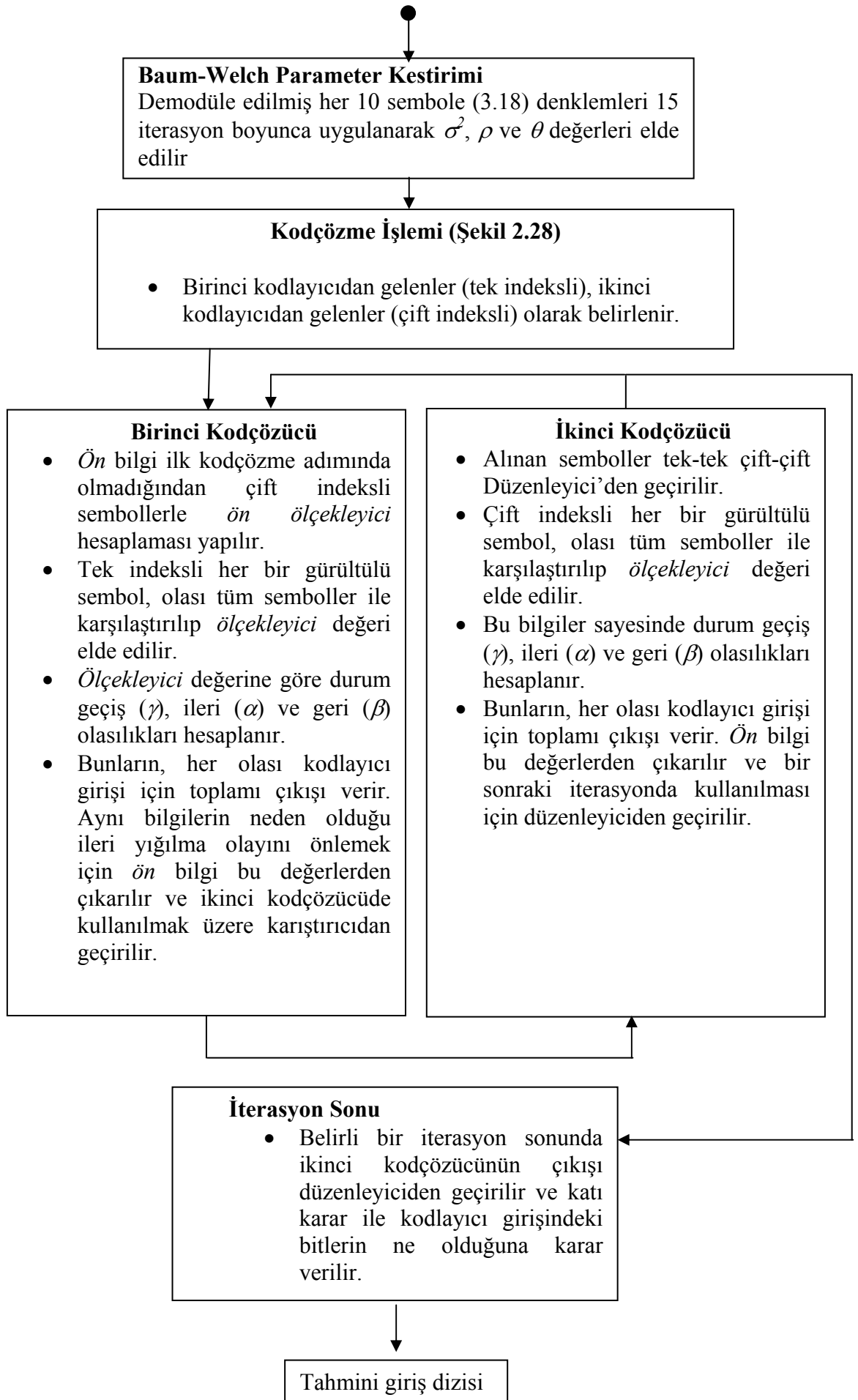
$$\nabla_{\theta_i'} \Theta = \frac{2}{\sigma_l'^2} \sum_{i=1}^N \sum_{m=1}^P \gamma_m^{(t)}(i) \Re \left\{ \left[y(i) - \rho_i' e^{j\theta_i'} (g(\phi_m)) \right] j e^{-j\theta_i'} \rho_i' g^*(\phi_m) \right\} \quad (3.18 b)$$

$$\nabla_{\sigma_l'^2} \Theta = -\frac{N}{2\sigma_l'^2} + \frac{1}{2\sigma_l'^4} \sum_{i=1}^N \sum_{m=1}^P \gamma_m^{(t)}(i) \left| y(i) - \rho_i' g(\phi_m) e^{j\theta_i'} \right|^2 \quad (3.18 c)$$

Simülasyonda sistemin verici, kanal ve alıcı blok diyagramlarının nasıl gerçekleştirildiğini ve hangi değişkenlerin nasıl ve nerede belirlendiğini göstermek amacı ile aşağıdaki gibi bir akış diyagramı verilmiştir.

Tablo 3.2 Kanal kestirimli Turbo kodlama akış şeması

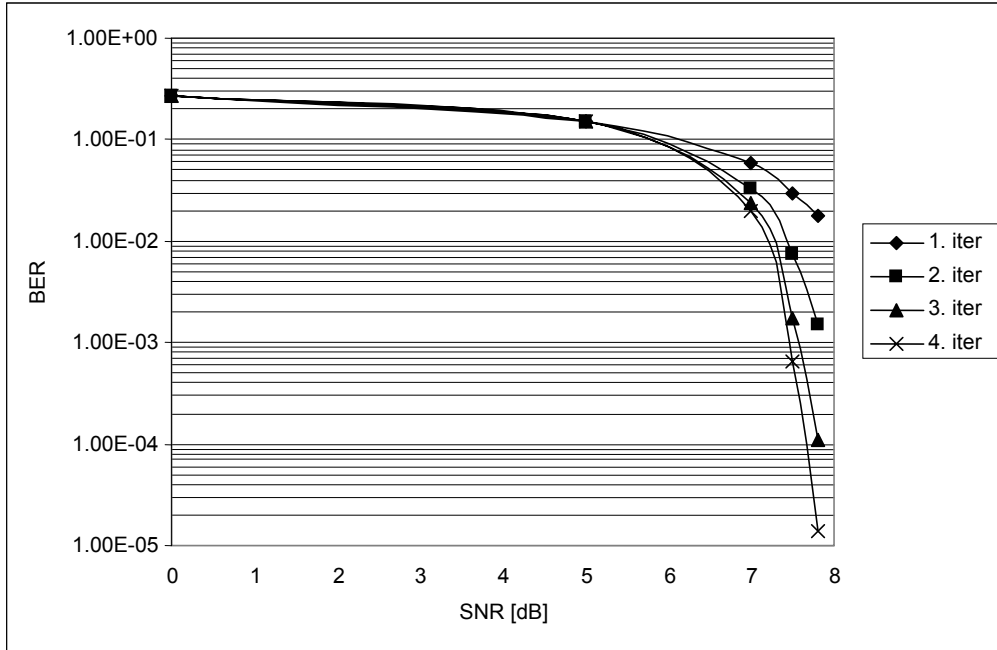




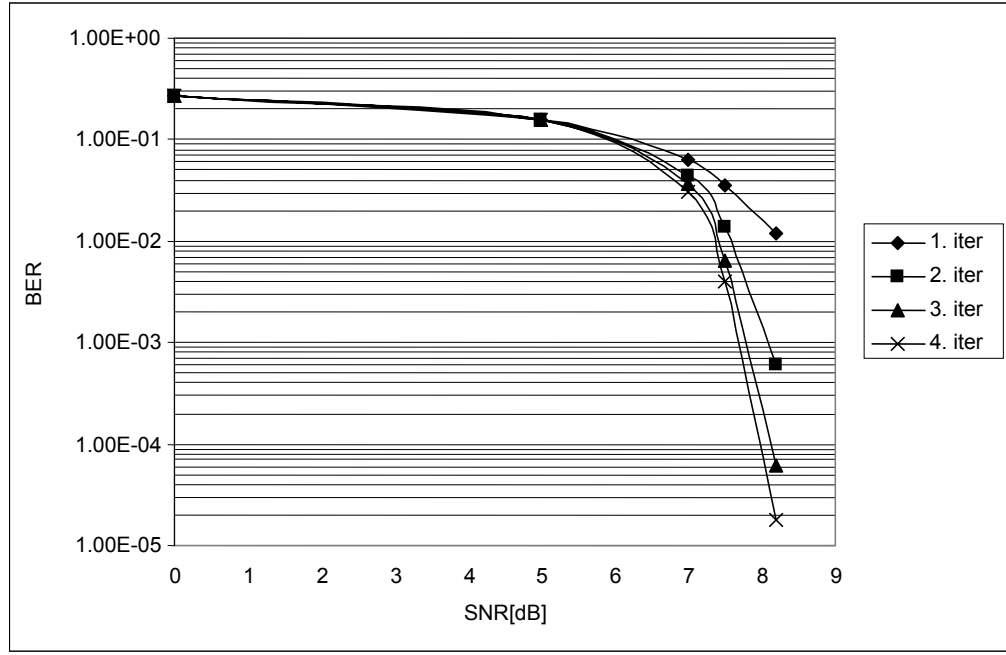
Denklem (3.16)'da kullanılan p değişkeni de her bir sinyalin oluşma olasılığını gösterir. Bu sistemde herbirinin oluşması olasılığı $1/8$ dir, bu yüzden $\mathbf{p}$ matrisi şu şekilde gösterilebilir.

$$\mathbf{p} = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \end{bmatrix} \quad (3.20)$$

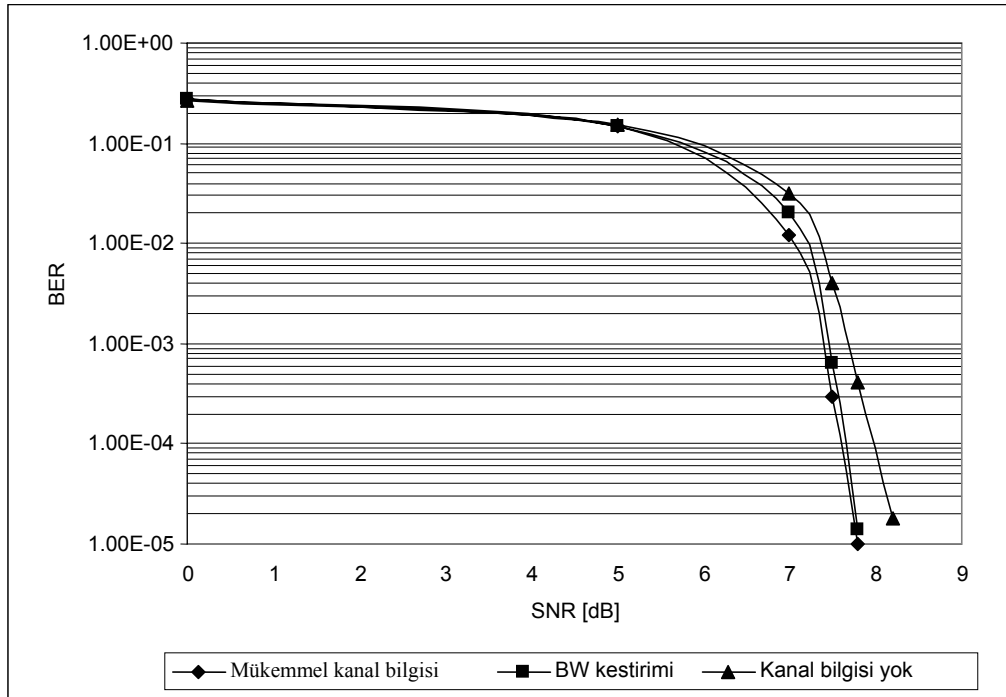
Alıcı tarafında gürültülü olarak alınan sinyaller öncelikle BW kestiriminden geçirilmiş, sönmüleme parametreleri elde edilmiş ve bu parametreler Turbo kafes kodlayıcıda kullanılmıştır.



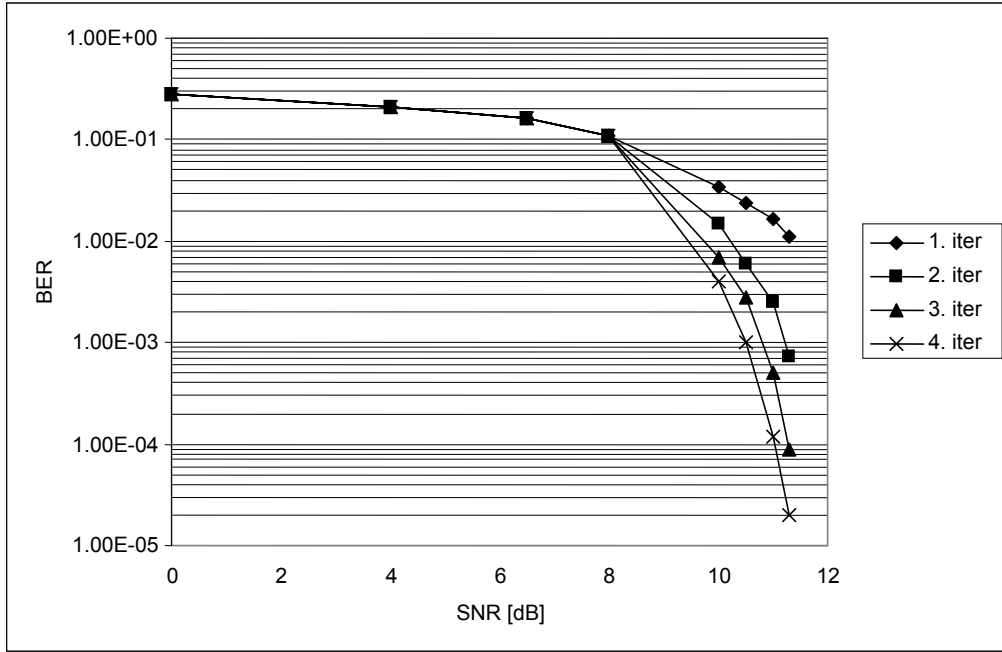
Şekil 3.23 K=10dB Rician kanal için BW kanal kestirimi kullanılarak elde edilen
TTCM başarımı



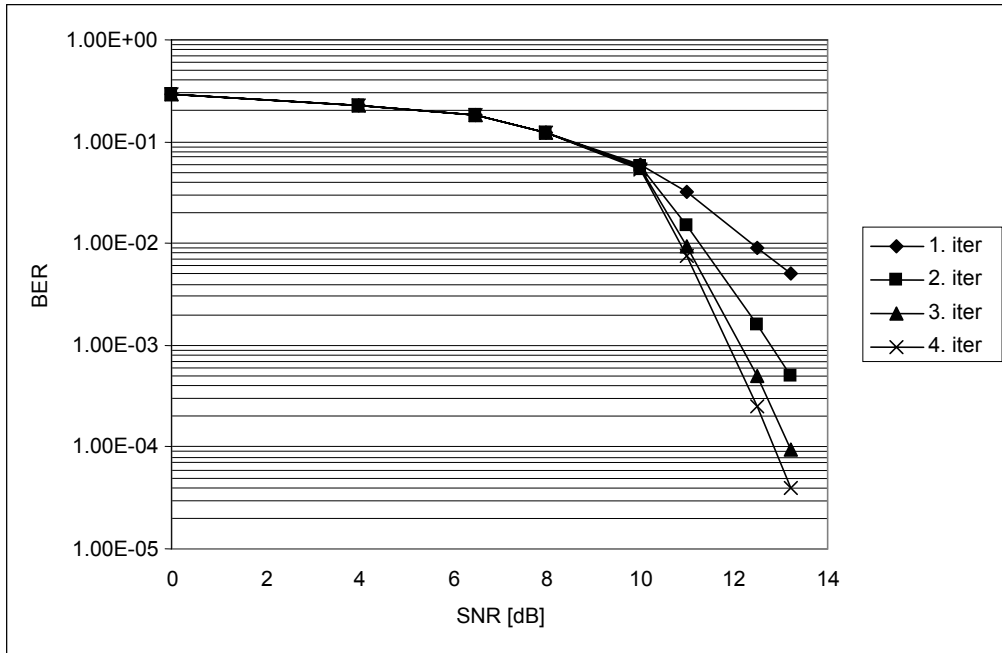
Şekil 3.24 $K=10$ dB Rician kanal için kanal bilgisi olmadan alınan TTCM başarıımı



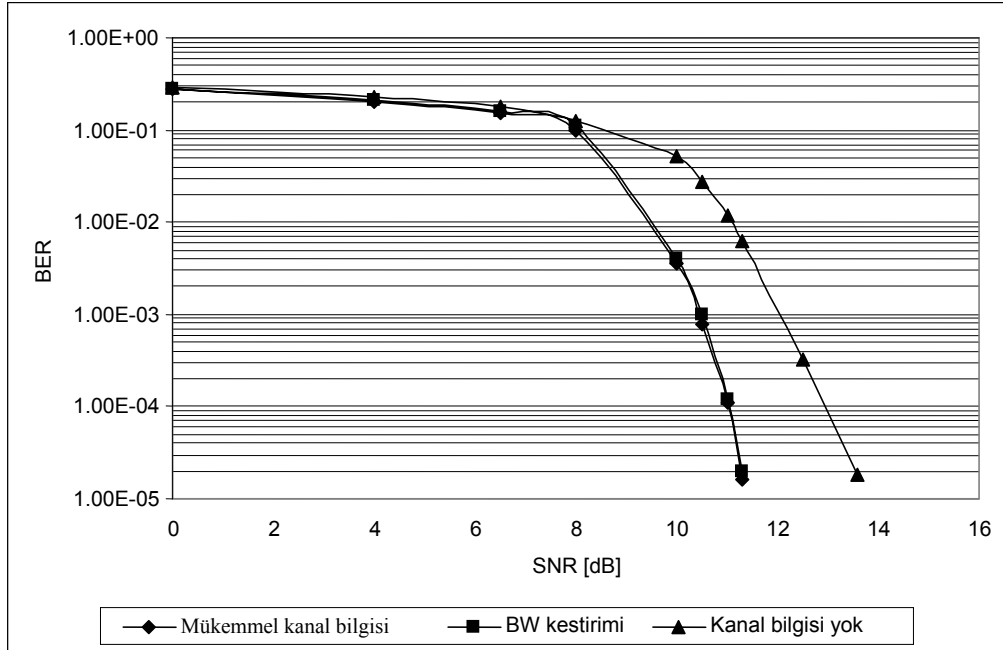
Şekil 3.25 $K=10$ dB Rician kanal için mükemmel kanal bilgisi olduğu, BW kestirimi yapıldığı ve kanal bilgis olmadığı durumlarda TTCM 4. iterasyon başarıımı



Şekil 3.26 Rayleigh kanal için BW kanal kestirimi ile alınan TTCM başarımı



Şekil 3.27 Rayleigh kanal için kanal bilgisi olmadan alınan TTCM başarımı



Şekil 3.28 Rayleigh kanal için mükemmel kanal bilgisi, BW kestirimi ve kanal bilgis olmadan alınan TTCM 4. iterasyon başarımları

Sonuçlara bakıldığında her iki durum için de BW algoritması ile kanal parametreleri kestirilerek Turbo kafes kod çözme işlemi yapıldığında elde edilen sonuçlar, kanal parametreleri tam olarak bilindiği durumda elde edilen sonuçlara çok yakındır.

Kanal parametrelerinin hiç bilinmemesi durumunda ise Rician kanalda yaklaşık 0.6 dB lik Rayleigh kanalda ise 2.5 dB lik bir kötüleşme olmaktadır. Gelen sinyallerin ön işleminden geçirilerek kanal parametrelerinin kestirilmesi ile bu kayıplar ortadan kaldırılabilmektedir.

Sistemin dezavantajı ise BW algoritmasının ek bir zaman kaybına neden olunmasıdır. BW algoritması BCJR algoritmasına benzerliği ile bilinen bir algoritmadır. Bundan dolayı ileri ve geri olasılıkların elde edilmesi ve iteratif olarak çözüm yapılması nedeniyle oldukça karmaşık olması, kanal parametrelerinin elde edilmesinde oldukça zaman almaktadır.

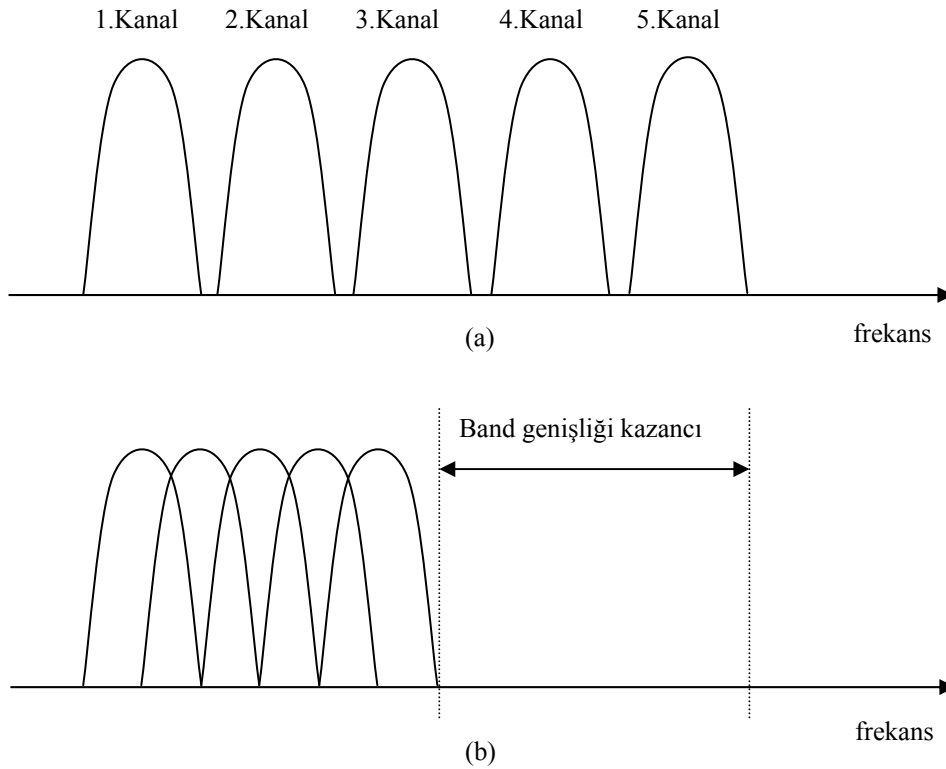
3.3. Dik Frekans Bölümlemeli Modülasyon

Dik Frekans Bölümlemeli Modülasyon (Orthogonal Frequency Division Multiplexing, OFDM) tek bir data dizisinin belli bir sayıdaki alt-taşıyıcılarla iletildiği çok-taşıyıcılı iletişimin özel bir halidir. Burada OFDM bir modülasyon çeşidi gibi görülebildiği gibi aynı zamanda çoklayıcı tekniğidir. OFDM'in en önemli kullanılma sebeplerinden birisi, frekans seçici sönmülemeli kanallara ve dar-band bozulmasına karşı sağlamlığıdır. Tek-taşıyıcılı sistemlerde, bir sönmüleme ya da bozucu etki tüm iletim hattının etkileyerek iletişimi bozmakta, fakat çok-taşıyıcılı sistemlerde ise sadece alt-taşıyıcıların çok az bir bölümünün etkilenmesine neden olmaktadır. Daha sonra hata düzeltici kodlama yöntemleri ile bu çok az hata da kolaylıkla ortadan kaldırılabilmektedir.

Klasik paralel data sistemlerinde 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 önlemek için alt-taşıyıcı frekanslarındaki girişimini önlemek iyi bir çözüm gibi görünmektedir. Fakat bu yöntem, kanalın frekans spektrumunun verimli bir biçimde kullanılmamasına neden olmaktadır. Bu verimsizliği ortadan kaldırmak için 1960 ların ortasında önerilen yöntem paralel data ve alt-kanalların üst üste bindiği frekans bölümlemeli çoğullama (Frequency Division Multiplexing, FDM) idi.

OFDM kullanılan sistemlerde kanal parametrelerinin ML ve pilot sembol (*Pilot-symbol-aided*) kullanarak bulunması [40,41]'de verilmiştir. Turbo kodların OFDM kullanıldığı durumda sönümlmeli kanallardaki performansı [42,43]'de gösterilmiştir. Aynı zamanda OFDM, sayısal televizyon sistemleri (DVB-T) ve sayısal uydu haberleşmesi kanalları için [44,45]'de sunulmuştur.

Şekil 3.29 geleneksel üst üste gelmeyen çok-taşıyıcılı sistemle frekansların üst üste geldiği modülasyon tekniği arasındaki farkı göstermektedir. Frekansları üst üste gelen yöntemin kullanılması band genişliğinde %50'lik bir kazançta neden olmaktadır. Üst üste gelme yöntemini gerçeklemek için modüle edilmiş farklı taşıyıcılar arasında bir dikliğe gerek vardır, bu da kanallar arasındaki crosstalk'un ihmal edilmesi anlamına gelir.



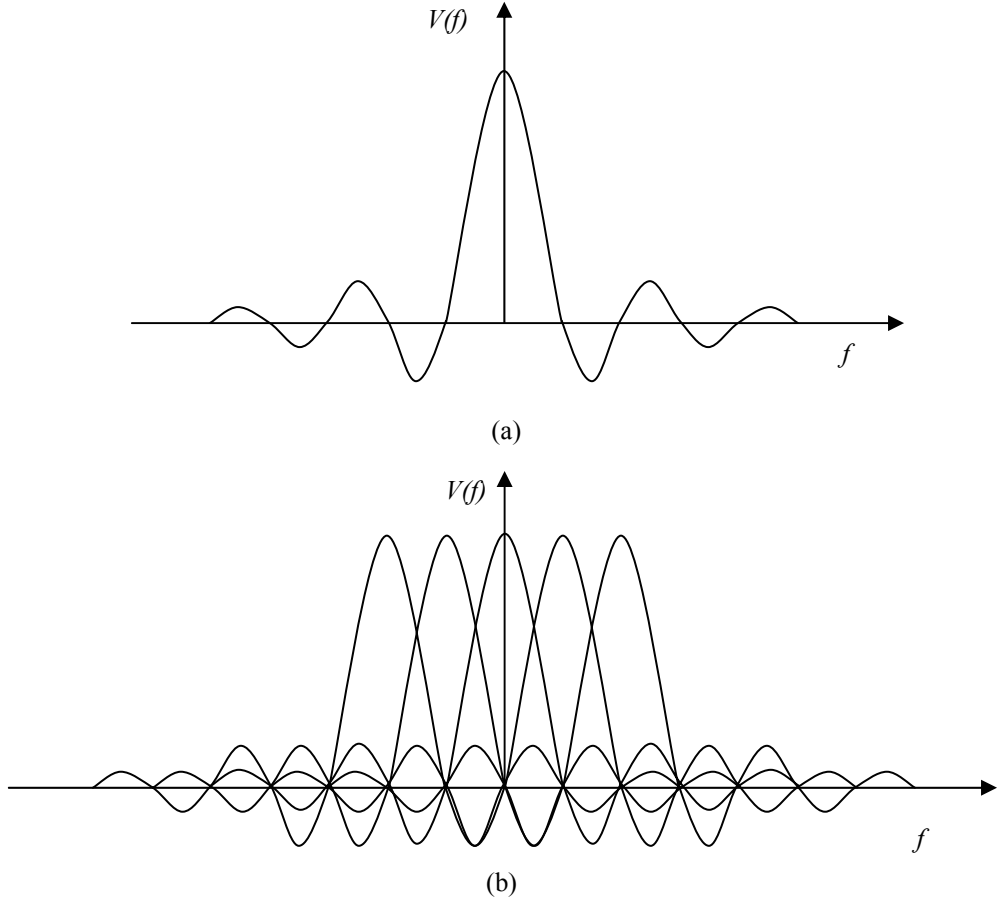
Şekil 3.29 (a) geleneksel çok-taşıyıcılı tekniği (b) dik çok-taşıyıcılı modülasyon tekniği

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 koruma bandı sokulur ve bu frekans domeninde verimliliğin düşmesine sebep olur.

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 kümesi gibidir, her taşıyıcıyı taşıyıcı frekansları ile çarpıp bir sembol periyodu süresince integre eder. Farklı taşıyıcı frekansları ile çarpılanlar diklikten dolayı bir sembol süresi boyunca integralleri sıfır olurlar.

Bir çok araştırmacı yüksek verimlilikli dik frekanslı çok taşıyıcılı sistemler üzerine çalışmışlardır. 1971 yılında Weinstein ve Ebert [46] ayrık Fourier dönüşümünü (Discrete Fourier Transform, DFT) paralel data iletiminde modülasyon ve demodülasyonun bir parçası olarak kullanmışlardır. Şekil 3.30 (a) tek bir alt-kanalın frekans spektrumunu göstermektedir ki bu da $\text{sinc}(f) = \sin(\pi f) / \pi f$ fonksiyonu ile adlandırılır. Frekansları çoğullanmış OFDM sinyallerinin frekans spektrumu da Şekil 3.30 (b) de gösterilmektedir. Şekil 3.30 da tam orta frekanslarda herhangi bir girişim olmadığı görülür. Bu yüzden eğer alıcıda DFT kullanılır ve her alt-taşıyıcının orta frekansı ile korelasyonu hesaplanırsa, iletilen datayı girişim olmadan elde edebiliriz.

Frekans bölümlmeli çoğullamanın içerdği alt-taşıyıcı osilatörlerin kümelerini ve uyumlu demodülatörleri ortadan kaldırmak için DFT'nin iyi bir uygulaması olan hızlı Fourier dönüşümü (Fast Fourier Transform, FFT) tamamen sayısal bir donanım olarak kullanılabilir.



Şekil 3.30 (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 [47].

1960’larda OFDM birçok askeri sistemlerde, 1980’lerde de yüksek hızlı modemlerde, sayısal mobil haberleşmede ve yüksek yoğunluklu kayıtlarda kullanılmıştır. 1990’larda ise mobil radyo FM kanalları üzerinden geniş bantlı data haberleşmesinde, yüksek bit oranlı sayısal alt-taşıyıcılı hatlarda, asimetrik sayısal alt-taşıyıcılı hatlarda, çok-yüksek-hızlı sayısal alt-taşıyıcılı hatlarda, sayısal ses yayıncılığında ve yüksek tanımlamalı televizyon yayıncılığında kullanılmıştır [48-53].

3.3.1. IFFT Kullanılarak Alt-Taşıyıcıların Oluşturulması

Bir OFDM sinyali PSK ya da QAM kullanılarak modüle edilmiş alt-taşıyıcıların toplamından oluşur. Bir OFDM sinyali, d_i kompleks bir sembol, N_s alt-taşıyıcı sayısı, T sembol süresi, f_c taşıyıcı frekansı ve t_s başlama anı olmak üzere aşağıdaki gibi yazılabilir.

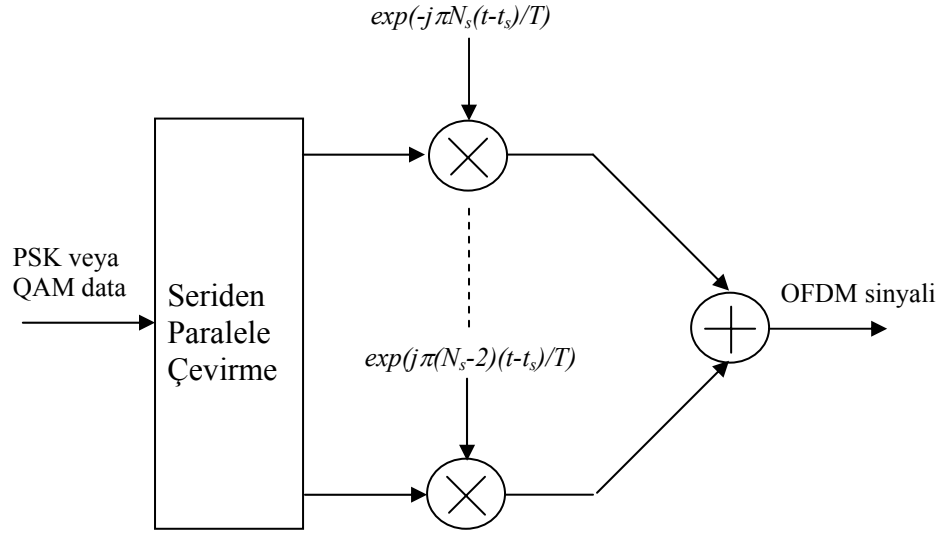
$$s(t) = \begin{cases} \operatorname{Re} \left\{ \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s/2} \exp(j2\pi(f_c - \frac{i+0.5}{T})(t-t_s)) \right\} & , t_s \leq t \leq t_s + T \\ 0 & , t < t_s \text{ ve } t > t_s + T \end{cases} \quad (3.21)$$

Temel band haberleşmesinde ise taşıyıcı frekansı olmadığı için (3.21) formülü yerine genellikle (3.22) formülü 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.

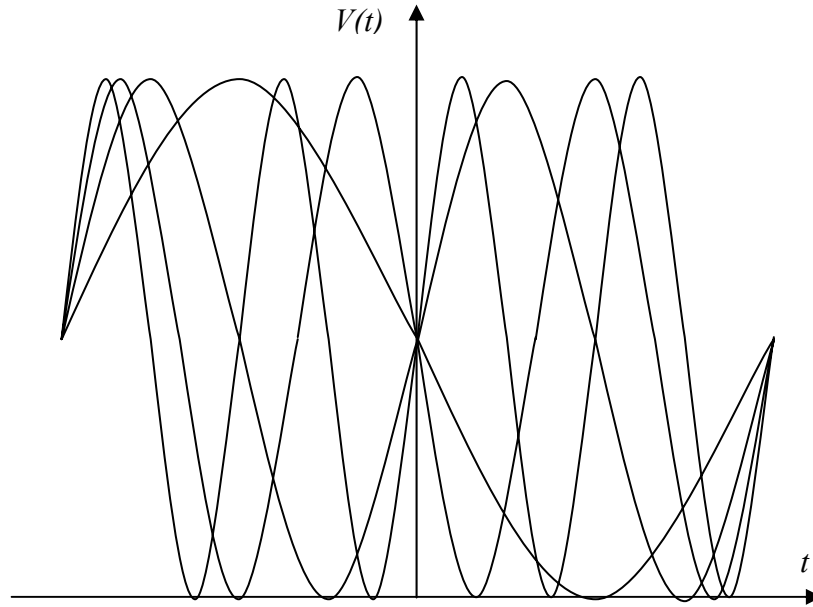
$$s(t) = \begin{cases} \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s/2} \exp(j2\pi \frac{i}{T}(t-t_s)) & , t_s \leq t \leq t_s + T \\ 0 & , t < t_s \text{ ve } t > t_s + T \end{cases} \quad (3.22)$$

Şekil 3.31 OFDM modülatörünün işlevini blok diyagram olarak göstermektedir. Şekil 3.32 dört alt-taşıyıcılı bir OFDM sinyali göstermektedir. Burada tüm alt-taşıyıcılar aynı faz ve genliğe sahiptirler, fakat pratikte genlik ve faz kanala bağlı olarak farklı şekilde modüle edilebilir. Her alt-taşıyıcının T aralığında tam sayıda periyodu vardır, ve bu sayı en yakınındaki alt-taşıyıcıdan bir farklıdır. Bu özellik alt-taşıyıcılar arasındaki diklik için önemlidir. Eğer, (3.22) eşitliğindeki j . alt-taşıyıcı j/T frekansı ile demodüle edilir ve sonra T süresi boyunca integrali alınırsa sonuç (3.23)'deki gibi olur. Formüle bakıldığında karmaşık taşıyıcı T süresi boyunca integre edilir. Demodüle edilmiş j . alt-taşıyıcı için bu integrasyonun sonucu $d_{j+N_s/2}$ 'nin bir oranını verir. Diğer tüm alt-taşıyıcılar için integrasyonun sonucu sıfırdır. Çünkü frekans farkı $(i-j)/T$, T integral süresi içinde tam sayıda periyod yaratır, bu nedenle integral sonucu her zaman sıfırdır.

$$\begin{aligned} & \int_{t_s}^{t_s+T} \exp(-j2\pi \frac{j}{T}(t-t_s)) \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s/2} \exp(j2\pi \frac{i}{T}(t-t_s)) dt \\ &= \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s/2} \int_{t_s}^{t_s+T} \exp(j2\pi \frac{i-j}{T}(t-t_s)) dt = d_{j+N_s/2} T \end{aligned} \quad (3.23)$$



Şekil 3.31 OFDM modülatörü



Şekil 3.32 Dört alt-taşıyıcılı OFDM sinyali

OFDM alt-taşıyıcıları arasındaki diklik başka bir şekilde de gösterilebilir. (3.21)

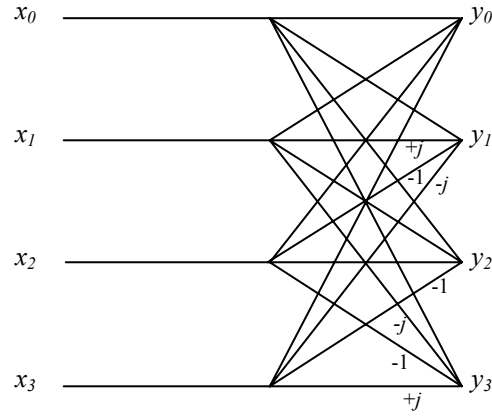
denklemine göre, her OFDM sumbolü T -saniye süresinde sıfır olmayan alt-

taşıyıcılardan oluşur. Bundan dolayı, bir sembolün spektrumu, alt-taşıyıcı frekansında bulunan Dirac sinyaller gurubu ile T -saniye boyunca bir, diğer zamanlarda sıfır olan kare dalganın spektrumu ile konvolüsyonudur. Kare dalganın genlik spektrumu, $1/T$ ile çarpımını tam yapan tüm f ler için sıfır veren $\text{sinc}(\pi f T)$ dir. Bu etki Şekil 3.30 (b)'de gösterilmektedir. Her alt-taşıyıcı spektrumunun maksimumunda diğer alt-taşıyıcıların spektrumları sıfır olur. Temel olarak Şekil 3.32'de, semboller arası bozucu olmayan bir dalga şekli için Nyquist kriterinin yerine getirildiği görülür.

Denklem (3.22) 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 (3.24) deki gibi elde edilir bu da ters ayrık Fourier dönüşümüdür (Inverse Discrete Fourier Transform, IDFT). Bu dönüşüm, pratikte ters hızlı Fourier dönüşümü (Inverse Fast Fourier Transform, IFFT) olarak kullanılır. IFFT, IDFT'den oldukça daha az işlem içerir.

$$s(n) = \sum_{i=0}^{N_s-1} d_i \exp(j2\pi \frac{in}{N}) \quad (3.24)$$

IFFT deki çarpma işlemi sayısı radix-4 algoritması ile azaltılabilir. Şekil 3.33 dört-noktalı radix-4 kelebeği olarak bilinen IFFT'yi göstermektedir. x_0 'dan x_3 'e kadar olan girişler sadece toplama ve faz değişimi işlemleriyle y_0 'dan y_3 'e kadar olan çıkışlar bulunur. Aynı zamanda radix-4 kelebeği daha büyük ölçütteki IFFT'ler için de kullanılır.

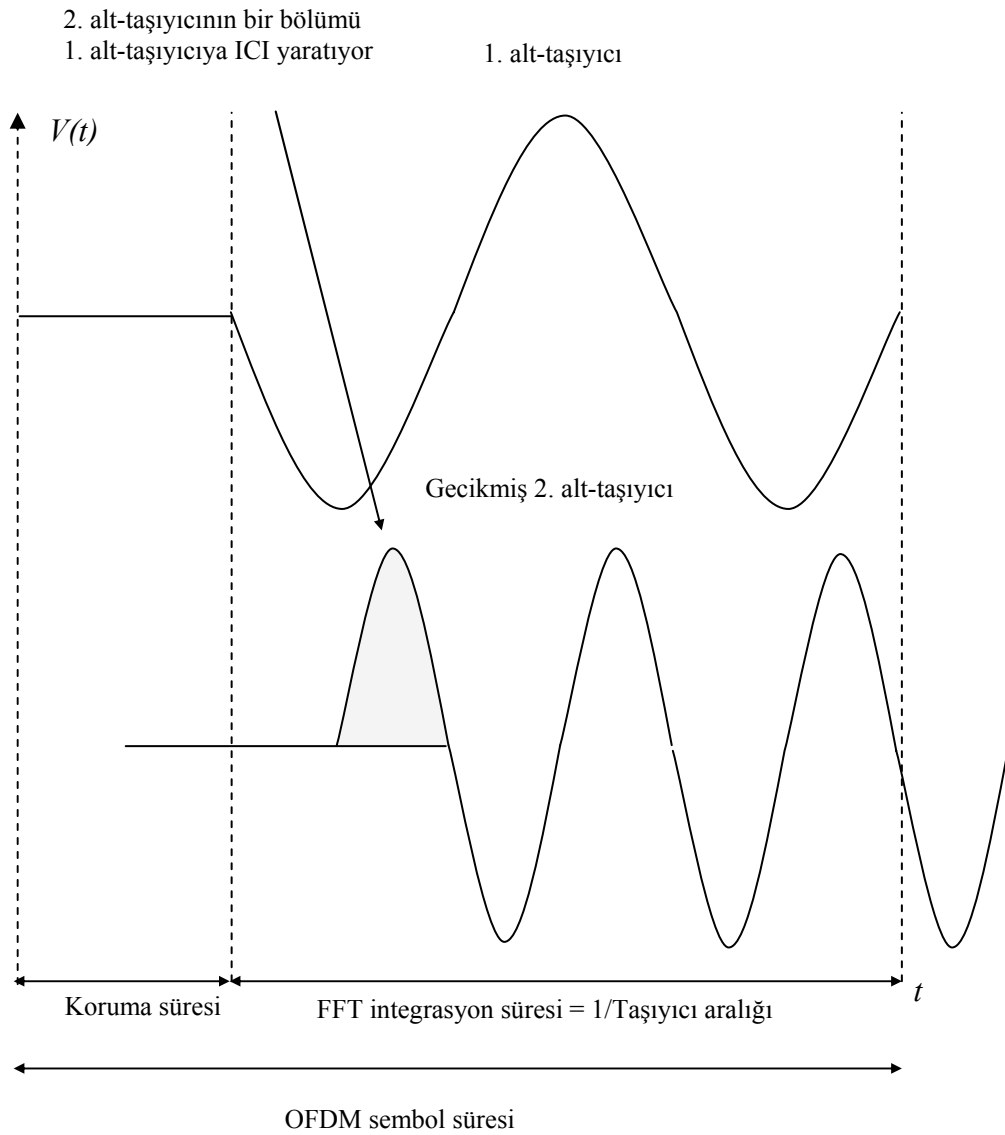


Şekil 3.33 Radix-4 kelebeği

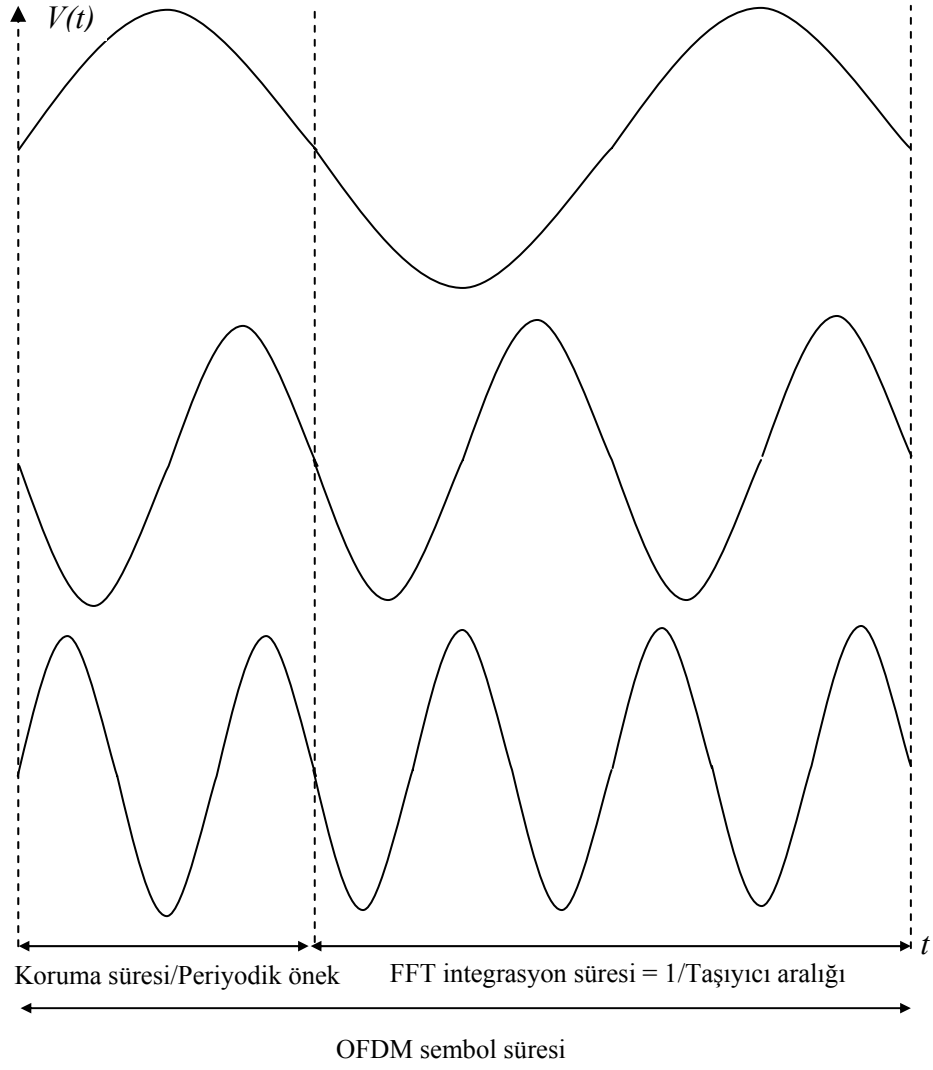
3.3.2. Koruma Süresi ve Periyodik İlaveler

OFDM yönteminin kullanılmasının en önemli nedeni çoklu-yol gecikme yayılımıdır. Giriş dizisini N_s adet alt-taşıyıcıya bölerek sembol süresi N_s defa küçültülmüş olur bu da sembol süresine göre çoklu-yol gecikme yayılımını azaltır. Semboller-arası bozulmayı (Inter Symbol Interference, ISI) tamamen ortadan kaldırmak için OFDM sembolüne koruma süresi eklenir. Koruma süresi beklenen en uzun gecikme yayılımından daha büyük seçilir. Böylece bir önceki sembolün bir sonraki sembole karışması önlenir. Koruma süresinde hiçbir sinyal olmazdı. Bu durumda taşıyıcılar-arası bozulma (Inter Carrier Interference, ICI) ortaya çıkar. ICI alt-taşıyıcılar arasında oluşan ve hiç bir şekilde dik olmayan girişimdir. Bu etki Şekil 3.34'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. alt-taşıyıcıda bir girişime sebebiyet verecektir.

ICI'yı ortadan kaldırmak için koruma süresi boyunca OFDM sembolü Şekil 3.35'de gösterildiği gibi periyodik olarak tekrarlanır. Bu, gecikme süresi koruma süresinden küçük olduğu sürece, OFDM sembolünün gecikmiş benzerinin FFT zaman aralığındaki tekrarları hep tam sayılı olacağını garanti eder. Sonuç olarak, çoklu-yol ortamında koruma süresinden küçük olan gecikmeler ICI'ya sebebiyet vermezler.



Şekil 3.34 Koruma süresi boyunca hiç bir sinyal olmamasının çoklu-yol ortamlardaki etkisi



Şekil 3.35 Periyodik önekli OFDM sinyali

3.3.3. Pencereleme

Sinyalin diğer bir sinyalle birleştiği yerlerde modülasyondan doğan keskin geçişler oluşur. Sonuç olarak, band dışı spektrum, *sinc* fonksiyonuna göre daha yavaş bir şekilde azalır. Fakat, periyod sayısı yüksek alt-taşıyıcılar için frekans spektrumu daha hızlı azalır.

Spektrumu daha hızlı aşağı çekebilmek için her OFDM sembole pencereleme uygulanır. Pencereleme OFDM sembolünün başlangıç ve bitiş noktalarında yumuşak bir şekilde sıfıra gitmesini sağlar. En çok kullanılan pencere tipi yükseltilmiş cosinüs (*raised cosine*) penceresidir ve şu şekilde tanımlanır.

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

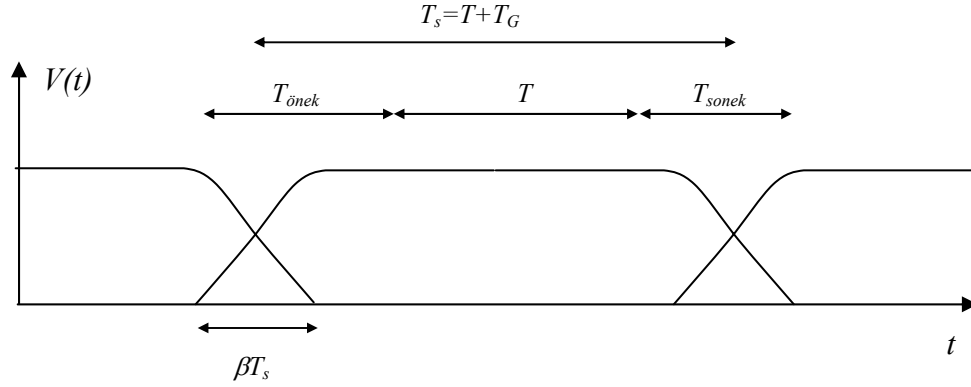
Burada T_s sembol süresidir ve bu süre toplam sembol süresinden kısadır. OFDM sinyalinin zamana göre değişimi Şekil 3.36'daki gibidir.

(3.21) denklemi ile tanımlanan, $t = t_s = kT_s$ zamanında başlayan OFDM sembolüne ön ek ve pencereleme uygulandığında aşağıdaki hali alır.

$$s_k(t) = \begin{cases} \operatorname{Re} \left\{ w(t - t_s) \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} d_{i+N_s(k+1/2)} \exp(j2\pi(f_c - \frac{i+0,5}{T})(t - t_s - T_{\text{önek}})) \right\} & , t_s \leq t \leq t_s + T_s(1 + \beta) \\ 0 & , t < t_s \text{ ve } t > t_s + T_s(1 + \beta) \end{cases}$$

Pratikte OFDM sinyali şu şekilde oluşturulur: 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.



Şekil 3.36 OFDM periyodik eki ve pencereleme

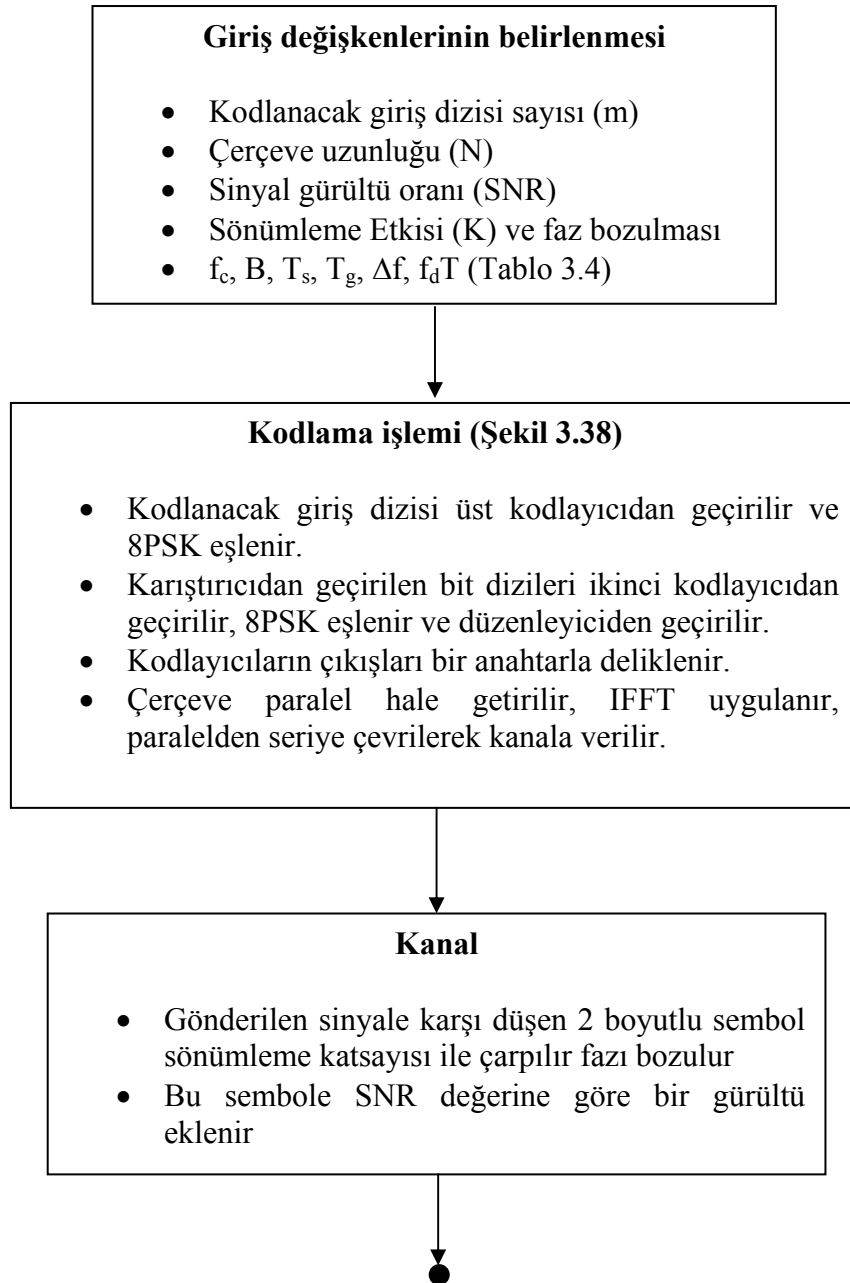
Simülasyonda sistemin verici, kanal ve alıcı blok diyagramlarının nasıl gerçekleştirildiğini ve hangi değişkenlerin nasıl ve nerede belirlendiğini göstermek amacı ile Tablo 3.3’de akış diyagramı verilmiştir.

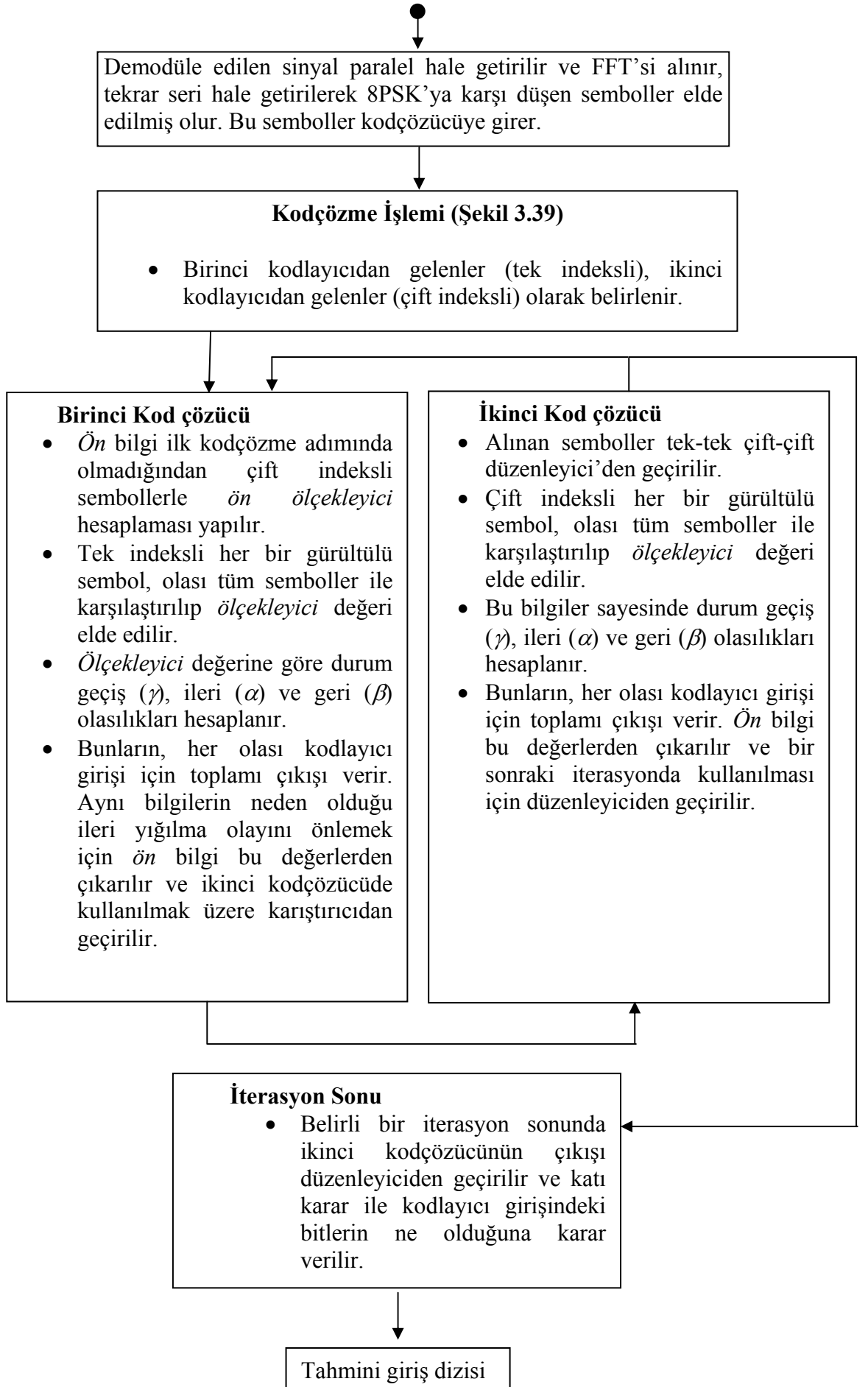
3.3.4. Simülasyon Sonuçları

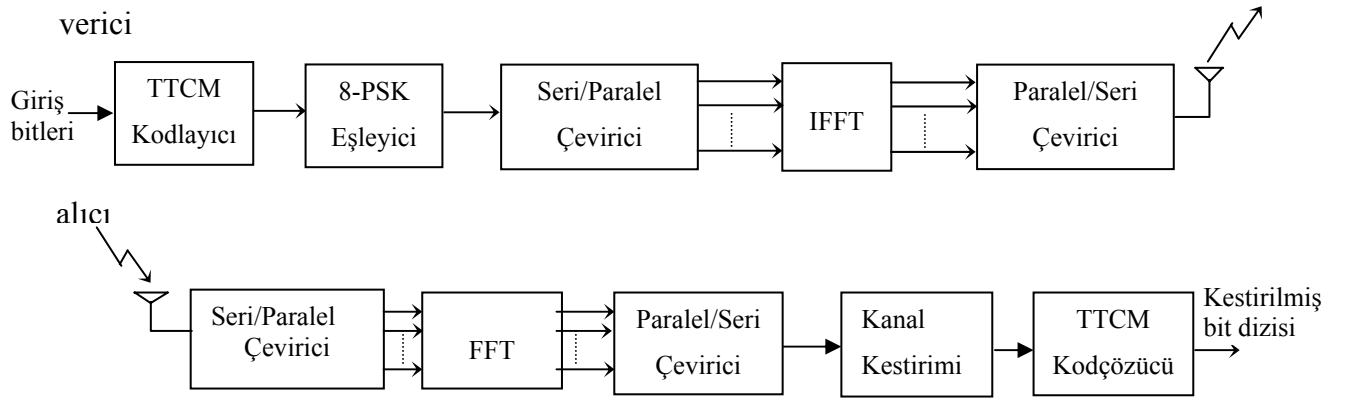
TTCM sisteminde OFDM kullanıldığında oluşacak sistemin genel yapısı Şekil 3.37’de verilmiştir. Daha açıkça görebilmek için bu sistemin kodlayıcı ve kod çözücü yapıları da Şekil 3.38 ve 3.39’da verilmiştir. Simülasyon MATLAB programı kullanılarak hazırlanmıştır. PIV 1.5 GHz bir bilgisayarda 4 iterasyonda, başlangıçta 256 bit uzunluklu bir çerçeve için 6 sn ve 1024 bit uzunluklu bir çerçeve için 16.5 sn sürmektedir.

Yukarıda da bahsedilmiş olduğu gibi OFDM ile ilgili birçok değişken vardır. Simülasyonda kullanılan bu parametrelerin değerleri Tablo 3.4’de açık olarak verilmiştir. Buna göre simülasyon aynı band genişliğide 256 ve 1024 alt taşıyıcı olduğu durumda AWGN, Rician ve Rayleigh kanalları için yapılmıştır.

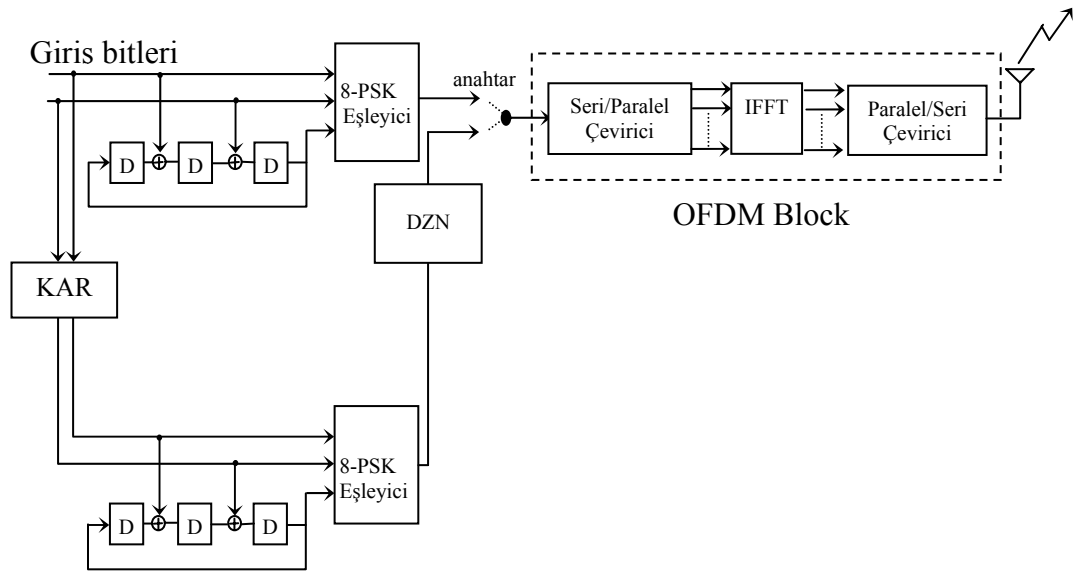
Tablo 3.3 OFDM ile bütünleşik Turbo kodlama akış şeması



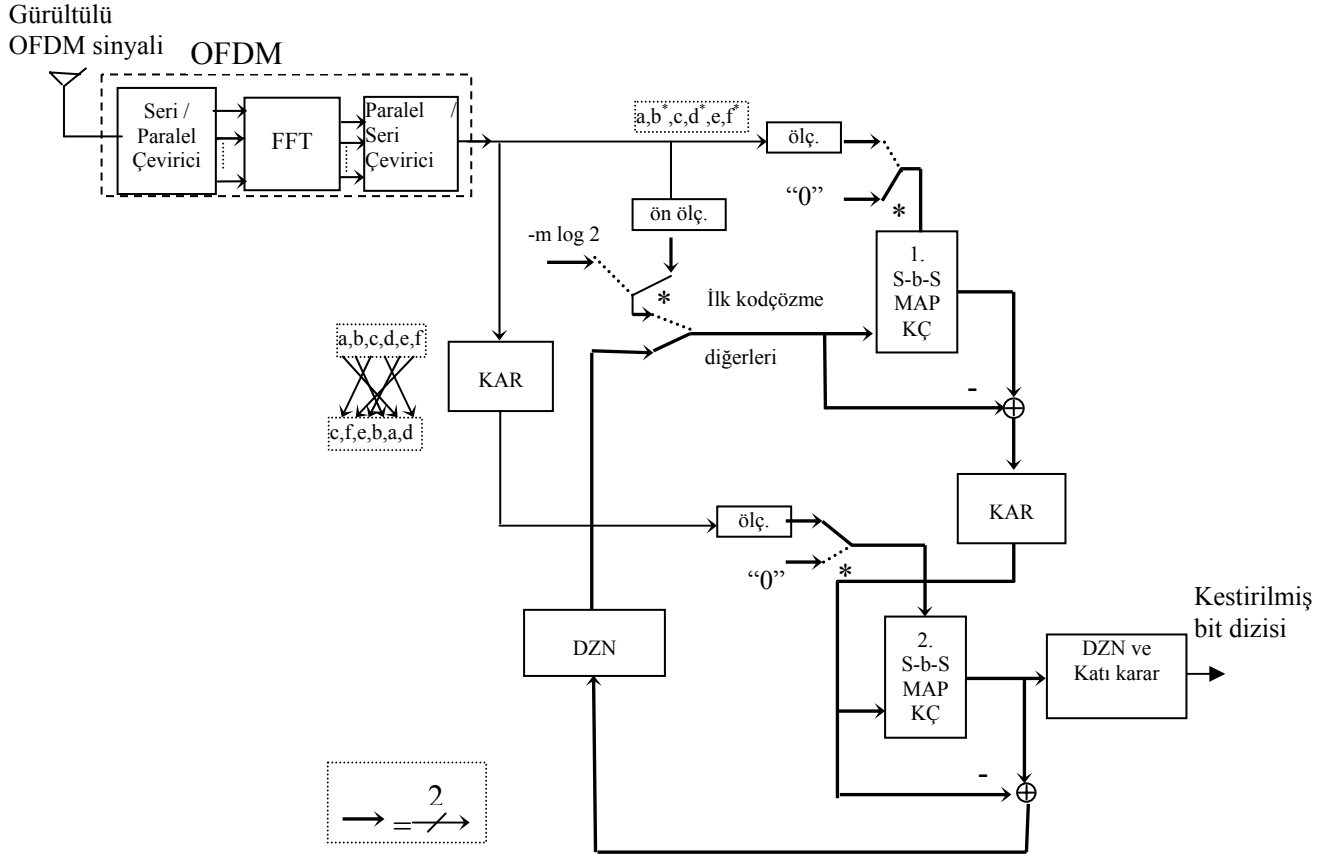




Şekil 3.37 OFDM'li bir TTCM sistemini alıcı ve verici genel yapısı

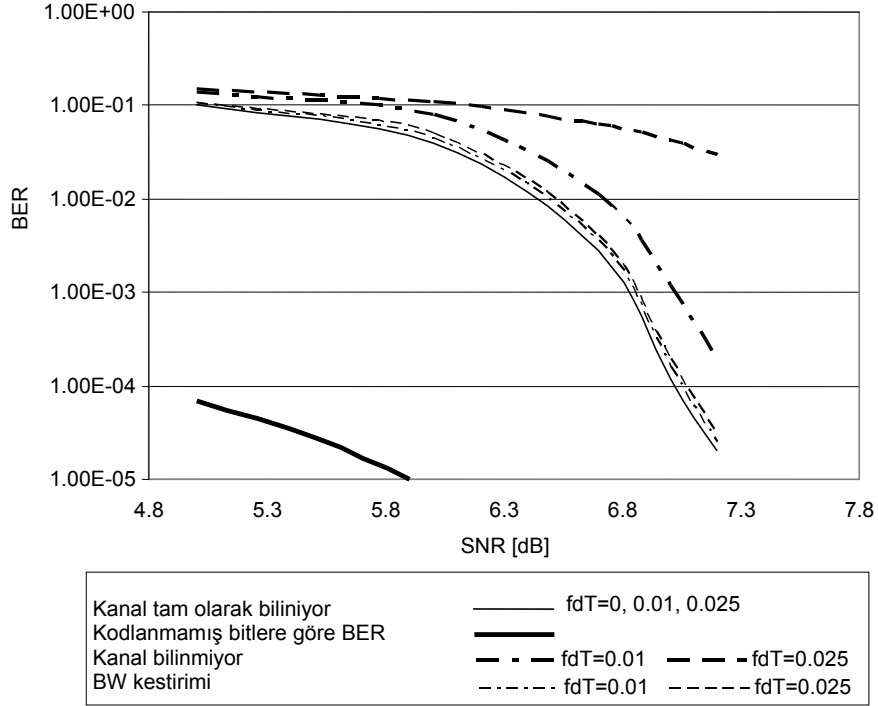


Şekil 3.38 TTCM-OFDM kodlayıcı yapısı

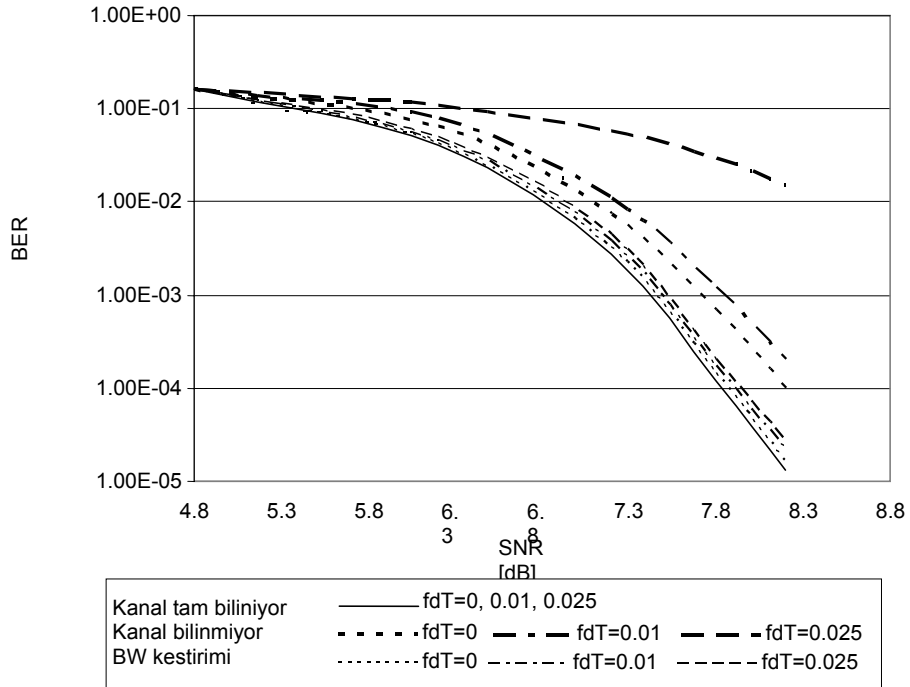


Tablo 3.4 OFDM parametreleri

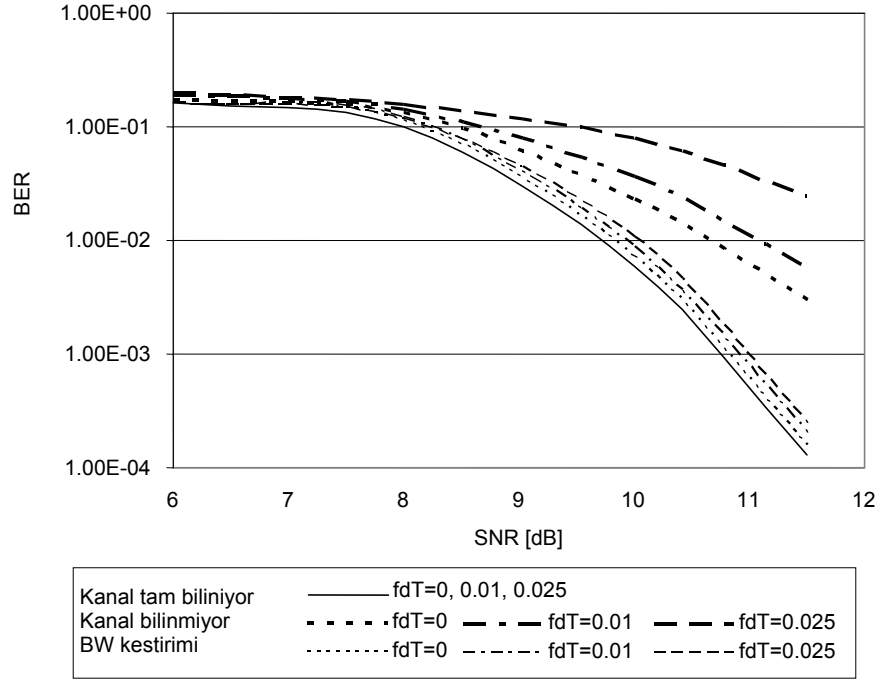
Parametreler	Değerleri
Taşıyıcı frekansı	$f_c=5$ GHz
Band genişliği	$B=80$ MHz
Alt taşıyıcı sayısı	$N=256, 1024$
Modülasyon türü	8PSK
Etkin sembol süresi	$T_s=6.4\mu s$
Koruma Süresi	$T_g=1.25\mu s$
Alt-taşıyıcı aralığı	$\Delta f=312.5$ kHz, 78.125 kHz
Paket yapısı	256, 1024 symbols with no pilot symbol
Kodlama yöntemi	TTCM
Sönümleme modeli	AWGN, Rician ($K=10$ dB), Rayleigh
Doppler kayması	$f_d T=0, 0.01, 0.025$
Kanal kestirimi	Tam biliniyor Baum-Welch Algoritması Kanal bilgisi yok



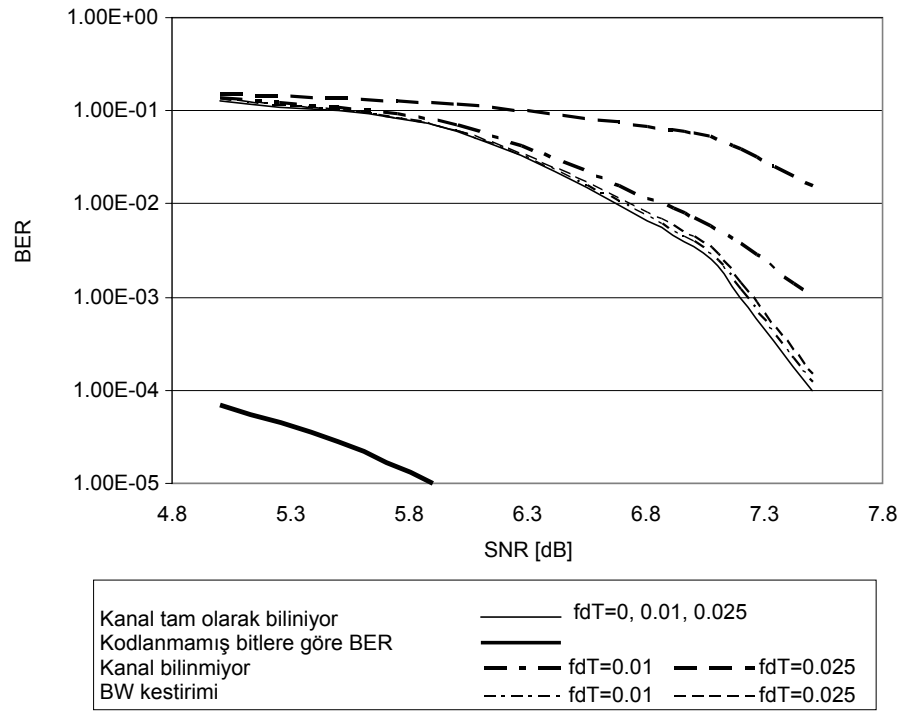
Şekil 3.40 AWGN kanalda $N=1024$ ve alt-taşıyıcı=1024 olduğu durumda TTCM-OFDM'in 4. iterasyondaki başarımı



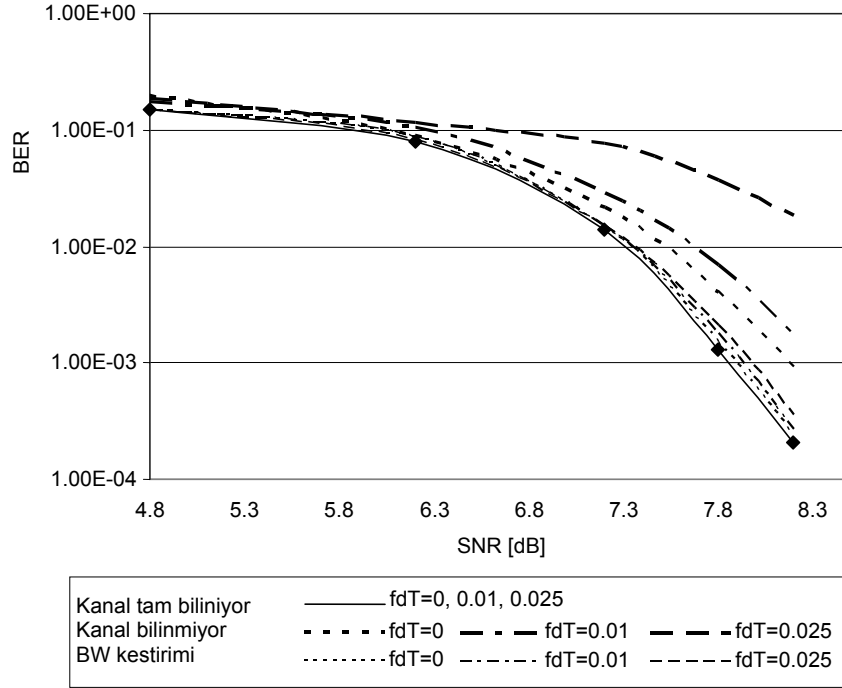
Şekil 3.41 Rician Kanalda ($K=10\text{dB}$) $N=1024$ ve alt-taşıyıcı=1024 olduğu durumda TTCM-OFDM'in 4. iterasyondaki başarımı



Şekil 3.42 Rayleigh kanalda $N=1024$ ve alt-taşıyıcı=1024 olduğu durumda TTCM-OFDM'in 4. iterasyondaki başarımı

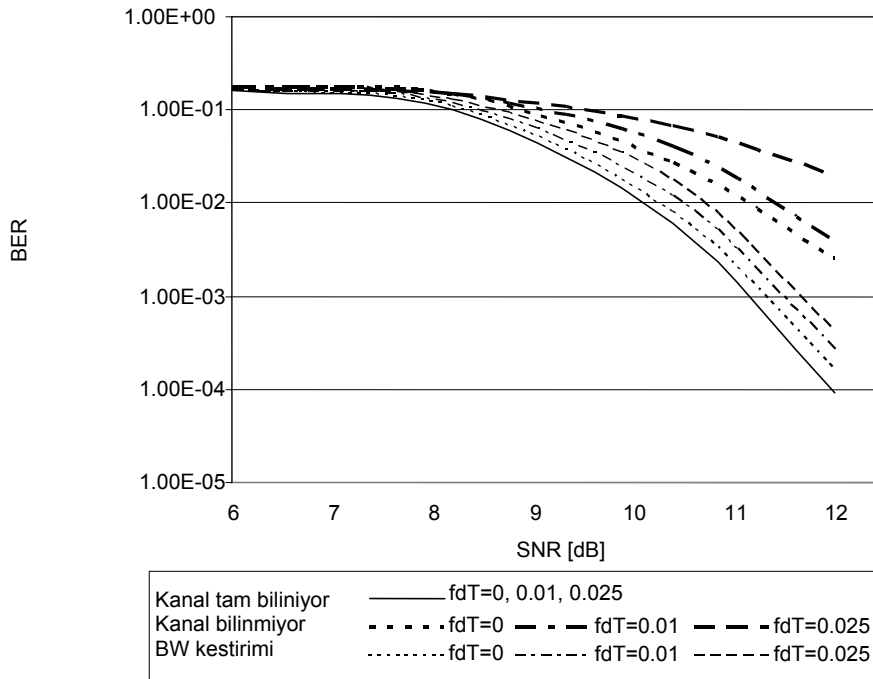


Şekil 3.43 AWGN kanalda $N=256$ ve alt-taşıyıcı=256 olduğu durumda TTCM-OFDM'in 4. iterasyondaki başarımı



Şekil 3.44 Rician Kanalda ($K=10\text{dB}$) $N=256$ ve alt-taşıyıcı=256 olduğu durumda

TTCM-OFDM'in 4. iterasyondaki başarımı



Şekil 3.45 Rayleigh kanalda $N=256$ ve alt-taşıyıcı=256 olduğu durumda TTCM-

OFDM'in 4. iterasyondaki başarımı

Simülasyon sonuçlarına bakıldığı zaman, performans üzerinde kanal sönümleme etkisinden çok Doppler kaymasının etkin olduğu görülmektedir. OFDM sistemlerinde genellikle paketin başına *pilot sembol* denilen bilinen semboller konulur ve bununla kanal hakkında bilgi edinilmeye çalışılır. Fakat bu tezde bu yöntemin yerine kanal kestirimi için BW algoritması kullanılmıştır.

Alt taşıyıcı sayısı yerine alt-taşıyıcının band genişliği daha önemlidir. Band ne kadar geniş olursa bozulma sonrasında diklik o kadar az etkilenir ve demodülasyon sonucunda sadece o taşıyıcının frekansından bir değer elde edilir, diğerleri sıfır değerini verir. Diklik kayboldukça, başka bir değişle alt-taşıyıcı diğer alt-taşıyıcılara etki etmeğe başlayınca, demodülasyon sonucunda diğer taşıyıcılara ilişkin sonuçlar sıfır olarak alınamaz.

BW algoritması ile kestirilen kanal parametreleri kullanılarak kod çözme işlemi yapıldığında sonucun oldukça iyi olduğu görülmektedir. Fakat bir önceki bölümde söylendiği üzere BW kanal kestirim yöntemi karmaşık ve zaman gerektiren bir yöntemdir. *Pilot sembol* yöntemine göre avantajı ise herhangi bir ek bilgiye ihtiyaç duymaması ve bundan dolayı iletişim hızında bir yavaşlamaya neden olmamasıdır. Bunun tersine *Pilot sembol* eklenmesi durumunda hiç bir bilgi içermeyen semboller gönderilerek boşa zaman geçmekte ve iletişim süresinde bir yavaşlamaya neden olmaktadır.

3.4. Turbo Kafes Kodlamalı Sistemlerin Uzay Zaman Kodları ile Başarımı

Günümüzde özellikle tek bir noktadan diğer nokatalara iletim yapıldığı durumlarda çok anten kullanımının oldukça yararları vardır. [54-56]'da Uzay Zaman Kafes Kodlaması için kodlama, modülasyon, iletimdeki çokluluk ve optimal alıcının ortak dizaynı sunulmuştur. Uzay Zaman Blok Kodları ilk olarak [57]'de Bauch tarafından TTCM ile birleştirilmiş (ST-TTCM) ve çoklu anten sistemine uygun kanal kodlaması ile önemli yararların elde edildiği gösterilmiştir. Uzay Zaman kodlarının Rician, Rayleigh ve Nakagami dağılımlı, faz bozulmasının sözkonusu olduğu kanallar için sistem performans analizleri [58,59]'da yapılmıştır.

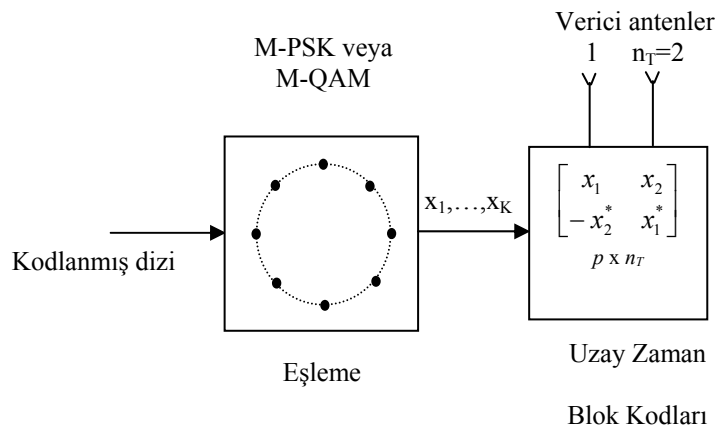
Uzay zaman kodları çoklu-yol (*multi-path*) etkisinin neden olduğu Rician ve özellikle Rayleigh sönümlmeli kanalların etkisini azaltmak için kullanılan çok etkin bir yöntemdir. Uzay Zaman Blok Kodları kullanılarak sönümleme etkisini azaltmaktan başka Doppler kaymasından doğan faz bozulması da tamamen ortadan kaldırılır.

Bu bölümde öncelikle Alamouti'nin önerdiği [54] Uzay Zaman Blok kodları (Space Time Block Codes, STBC) anlatılacak daha sonra da çok antenli sistemlerde yeni bir yöntem olarak önerilen Tekrarlamalı Uzay Zaman Kodları (Repeating Space Time Codes, RSTC) verilecektir.

3.4.1. Uzay Zaman Blok Kodları

Kablosuz haberleşmede çoklu-yol sönümlemesi şiddetli bir genlik ve faz bozulmasına neden olur. Bundan dolayı hem baz istasyonlarında hem de uzaktaki istasyonlarda bu sönümleme ile band genişliğinde ve sinyal gücünde bir değişiklik yapmadan mücadele etmek çok önemlidir. Bazı sistemlerde alıcı ile verici arasında kanal bilgilerini ileten bir hat vardır, fakat bu sistemin karmaşıklığını arttırdığı için pek önerilen bir yöntem değildir. Bu yüzden Uzay Zaman Blok kodları çok pratik ve sinyal enerjisi bakımından da verimli bir yöntemdir.

Şekil 3.46'da iki verici anteni olduğu durum için Uzay Zaman Blok kodlayıcısı gösterilmektedir. Uzay Zaman Blok kodlayıcısı n_T verici anten sayısı olmak üzere $p \times n_T$ lik $\mathbf{G}$ matrisine giren sembolleri eşler.



Şekil 3.46 Uzay Zaman Blok Kodlayıcısı

$\mathbf{G}$ matrisinin girişi karmaşık x_i , nin karmaşık eşleniği x_i^* ve bunların lineer kombinasyonlarıdır. $\mathbf{G}$ [54]'de tanımlandığı gibi genelleştirilmiş karmaşık dik bir matristir. Örnek olarak $n_T = 2$ için genelleştirilmiş karmaşık dik dizayn

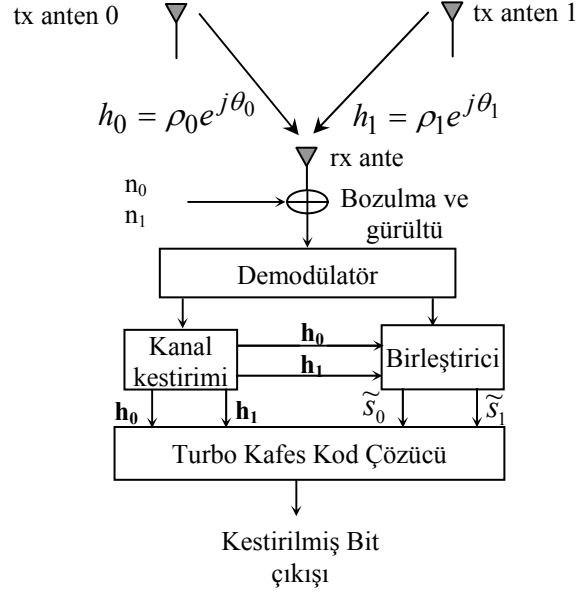
$$G_1 = \begin{bmatrix} g_{11} & \cdots & g_{1n_T} \\ \vdots & & \vdots \\ g_{p1} & \cdots & g_{pn_T} \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \\ -x_2^* & x_1^* \end{bmatrix} \quad (3.25)$$

şeklindedir. Burada Şekil 3.47'de gösterilen 2 vericili bir alıcılı basit bir anten yapısı göz önünde bulundurulacaktır. Her iki anten tx_o, tx_1 den gelen sinyalin aynı anda alındığı varsayılmaktadır. İlk kodlamam adımında birinci anten tx_o dan gönderilen sinyal, $x_1 = s_o$ olarak gösterilmekte, ikinci anten tx_1 den gönderilen sinyal de $x_2 = s_1$ olarak gösterilmektedir. İkinci kodlama adımında tx_o anteni $(-s_1^*)$ 'i tx_1 anteni de s_o^* 'ı göndermektedir. Bu iki kodlama adımı boyunca sönümleme ve faz bozulma etkisinin sabit kaldığı düşünülürse; tx_o anteni ile alıcı arasındaki kanal parametresi h_o ile, tx_1 ile alıcı arasındaki kanal parametresi de h_1 ile gösterilir. h_o ve h_1 çoklu-yol kanalları aşağıdaki gibi modellenir.

$$h_o(t) = h_o(t + T) = \rho_o e^{j\theta_o} \quad (3.26a)$$

$$h_1(t) = h_1(t + T) = \rho_1 e^{j\theta_1} \quad (3.26b)$$

Burada T sembol süresini, ρ sönümlleme katsayısını ve $e^{j\theta}$ birim çemberindeki θ da faz bozulmasını göstermektedir.



Şekil 3.47 ST-TTCM alıcı tarafı

Bu durumda alıcıdaki sinyal şu şekilde gösterilebilir,

$$r_0 = r(t) = h_0 s_0 + h_1 s_1 + n_0 \quad (3.27a)$$

$$r_1 = r(t+T) = -h_0 s_1^* + h_1 s_0^* + n_1 \quad (3.27b)$$

r_0 ve r_1 t ve $t+T$ anlarında alınan sinyallerdir. n_0 ve n_1 de bu zaman dilimlerine ait Gauss gürültüleridir. Şekil 3.47'deki birleştirici (combiner) yapısı,

$$\tilde{s}_0 = h_0^* r_0 + h_1 r_1^* \quad (3.28a)$$

$$\tilde{s}_1 = h_1^* r_0 - h_0 r_1^* \quad (3.29b)$$

Denklem (3.26) ve (3.28) kullanılarak,

$$\tilde{s}_0 = (\rho_0^2 + \rho_1^2) s_0 + h_0^* n_0 + h_1 n_1^* \quad (3.30a)$$

$$\tilde{s}_1 = (\rho_0^2 + \rho_1^2) s_1 - h_0 n_1^* + h_1^* n_0 \quad (3.30b)$$

elde edilir. (3.29) denkleminin çıktısı Turbo Kafes kod çözücünün girişi olarak kullanılır. (3.29) denklemi iyice incelendiğinde, Uzak Zaman Blok Kodları ile gönderilen s_0 ve s_1 sinyallerinin faz bozulmasından hiç bir şekilde etkilenmediği görülmektedir. Çünkü h_0 ve h_1 terimleri gürültü ile çarpılmaktadır. Bunların içinde bulunan faz bozulma etkisi $e^{j\theta}$ sadece gürültünün fazını değiştirmekte gücünde herhangi bir değişiklik yapmamaktadır. Sinyallerin başındaki $(\rho_0^2 + \rho_1^2)$ terimi de yollardan herhangi birisinin kötü olması ile sönümlenme parametresinin çok düşmesi durumunda diğer yoldan alınan bilginin sinyali biraz iyileştirdiğine işaret etmektedir.

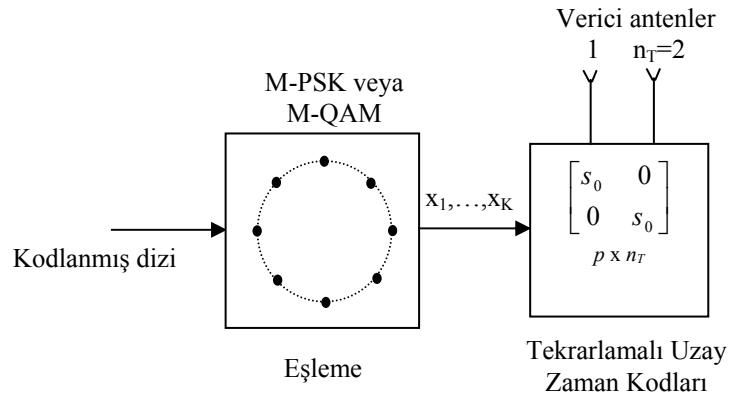
3.4.2. Tekrarlamalı Uzak Zaman Kodları

Tekrarlamalı Uzak Zaman kodlarının (Repeating Space Time Codes, RSTC) oluşturulması her antenden sırasıyla aynı sinyalin gönderilmesi temeline dayanır. Her

anda sadece bir anten sinyal gönderir, aynı sinyal teker teker tüm antenlerden gönderildikten sonra bir sonraki sinyal teker teker gönderilmeye başlanır. Her an tek bir sinyal gönderildiği için sinyal gücünde bir değişiklik yapılmamış olur. Tekrarlamalı Uzay Zaman kodlarını oluşturan $\mathbf{G}$ matrisi aşağıdaki gibi gösterilebilir.

$$\mathbf{G} = \begin{bmatrix} s_0 & 0 & \cdots & 0 \\ 0 & s_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & s_0 \end{bmatrix} \quad (3.31)$$

Şekil 3.48’de iki anten için kod oluşumu gösterilmektedir.



Şekil 3.48 Tekrarlamalı Uzay Zaman kodlayıcısı

Alınan sinyal şu şekilde gösterilebilir,

$$r_0 = r(t) = h_0 s_0 + n_0 \quad (3.32a)$$

$$r_1 = r(t+T) = h_1 s_0 + n_1 \quad (3.32b)$$

r_0 ve r_1 , t ve $t+T$ anlarında alınan sinyallerdir. n_0 ve n_1 de bu zaman dilimlerine ait Gauss gürültüleridir. Gönderilen sinyal s_0 'ı tekrar elde etmek için birleştiricide aşağıdaki işlem yapılır.

$$\tilde{s}_0 = h_0^* r_0 + h_1^* r_1 \quad (3.33a)$$

Çeşitli matematiksel işlemler yapılarak birleştirici çıktısı,

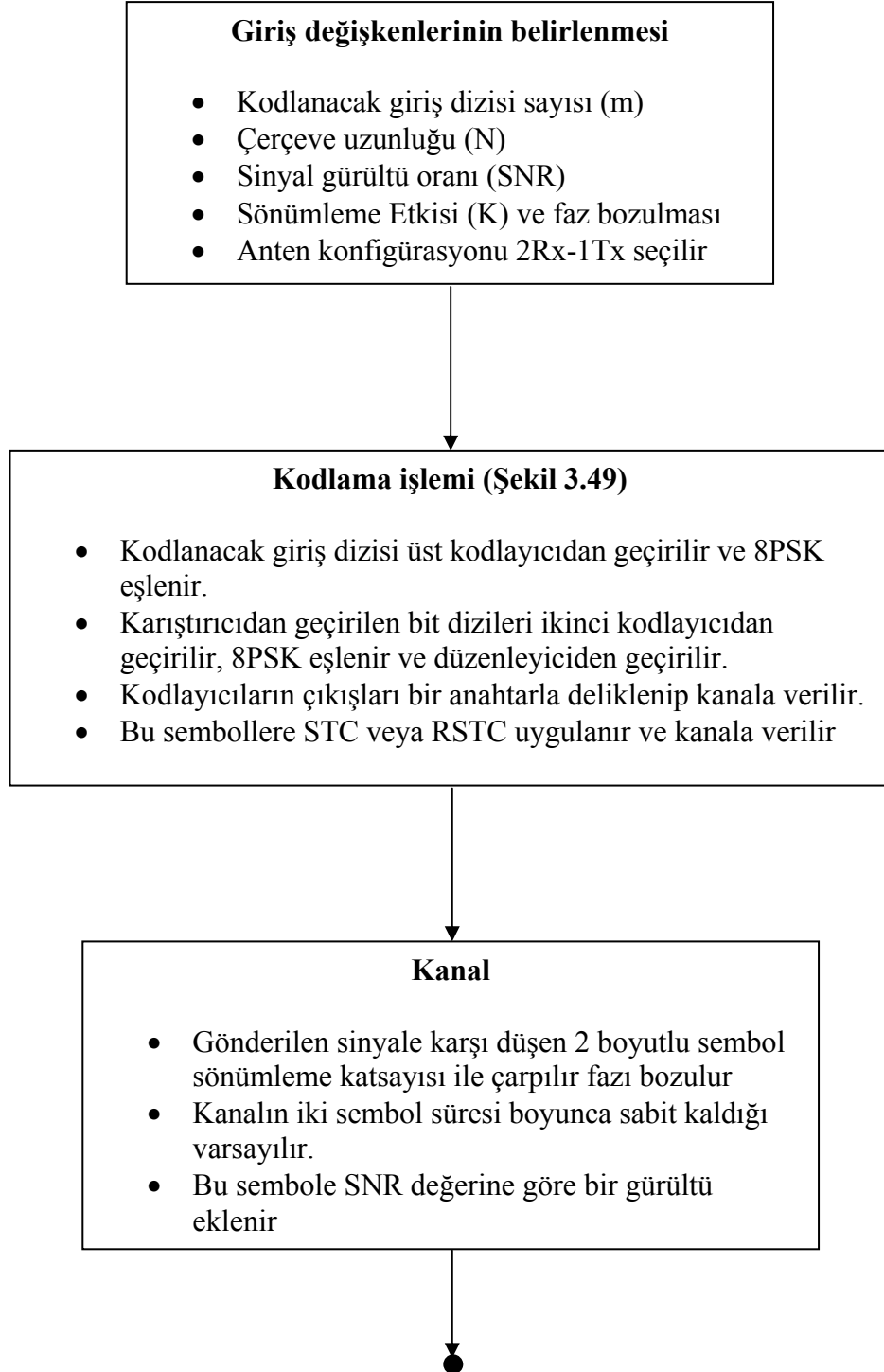
$$\tilde{s}_0 = (\rho_0^2 + \rho_1^2) s_0 + h_0^* n_0 + h_1^* n_1 \quad (3.33b)$$

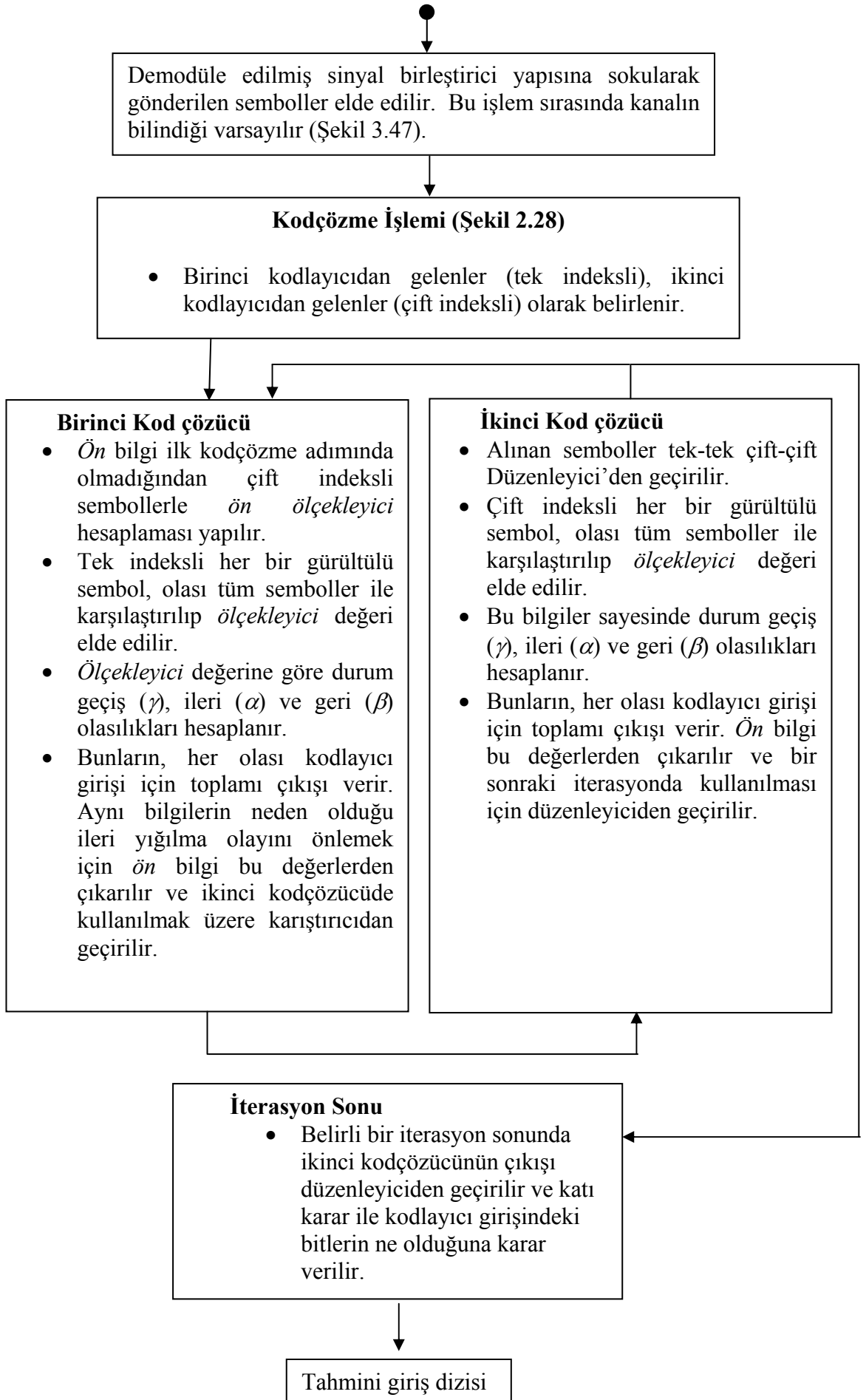
olarak elde edilir.

Uzay Zaman Blok kodlarında ve Tekrarlamalı Uzay Zaman kodlarında birleştirici çıkışının doğru olarak elde edilebilmesi için kanal parametrelerinin tam olarak kestirilmesi gerekmektedir. Bunun için ML ya da BW algoritmalarının kullanılması uygun olur. Burada her iki Uzay Zaman kodunun TTCM üzerindeki başarımını elde etmek için kanal parametrelerinin tam olarak bilindiği varsayılmaktadır.

Simülasyonda sistemin verici, kanal ve alıcı blok diyagramlarının nasıl gerçekleştirildiğini ve hangi değişkenlerin nasıl ve nerede belirlendiğini göstermek amacı ile aşağıdaki gibi bir akış diyagramı verilmiştir.

Tablo 3.5 Uzay Zaman kodlar ile bütünleşik Turbo kodlama akış şeması



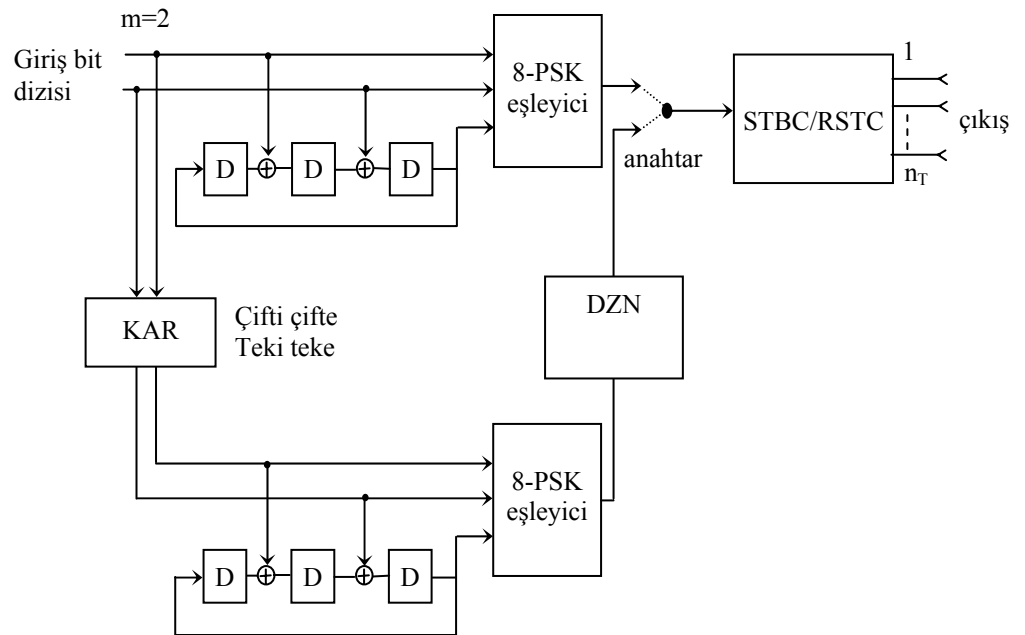


3.4.3. Uzay Zaman Blok Kodları ve Tekrarlamalı Uzay Zaman Kodlar için TTCM

Başarımı

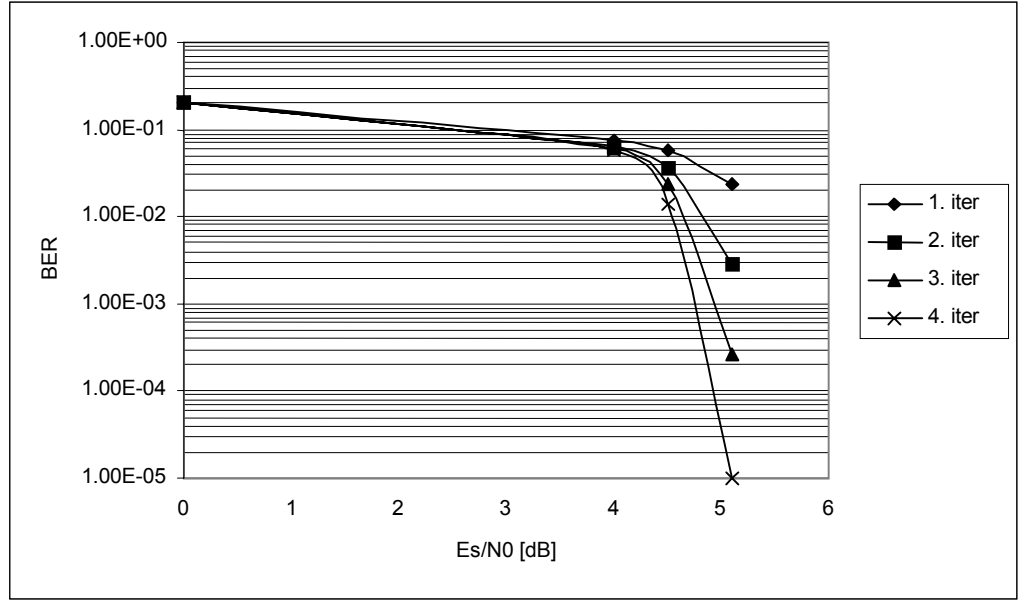
Burada Şekil 3.49'daki kodlayıcı modeli kullanılacaktır. Turbo kafes kodlayıcının her birinde 3 adet bellek bulunmakta ve iki adet 1024 bit data dizisi giriş olarak kullanılmaktadır. Modülasyonun 8-PSK, anten konfigürasyonunun da 2 verici bir alıcı şeklinde olduğu düşünülmüştür. Turbo kafes kod çözücüsünde giriş data dizisinin kestirimi için 4 iterasyon yapılacaktır. Her iki Uzay Zaman kodlaması için sonuçlar AWGN, Rician ($K=10\text{dB}$) ve Rayleigh kanalları için elde edilecektir.

Simülasyon MATLAB programı kullanılarak hazırlanmıştır. PIV 1.5 GHz bir bilgisayarda 4 iterasyonda, başlangıçta 256 bit uzunluklu bir çerçeve için 5.4 sn ve 1024 bit uzunluklu bir çerçeve için 15 sn sürmektedir.

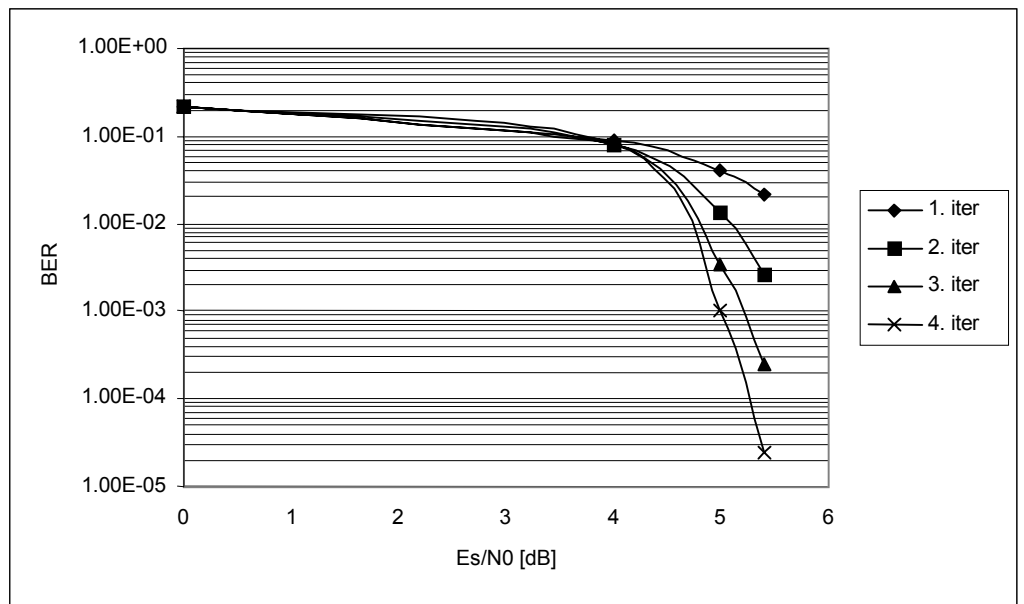


Şekil 3.49 STBC/RSTC ile TTCM kodlayıcı yapısı

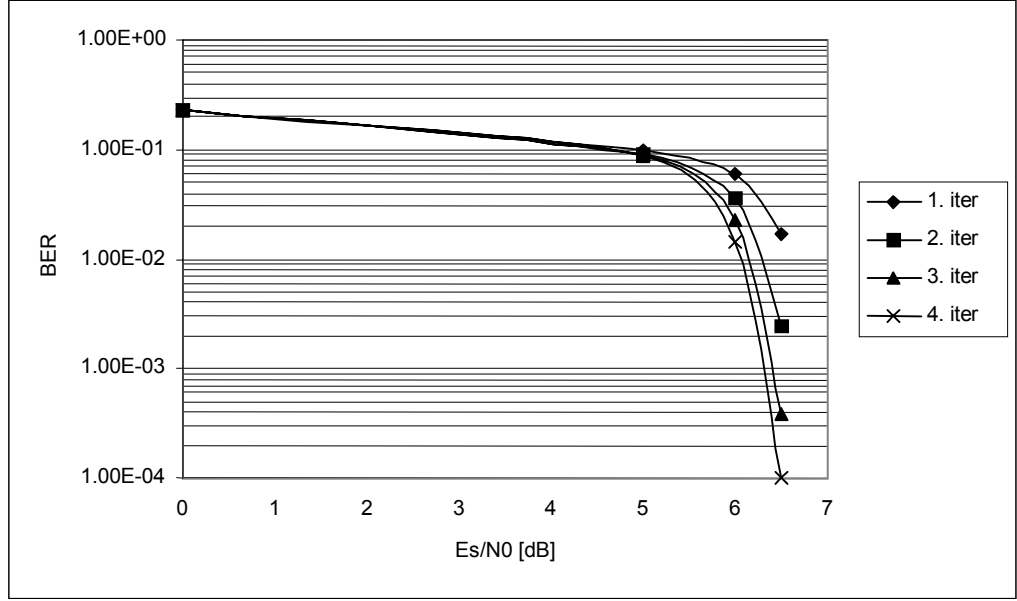
Şekil 3.50 3.51 ve 3.52’de anten konfigürasyonunun 2 vericili 1 alıcılı olduğu durumda AWGN, Rician ve Rayleigh kanallarda TTCM’in Uzay Zaman Blok kodlar ile 4 iterasyon için başarımı gösterilmektedir.



Şekil 3.50 ST-TTCM’in AWGN ortamındaki başarımı

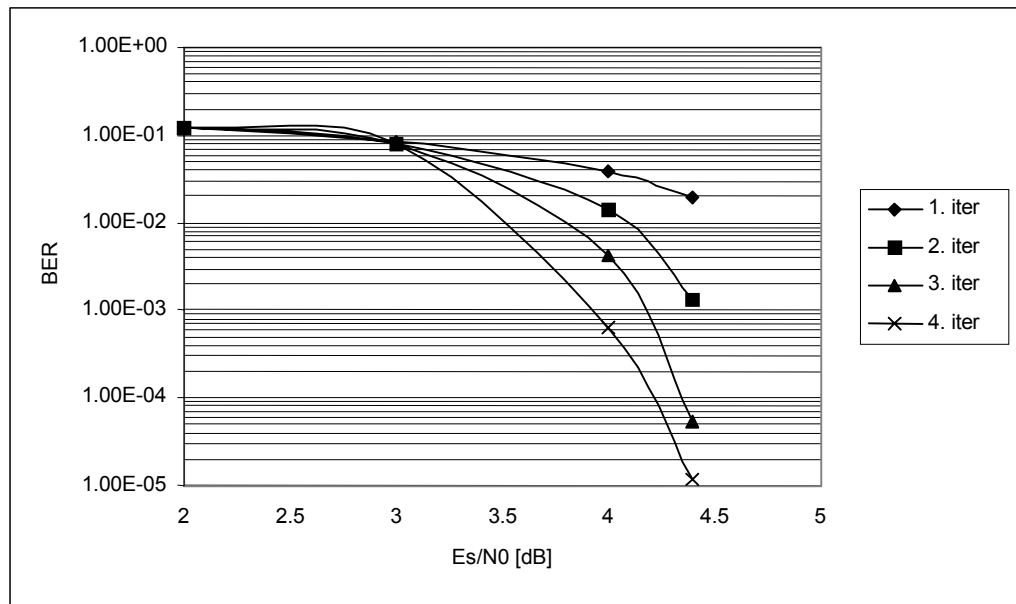


Şekil 3.51 ST-TTCM’in Rician K=10dB ortamındaki başarımı

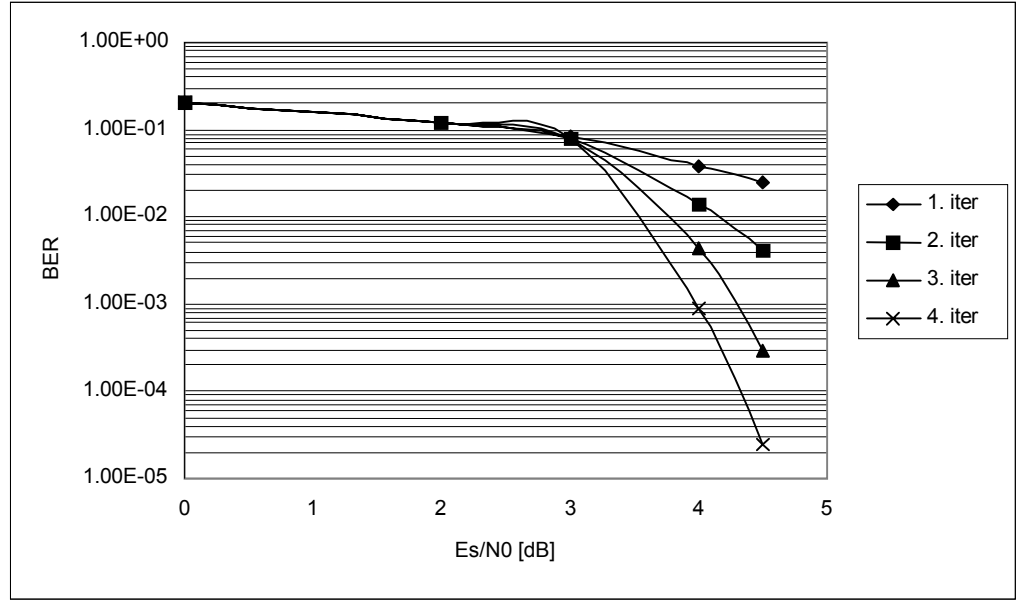


Şekil 3.52 ST-TTCM'in Rayleigh ortamındaki başarımı

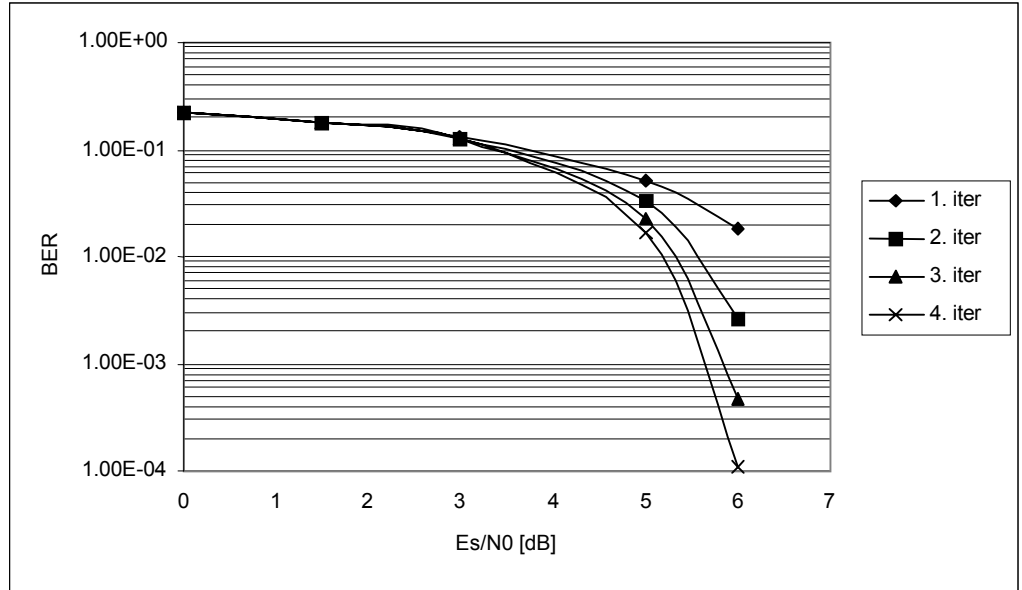
Şekil 3.53 3.54 ve 3.55'de anten konfigürasyonunun 2 vericili 1 alıcılı olduğu durumda AWGN, Rician ve Rayleigh kanallarda TTCM'in Tekrarlamalı Uzay Zaman kodlar ile 4 iterasyon için başarımı gösterilmektedir.



Şekil 3.53 RST-TTCM'in AWGN ortamındaki başarımı



Şekil 3.54 RST-TTCM'in Rician K=10dB ortamındaki başarımı



Şekil 3.55 ST-TTCM'in Rayleigh ortamındaki başarımı

Uzay Zaman kodlaması genel olarak kanal sayısını arttırarak gönderilen sinyallerde tüm kanlların ortalama etkisini sinyale yansıtıp herhangi bir kanaldaki yüksek sönümlleme etkisini diğer kanallarla kompanze etmeyi amaçlar. STBC için (3.30) ve RSTC için (3.33 b) denklemlerinde, sembol çarpanı olan kanalların sönümlleme katsayısı ρ 'ların karelerinin toplamının değişiminin az olabilmesi için kanal sayısının çok olması gereği açıkça görülmektedir. Bu ancak alıcı ve vericideki anten sayılarının arttırılması ile sağlanabilir. Sonuç olarak anten sayısının artması, gönderilen sembolü etkileyen kanal sönümlleme katsayısının ortalama bir değere çekilmesine, bu da hata başarımının iyileşmesine olanak tanır.

Her iki Uzay Zaman kodlama simülasyon sonuçlarını karşılaştırdığımızda Tekrarlamalı Uzay Zaman kodlarının 0.6 dB daha iyi olduğu anlaşılmaktadır. Bu avantaja karşılık her sinyalin tüm antenlerden teker teker gönderilmesi bit transfer hızını anten sayısı oranında azaltmaktadır.

3.5. Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama

Band verimliliğini arttırmak için genellikle Sürekli Faz Modülasyonu (Continuous Phase Modulation, CPM) kullanılan yaklaşımların uydu haberleşmesine uygulanması iyi sonuçlar vermektedir. Aynı zamanda CPM'in sabit bir zarfa sahip olması, daha düşük fiyatlı ve güç verimliliği lineer olanlardan daha yüksek olan lineer olmayan yükselticilerin kullanılmasına imkan tanır. CPM bloğu iki parçadan oluşur; birincisi Sürekli Faz Kodlayıcısı (Continuous Phase Encoder, CPE), diğeri de Belleksiz Eşleyici (Memoryless Mapper, MM) [60]. CPM sürekli fazlı sinyaller yaratan, kod kelime dizilerini MM ile dalga şekline sokan katlamalı bir kodlayıcıdır. CPM'in bellek yapısı belirlendikten sonra, kafes yapıyı ve CPE'yi tek bir katlamalı kod olarak birleştirmek mümkündür [61-63]. CPE li sistemlerin Öklid uzaklığı geleneksel kafes yapılı sistemlere göre daha fazla olduğu için performans da artar.

Sönümlemeli kanalların yarattığı korelasyon etkisini azaltmak için karıştırıcı, kafes yapı ile CPE arasına konulabilir [64,65]. Aksi takdirde karıştırıcıyı modülatörden sonra koymak, kanala verilecek sembollerin sırasının değişmesine neden olacağından fazın sürekliliğini ortadan kaldıracaktır. [66-68]'de kafes yapı ile CPE arka arkaya koyularak iteratif kod çözme algoritmasını uygulandığı gösterilmektedir. [69]'de ise CPM yapısı ikili Turbo kodlara uygulanmıştır.

Sürekli Faz Frekans Kaydırmalı Anahtarlama (Continuous Phase Frequency Shift Keying, CPFSK) adı verilen lineer faz dalgalı tam cevaplı CPM, katlamalı kodlarla birleştirildiğinde 5-6 dB kazanç sağladığı [70]'de belirtilmiştir. Aynı zamanda bu kazanç, daha fazla bellek kullanarak ve CPFSK'nın boyutunu büyüterek arttırılabilir. Bu kazanç band genişliğinde herhangi bir artma olmadan elde edilmiştir.

Bu bölümde katlamalı kodlayıcının bellek yapısı değiştirilmeden ve CPE için ekstra bellek kullanılmadan, CPE için gerekli olan bellekleri katlamalı kodlayıcının bazı belleklerinden kullanılabileceği gösterilecektir. İkinci ve daha önemlisi ise kodlayıcının çıkışında bir anahtar (*puncturer*) olmasına rağmen sürekliliğin nasıl bozulmayacağı açıklanacaktır.

3.5.1. Sürekli Faz Modülasyonu

Bu bölümde, sürekli faz modülasyonu ve onun özel bir biçimi olan sürekli fazlı frekans kaydırmalı anahtarlama ile ilgili bilgiler verilmektedir. Ayrıca, sürekli fazlı frekans kaydırmalı anahtarlamanın eğik fazlı biçimi ayrıntılı olarak tanıtılmaktadır.

Sabit zarflı CPM işaretleri

$$s(t, \alpha) = (2E_s / T)^{1/2} \cos(2\pi f_c t + \Phi(t, \alpha) + \Phi_0), \quad t \geq 0 \quad (3.34)$$

olarak gösterilebilir. Burada $E = R_s E_b$ simge enerjisini (E_b : Bilgi biti enerjisi), T modülasyon aralığını, f_c taşıyıcı frekansını ve Φ_0 başlangıç fazını göstermektedir. Genelliği bozmaksızın $\Phi_0 = 0$ alınabilir. Bilgi taşıyan faz terimi

$$\Phi(t, \alpha) = 4\pi h \sum_{i=0}^{\infty} \alpha_i q(t - iT), \quad t \geq 0 \quad (3.35)$$

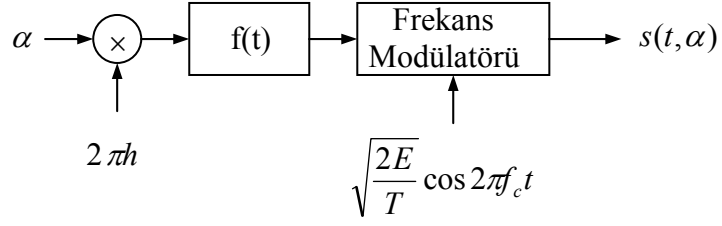
şeklinde dir. h modülasyon indisini göstermektedir ve pratik amaçlar için, J ve P ortak bölenleri olmayan iki tamsayı olmak üzere,

$$h = \frac{J}{P} \quad (3.36)$$

biçiminde seçilir. Aksi halde CPM sisteminin durum sayısı ve dolayısıyla alıcının karmaşıklığı sonsuz olmaktadır. M , 2'nin kuvveti olmak üzere M düzeyli bir CPM için α_k bağımsız bilgi simgeleri, $\mp 1, \mp 3, \dots, \mp (M-1)$ değerlerini $1/M$ eş olasılığı ile alırlar. Örneğin ikili ($M=2$) CPM için $\alpha_k = \mp 1$ 'dir. $q(t)$ faz yanıtı olarak adlandırılır ve

$$q(t) = \int_{-\infty}^t f(\tau) d\tau \quad (3.37)$$

ile ifade edilir. $f(t)$ ani frekans darbe yanıtı işlevidir. N darbe uzunluğunu göstermek üzere, $f(t)$ işlevi $0 \leq t \leq NT$ süresince sıfırdan farklı, bu aralık dışında sıfır seçilir. Sürekli faz modülatörünü simgeleyen bir model, (3.34) eşitliğinden yararlanarak Şekil 3.56 daki gibi elde edilebilir.



Şekil 3.56 Sürekli faz modölatörü

$f(t)$ 'nin seçimi faz teriminin düzlüğünü/biçimini belirler. Herhangi bir simge aralığında, α_k simgesinin neden olduğu en büyük faz değişiminin $\pi h \alpha_k$ biçiminde olabilmesi için,

$f(t)$ işlevi $\int_{-\infty}^{\infty} f(t) dt = 1/2$ olacak şekilde seçilir. Bu koşul altında $f(t)$ 'nin biçimi, h ve M

parametreleri değiştirilerek birçok CPM ifadesi elde edilebilir. Örneğin $f(t)$, N uzunluklu (NT süreli) yükseltilmiş cosinüs (N Raised Cosine, NRC) veya dikdörtgen darbe (N Rectangle, NREC) seçilebilir. Bu durumlar için $f(t)$ sırasıyla,

$$f(t) = \begin{cases} \frac{1}{2NT} (1 - \cos \frac{2\pi t}{NT}), & 0 \leq t \leq NT \\ 0, & \text{diger} \end{cases} \quad (3.38)$$

ve

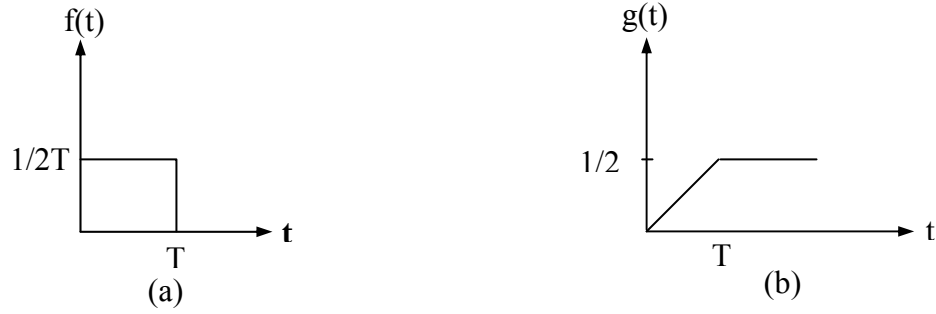
$$f(t) = \begin{cases} \frac{1}{2NT}, & 0 \leq t \leq NT \\ 0, & \text{diger} \end{cases} \quad (3.39)$$

biçimindedir. Sürekli faz modülasyonu, $N=1$ için “tam yanıtı (full response) CPM” $N > 1$ için “kısmi yanıtı (partial response) CPM” [71] olarak adlandırılır. Çoğu durumda kısmi yanıtı CPM sistemleri, tam yanıtı CPM sistemlerinden daha iyi spektral özellik gösterirler. Buna karşın, kısmi yanıtı CPM sistemlerinin özellikle alıcı kısmındaki donanım karmaşıklığı çok artmaktadır. Bu nedenle ve tasarım/analiz kolaylığından dolayı sürekli faz modülasyonu konusundaki araştırmaların büyük çoğunluğu tam yanıtı CPM üzerine yapılmaktadır.

3.5.2. Sürekli Faz Frekans Kaydırmalı Anahtarlama

Sürekli faz modülasyonuna ilişkin ani frekans darbe yanıtı işlevi $f(t)$, Şekil 3.57 (a)'da gösterildiği gibi 1REC, yani T süreli bir dikdörtgen darbedir. Dolayısıyla faz yanıtının Şekil 3.57 (b)'deki gibi olduğu durum, sürekli faz frekans kaydırmalı anahtarlama (CPFSK) olarak adlandırılır. CPFSK, bilinen frekans kaydırmalı anahtarlama (Frequency Shift Keying, FSK) modülasyonunun sürekli fazlı biçimidir. Ancak, faz sürekliliği özelliğinden dolayı daha dar bir frekans spektrumu sağlamanın yanında, modülatöründe tek bir osilatör gerektirmesi, CPFSK için FSK'ya göre önemli üstünlükler getirmektedir. Gerçekten de, FSK işaretlerini elde edebilmek için toplam işaret sayısı kadar osilatör gereklidir ve bilgi bit dizisine göre frekans değiştirilirken bir osilatörden diğer osilatöre anahtarlama yapılmalıdır. Uygulamada bu anahtarlama işleminin ideal olarak gerçekleştirilememesi faz atlamalarını arttıracak ve FSK işaretine ilişkin frekans spektrumunun daha da yayılmasına neden olacaktır. CPFSK modülasyonunda ise tek bir osilatör olduğu için, hem osilatör sayısı minimumda

tutulmuş olmakta, hem de faz atlamaları önlenmiş olmaktadır. Böylece, FSK'ya kıyasla daha dar bir iletim band genişliği elde edilebilmektedir.



Şekil 3.57 (a) CPFSK'ya ait $f(t)$. (b) $f(t)$ 'nin integrali $g(t)$.

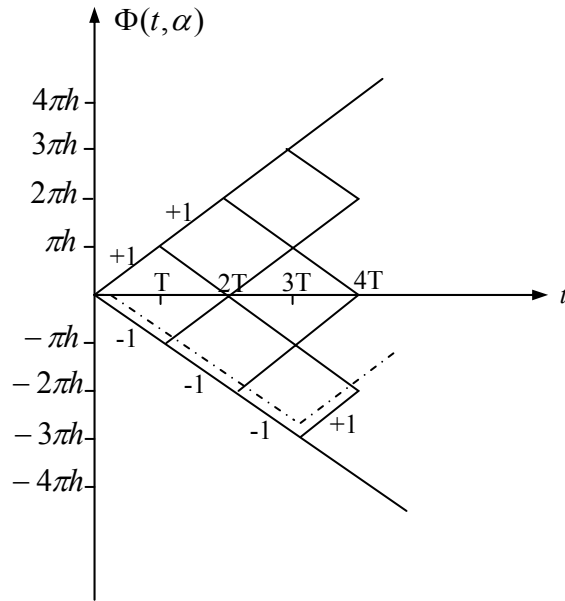
$nt \leq t \leq (n+1)T$ zaman aralığında CPFSK işaretleri

$$\begin{aligned}
 s(t, \alpha) &= \sqrt{\frac{2E}{T}} \cos \left[2\pi f_c t + 2\pi h \sum_{k=-\infty}^{\infty} \alpha_k \frac{(t - kT)}{2T} \right] \\
 &= \sqrt{\frac{2E}{T}} \cos \left[2\pi f_c t + \pi h \alpha_n \frac{(t - nT)}{T} + \varphi_n \right]
 \end{aligned} \tag{3.40}$$

olarak yazılabilir. Burada φ_n ,

$$\varphi_n = \left[\pi h \sum_{k=-\infty}^{n-1} \alpha_k \right]_{\text{mod } 2\pi} \tag{3.41}$$

eşitliği ile ifade edilen ve $(n-1)T$ anına kadar toplamsal olarak oluşan faz değerlerini gösteren bellek terimidir. Gösterilebilir ki, eğer J çift ise sinyalin başlangıç faz açısı $\varphi_n \in \{0, \pi J / P, 2\pi J / P, \dots, (P-1)\pi J / P\}$ 'dir. Genel olarak, CPFSK işaretlerinin AM, PSK ve QAM modülasyonlarında olduğu gibi işaret uzayında gösterilimi mümkün olmamaktadır. Bunun sebebi ise, CPFSK'ya ilişkin işaret uzayı 3'ten fazla boyut içerebilmektedir. Bu nedenle CPFSK işaretlerinin simgelenmesinde bir alternatif olarak (3.35) eşitliğindeki faz terimi $\Phi(t, \alpha)$, zamanın bir işlevi olarak "faz ağacı" yardımıyla gösterilmektedir. Örneğin, başlangıç fazı ve zamanı sıfır alınarak ikili CPFSK'ya ilişkin faz ağacı Şekil 3.58'deki gibi verilebilir.



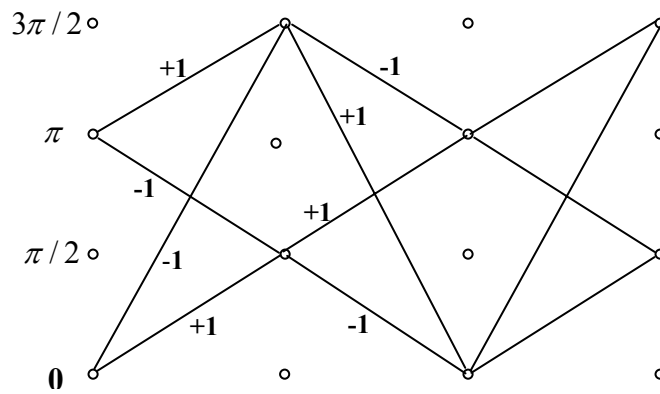
Şekil 3.58 İkili CPFSK faz ağacı

Görüldüğü gibi, her T modülasyon aralığında $\alpha_n = \mp 1$ değerine göre düzgün bir faz değişimi vardır. Ayrıca şekil üzerinde örnek olarak verilen $\alpha = \{-1, -1, -1, +1\}$ bilgi dizisine ilişkin faz değişim yolu kesikli çizgilerle gösterilmiştir.

Zamanla sürekli büyüyen faz ağacı yerine, taşıyıcı fazının 0 ile 2π aralığında değer alabileceği gerçeğinden hareketle,

$$\Phi(t, \alpha) = [\Phi(t, \alpha)]_{\text{mod } 2\pi} \quad (3.42)$$

tanımı yapılmaktadır. Bu durumda CPFSK modülasyonunun fiziksel faz kafesi çizilebilir. Örneğin, CPFSK modülasyonunda $M=2$ ve $h=1/2$ seçilmiş durum için elde edilen modülasyonunun faz kafesi Şekil 3.59'daki gibidir.



Şekil 3.59 $M=2$, $h=1/2$ için faz kafes diyagramı.

Şekilden görüldüğü gibi, bu modülasyonunun kafes yapısı zamanla değişen türdendir. Kafes yapısının zamanla değişen türden olması, sürekli faz modülasyonunun bir kodlayıcı yardımıyla kodlanması durumundaki analizleri (örneğin hata başarımlar analizi) güçleştirmektedir. Bu nedenle, kodlama tekniği açısından CPFSK modülasyonunun zamanla değişmeyen bir kafes diyagramına sahip olması daha uygun olmaktadır. CPFSK modülasyonunun bu özelliği sağlayan bir türü Rimoldi [60] tarafından önerilmiştir ve “eğik fazlı CPFSK modülasyonu” olarak adlandırılmaktadır.

Bu amaçla (3.40) ifadesi,

$$s(t, \alpha) = \sqrt{\frac{2E}{T}} \cos[2\pi f_1 t + \Psi(t, \alpha)] , \quad t \geq 0 \quad (3.43)$$

biçiminde yeniden yazılsın. Burada $f_1 = f_c - h(M-1)/2T$ değiştirilmiş taşıyıcı frekansı ve $\Psi(t, \alpha)$,

$$\Psi(t, \alpha) = \Phi(t, \alpha) + \frac{\pi h(M-1)t}{T} \quad (3.44)$$

ile verilen eğik faz terimidir. Faz terimi 0 ile 2π aralığında değer alabileceğine göre ,

$$\Psi(t, \alpha) = [\Psi(t, \alpha)]_{\text{mod } 2\pi} \quad (3.45)$$

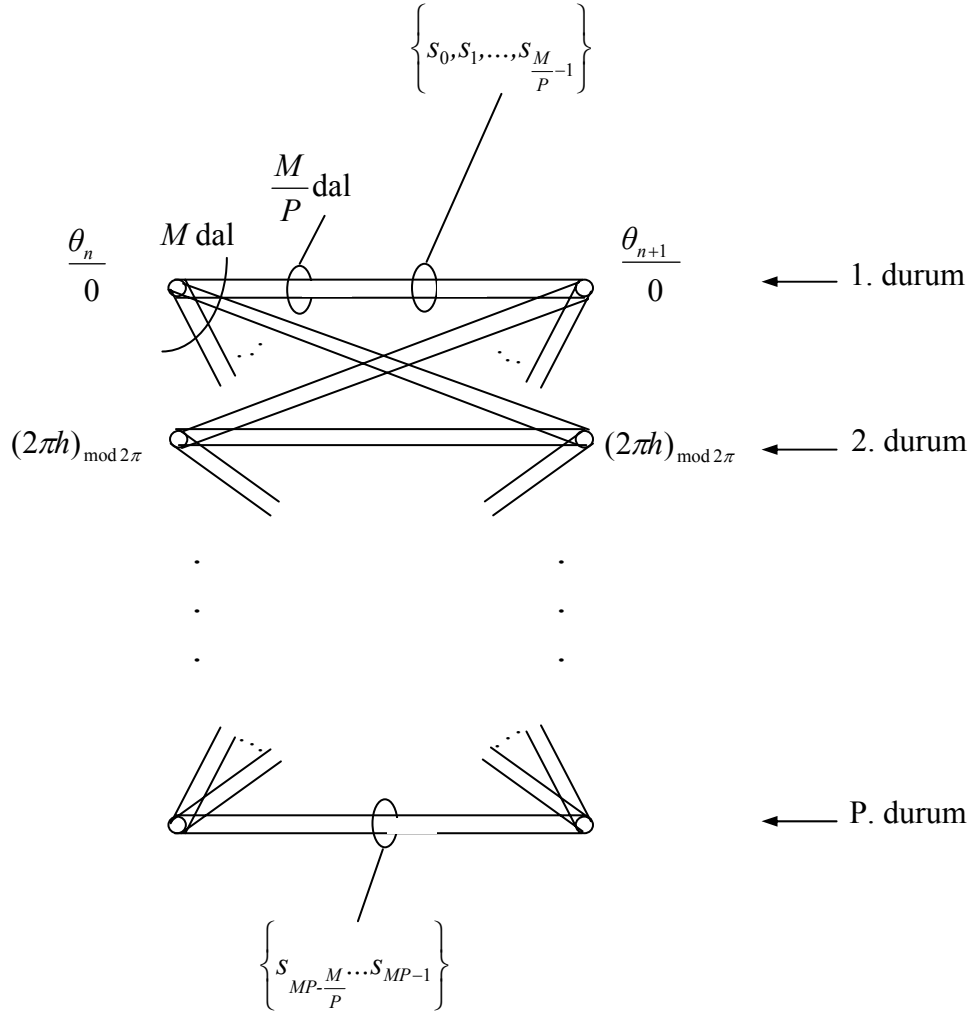
şeklinde yazılabilir. (3.34) ifadesi yeniden düzenlenirse,

$$s(t, \alpha) = \sqrt{\frac{2E}{T}} \cos \left[2\pi f_1 t + 2\pi h \beta_n \frac{(t - nT)}{T} + \theta_n \right] \quad (3.46)$$

yazılabilir. (3.46) ifadesi, eğik fazlı CPFSK modülasyonuna ilişkin genel işaret biçimini göstermektedir. θ_n , $t = nT$ anında fazın başlangıç değeri, yani modülasyonun faz sürekliliğini sağlayan bellek terimidir. Ayrıca,

$$\theta_{n+1} = (\theta_n + 2\pi h \beta_n)_{\text{mod } 2\pi} \quad (3.47)$$

olduğu kolaylıkla gösterilebilir. θ_{n+1} ile θ_n aynı eleman kümesinden değer almak zorundadır. Dolayısıyla $\theta_n \in \{0, (2\pi h)_{\text{mod } 2\pi}, (4\pi h)_{\text{mod } 2\pi}, \dots\}$ biçiminde P farklı değer alacaktır [72]. Buradan yararlanarak (3.46) ifadesi dikkatle incelenecek olursa, bu ifadenin θ_n faz değerlerini durum olarak alan ve zamanla değişmeyen sonlu ve P durumlu bir kafes diyagram ile simgelenebileceği görülecektir. Üstelik bu diyagramda, her biri β_n giriş değerine ve θ_n başlangıç fazına bağlı olarak elde edilecek $s(t, \beta)$ çıkış işareti de gösterilebilir.



Şekil 3.60 M 'li CPFSK, $h=J/P$ 'nin fiziksel eğik faz kafesi (işaret kümesi)

Elde edilen kafes diyagramı Şekil 3.60'daki gibidir. Kafes diyagramında her durumdan M adet dal çıkmaktadır. s_i , β_n ve θ_n değerlerine göre elde edilebilecek $M \times P$ adet $s(t, \beta)$ işaretinden i 'sini simgelemektedir ($i = 0, 1, \dots, MP - 1$). Tezin bu bölümünden sonra, M bilgi genlik düzeyine ve $h = J/P$ modülasyon indisine sahip eğik fazlı CPFSK, kısaca M 'li CPFSK, $h = J/P$ olarak gösterilecek ve Şekil 3.60'daki yapı M 'li CPFSK, $h = J/P$ işaret kümesi olarak adlandırılacaktır. İzleyen alt bölümlerde, bu tezde ele alınacak M 'li CPFSK, $h = J/P$ modülasyon türleri için $nT \leq t \leq (n+1)T$ zaman aralığında $s(t, \beta)$

işaret biçimi, bilgi genlik düzeyleri, $\Psi(t, \beta)$ değerleri, modülasyonlu işarete karşı düşen s_i simgeleri ile işaret kümeleri verilmektedir. Ayrıca, ilgili modülasyon türlerine ilişkin karesel serbest Öklid uzaklıkları ve R_s [bit/simge] değerleri de verilmektedir. Karesel serbest Öklid uzaklığının hesabında gerekli olan, M 'li CPFSK, $h=J/P$ için β ve β' bilgi simge dizileri ile belirlenen herhangi iki işaret arasındaki karesel Öklid uzaklığı

$$d^2(\beta, \beta') = \begin{cases} 2E \left[1 - \frac{\sin \Delta_{\Psi_{n+1}} - \sin \Delta_{\Psi_n}}{\Delta_{\Psi_{n+1}} - \Delta_{\Psi_n}} \right], & \Delta_{\Psi_{n+1}} \neq \Delta_{\Psi_n} \\ 2E [1 - \cos \Delta_{\Psi_n}] & \Delta_{\Psi_{n+1}} = \Delta_{\Psi_n} \end{cases} \quad (3.48)$$

olarak verilir [36]. Burada Δ_{Ψ_n} ve $\Delta_{\Psi_{n+1}}$,

$$\Delta_{\Psi_n} = \Psi(nT, \beta) - \Psi(nT, \beta') \quad (3.49)$$

$$\Delta_{\Psi_{n+1}} = \Psi((n+1)T, \beta) - \Psi((n+1)T, \beta') \quad (3.50)$$

ile ifade edilen, sırasıyla nT ve $(n+1)T$ anlarındaki faz farklarını göstermektedir. CPM modülasyonunda h modülasyon indisi ne kadar küçükse, işaretin band genişliğide o kadar dar olur. Bu nedenle, bu tezde modülasyon indisi ifadesinde $J=1$ alınmıştır.

3.5.2.1. 2'li CPFSK , h=1/2

Burada $h=1/2$ olduğundan iki ayrı faz değeri vardır, bu faz değerini saklamak için 1 bellek kullanmak gerekir. $M=2$ olduğundan toplam 4 sinyal oluşur.

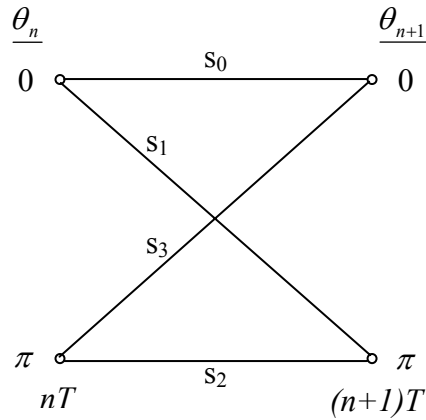
$$s(t, \underline{\beta}) = \sqrt{\frac{2E}{T}} \cos \left[2\pi f_1 t + \pi \beta_n \frac{t - nT}{T} + \theta_n \right] \quad (3.51)$$

$$\alpha_n \in \{-1, +1\} \Rightarrow \beta_n \in \{0, 1\}$$

$$\theta_n \in \{0, \pi\}$$

$$R_s = 1 \text{ bit/simge}$$

$$d_{serbest}^2 = 4E$$



Şekil 3.61 2'li CPFSK $h=1/2$ 'nin işaret kümesi.

Şekil 3.61'de görüldüğü gibi 4 sinyale ilişkin faz geçiş kafesi verilmektedir. Burada başlangıç açısı θ_n ve bitiş açısı da θ_{n+1} 'dir. Bu açı bilgisi bir bellek kullanılarak bir sonraki sinyale geçirilebilmekte, belleğin 0 ve 1 durumları açısal olarak 0 ve π değerlerini göstermektedir.

Tablo 3.6 2'li CPFSK, h=1/2 değerleri

θ_n	β_n	$\Psi(t, \underline{\beta})$	$\Psi(nT, \underline{\beta})$	$\Psi((n+1)T, \underline{\beta})$	s_i	θ_{n+1}
0	0	0	0	0	s_0	0
	1	$\pi \frac{(t-nT)}{T}$	0	π	s_1	π
π	0	π	π	π	$s_2 = -s_0$	π
	1	$\pi \frac{(t-nT)}{T} + \pi$	π	2π	$s_3 = -s_1$	0

Bu tabloda θ_n başlangıç açısını, β_n katlamalı kodlayıcı çıkışını, $\psi(t, \beta)$ frekansın ne kadar kaydırılacağını, $\psi(nT, \beta)$ bulunulan fazı, $\psi((n+1)T, \beta)$ $\psi(t, \beta)$ ve $\psi(nT, \beta)$ 'ya göre bitiş fazının ne olacağını, s_i faz açılarına ve Şekil 3.61'e göre oluşacak sinyali, θ_{n+1} de bitiş açısını göstermektedir.

3.5.2.2. 4'lü CPFSK , h=1/2

Burada $h=1/2$ olduğundan iki ayrı faz değeri vardır, bu faz değerini saklamak için 1 bellek kullanmak gerekir. $M=4$ olduğundan toplam 8 sinyal oluşur.

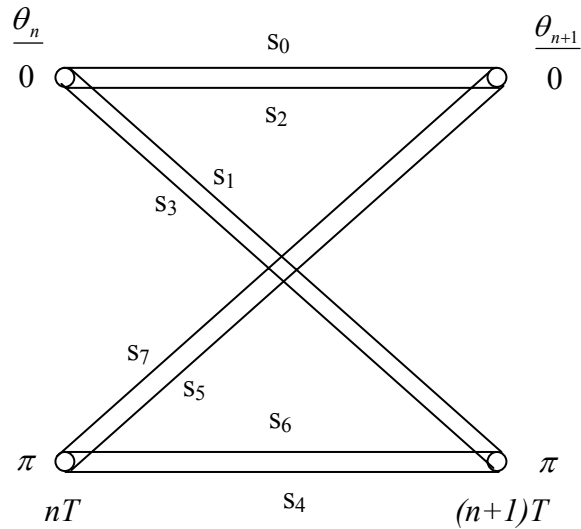
$$s(t, \underline{\beta}) = \sqrt{\frac{2E}{T}} \cos \left[2\pi f_1 t + \pi \beta_n \frac{t - nT}{T} + \theta_n \right] \quad (3.52)$$

$$\alpha_n \in \{-3, -1, +1, +3\} \Rightarrow \beta_n \in \{0, 1, 2, 3\}$$

$$\theta_n \in \{0, \pi\}$$

$$R_s = 2 \text{ bit/simge}$$

$$d_{serbest}^2 = 4E$$



Şekil 3.62 4'lü CPFSK, $h=1/2$ 'nin işaret kümesi.

Şekil 3.62'de görüldüğü gibi 8 sinyale ilişkin faz geçiş kafesi verilmektedir. Burada başlangıç açısı θ_n ve bitiş açısı da θ_{n+1} 'dir. Bu açı bilgisi bir bellek kullanılarak bir sonraki sinyale geçirilebilmekte, belleğin 0 ve 1 durumları açısal olarak 0 ve π değerlerini göstermektedir.

Tablo 3.7 4'lü CPFSK, $h=1/2$ değerleri

θ_n	β_n	$\Psi(t, \underline{\beta})$	$\Psi(nT, \underline{\beta})$	$\Psi((n+1)T, \underline{\beta})$	s_i	θ_{n+1}
0	0	0	0	0	s_0	0
	1	$\pi \frac{(t-nT)}{T}$	0	π	s_1	π
	2	$2\pi \frac{(t-nT)}{T}$	0	2π	s_2	0
	3	$3\pi \frac{(t-nT)}{T}$	0	3π	s_3	π
π	0	π	π	π	$s_4 = -s_0$	π
	1	$\pi \frac{(t-nT)}{T} + \pi$	π	2π	$s_5 = -s_1$	0
	2	$2\pi \frac{(t-nT)}{T} + \pi$	π	3π	$s_6 = -s_2$	π
	3	$3\pi \frac{(t-nT)}{T} + \pi$	π	4π	$s_7 = -s_3$	0

Bu tabloda θ_n başlangıç açısını, β_n katlamalı kodlayıcı çıkışını, $\psi(t, \beta)$ frekansın ne kadar kaydırılacağını, $\psi(nT, \beta)$ bulunulan fazı, $\psi((n+1)T, \beta)$ $\psi(t, \beta)$ ve $\psi(nT, \beta)$ 'ya göre bitiş fazının ne olacağını, s_i faz açılarına ve Şekil 3.62'ye göre oluşacak 8 sinyali, θ_{n+1} de bitiş açısını göstermektedir.

3.5.2.3. 8'li CPFSK, h=1/2

Burada $h=1/2$ olduğundan iki ayrı faz değeri vardır, bu faz değerini saklamak için 1 bellek kullanmak gerekir. $M=8$ olduğundan toplam 16 sinyal oluşur.

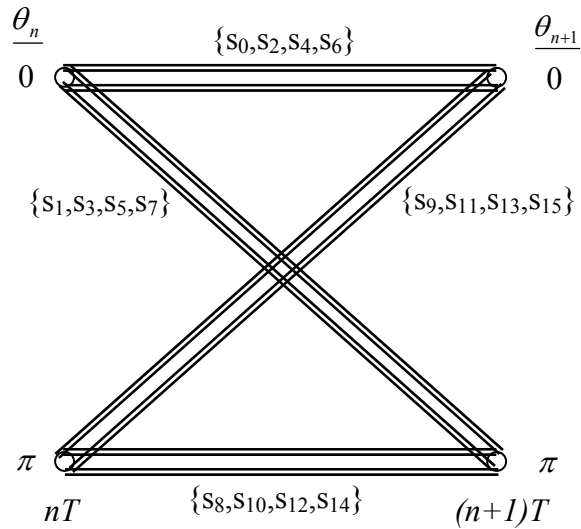
$$s(t, \underline{\beta}) = \sqrt{\frac{2E}{T}} \cos \left[2\pi f_1 t + \pi \beta_n \frac{t - nT}{T} + \theta_n \right] \quad (3.53)$$

$$\alpha_n \in \{-7, -5, -3, -1, +1, +3, +5, +7\} \quad \beta_n \in \{0, 1, 2, 3, 4, 5, 6, 7\}$$

$$\theta_n \in \{0, \pi\}$$

$$R_s = 3 \text{ bit/simge}$$

$$d_{serbest}^2 = 2E$$



Şekil 3.63 8'li CPFSK, $h=1/2$ 'nin işaret kümesi.

Şekil 3.63'de görüldüğü gibi 16 sinyale ilişkin faz geçiş kafesi verilmektedir. Burada başlangıç açısı θ_n ve bitiş açısı da θ_{n+1} 'dir. Bu açı bilgisi bir bellek kullanılarak bir sonraki sinyale geçirilebilmekte, belleğin 0 ve 1 durumları açısal olarak 0 ve π değerlerini göstermektedir.

Tablo 3.8 8’li CPFSK, h=1/2 değerleri

θ_n	β_n	$\Psi(t, \underline{\beta})$	$\Psi(nT, \underline{\beta})$	$\Psi((n+1)T, \underline{\beta})$	s_i	θ_{n+1}
0	0	0	0	0	s_0	0
	1	$\pi \frac{(t-nT)}{T}$	0	π	s_1	π
	2	$2\pi \frac{(t-nT)}{T}$	0	2π	s_2	0
	3	$3\pi \frac{(t-nT)}{T}$	0	3π	s_3	π
	4	$4\pi \frac{(t-nT)}{T}$	0	4π	s_4	0
	5	$5\pi \frac{(t-nT)}{T}$	0	5π	s_5	π
	6	$6\pi \frac{(t-nT)}{T}$	0	6π	s_6	0
	7	$7\pi \frac{(t-nT)}{T}$	0	7π	s_7	π

Tablo 3.8 8’li CPFSK, h=1/2 değerleri (Devam)

θ_n	β_n	$\Psi(t, \underline{\beta})$	$\Psi(nT, \underline{\beta})$	$\Psi((n+1)T, \underline{\beta})$	s_i	θ_{n+1}
π	0	π	π	π	$s_8^- - s_0$	π
	1	$\pi \frac{(t - nT)}{T} + \pi$	π	2π	$s_9^- - s_1$	0
	2	$2\pi \frac{(t - nT)}{T} + \pi$	π	3π	$s_{10}^- - s_2$	π
	3	$3\pi \frac{(t - nT)}{T} + \pi$	π	4π	$s_{11}^- - s_3$	0
	4	$4\pi \frac{(t - nT)}{T} + \pi$	π	5π	$s_{12}^- - s_4$	π
	5	$5\pi \frac{(t - nT)}{T} + \pi$	π	6π	$s_{13}^- - s_5$	0
	6	$6\pi \frac{(t - nT)}{T} + \pi$	π	7π	$s_{14}^- - s_6$	π
	7	$7\pi \frac{(t - nT)}{T} + \pi$	π	8π	$s_{15}^- - s_7$	0

Bu tabloda θ_n başlangıç açısını, β_n katlamalı kodlayıcı çıkışını, $\psi(t, \beta)$ frekansın ne kadar kaydırılacağını, $\psi(nT, \beta)$ bulunulan fazı, $\psi((n+1)T, \beta)$ $\psi(t, \beta)$ ve $\psi(nT, \beta)$ ’ya göre bitiş fazının ne olacağını, s_i faz açılarna ve Şekil 3.63’ye göre oluşacak 16 sinyali, θ_{n+1} de bitiş açısını göstermektedir.

3.5.2.4. 16'lı CPFSK, $h=1/2$

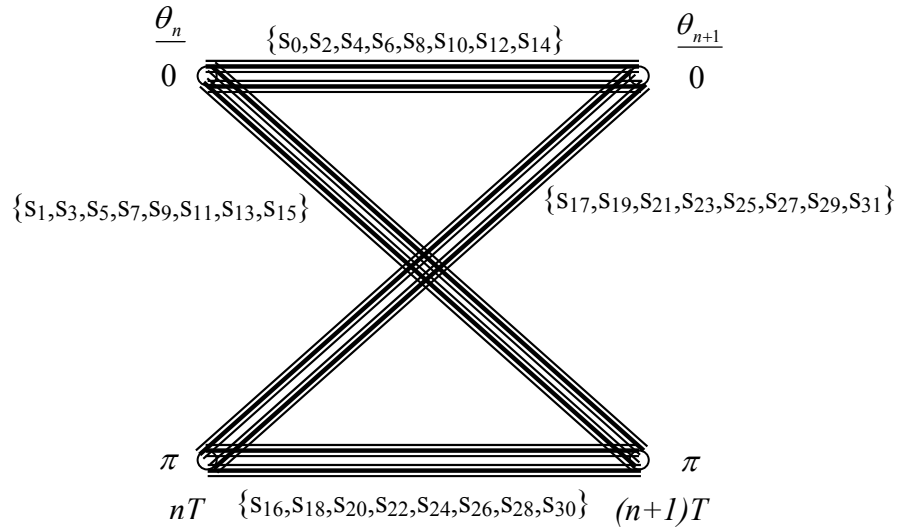
Burada $h=1/2$ olduğundan iki ayrı faz değeri vardır, bu faz değerini saklamak için 1 bellek kullanmak gerekir. $M=16$ olduğundan toplam 32 sinyal oluşur.

$$s(t, \underline{\beta}) = \sqrt{\frac{2E}{T}} \cos \left[2\pi f_1 t + \pi \beta_n \frac{t - nT}{T} + \theta_n \right] \quad (3.54)$$

$$\alpha_n \in \{-15, -13, -11, -9, -7, -5, -3, -1, +1, +3, +5, +7, +9, +11, +13, +15\}$$

$$\Rightarrow \beta_n \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$$

$$\theta_n \in \{0, \pi\}, \quad R_s=4 \text{ bit/simge}, \quad d_{\text{serbest}}^2 = 2E$$



Şekil 3.64 16'lı CPFSK, $h=1/2$ 'nin işaret kümesi.

Şekil 3.64'de görüldüğü gibi 32 sinyale ilişkin faz geçiş kafesi verilmektedir. Burada başlangıç açısı θ_n ve bitiş açısı da θ_{n+1} 'dir. Bu açı bilgisi bir bellek kullanılarak bir sonraki sinyale geçirilebilmekte, belleğin 0 ve 1 durumları açısal olarak 0 ve π değerlerini göstermektedir.

Tablo 3.9 16'lı CPFSK, h=1/2 değerleri

θ_n	β_n	$\Psi(t, \underline{\beta})$	$\Psi(nT, \underline{\beta})$	$\Psi((n+1)T, \underline{\beta})$	s_i	θ_{n+1}
0	0	0	0	0	s_0	0
	1	$\pi \frac{(t-nT)}{T}$	0	π	s_1	π
	2	$2\pi \frac{(t-nT)}{T}$	0	2π	s_2	0
	3	$3\pi \frac{(t-nT)}{T}$	0	3π	s_3	π
	4	$4\pi \frac{(t-nT)}{T}$	0	4π	s_4	0
	5	$5\pi \frac{(t-nT)}{T}$	0	5π	s_5	π
	6	$6\pi \frac{(t-nT)}{T}$	0	6π	s_6	0
	7	$7\pi \frac{(t-nT)}{T}$	0	7π	s_7	π

Tablo 3.9 16'lı CPFSK, $h=1/2$ değerleri (Devam)

0	8	$8\pi \frac{(t-nT)}{T}$	0	8π	s_8	0
	9	$9\pi \frac{(t-nT)}{T}$	0	9π	s_9	π
	10	$10\pi \frac{(t-nT)}{T}$	0	10π	s_{10}	0
	11	$11\pi \frac{(t-nT)}{T}$	0	11π	s_{11}	π
	12	$12\pi \frac{(t-nT)}{T}$	0	12π	s_{12}	0
	13	$13\pi \frac{(t-nT)}{T}$	0	13π	s_{13}	π
	14	$14\pi \frac{(t-nT)}{T}$	0	14π	s_{14}	0
	15	$15\pi \frac{(t-nT)}{T}$	0	15π	s_{15}	π

Tablo 3.9 16'lı CPFSK, h=1/2 değerleri (Devam)

θ_n	β_n	$\Psi(t, \underline{\beta})$	$\Psi(nT, \underline{\beta})$	$\Psi((n+1)T, \underline{\beta})$	s_i	θ_{n+1}
π	0	π	π	π	$s_{16}^- - s_0$	π
	1	$\pi \frac{(t-nT)}{T} + \pi$	π	2π	$s_{17}^- - s_1$	0
	2	$2\pi \frac{(t-nT)}{T} + \pi$	π	3π	$s_{18}^- - s_2$	π
	3	$3\pi \frac{(t-nT)}{T} + \pi$	π	4π	$s_{19}^- - s_3$	0
	4	$4\pi \frac{(t-nT)}{T} + \pi$	π	5π	$s_{20}^- - s_4$	π
	5	$5\pi \frac{(t-nT)}{T} + \pi$	π	6π	$s_{21}^- - s_5$	0
	6	$6\pi \frac{(t-nT)}{T} + \pi$	π	7π	$s_{22}^- - s_6$	π
	7	$7\pi \frac{(t-nT)}{T} + \pi$	π	8π	$s_{23}^- - s_7$	0

Tablo 3.9 16'lı CPFSK, h=1/2 değerleri (Devam)

θ_n	β_n	$\Psi(t, \underline{\beta})$	$\Psi(nT, \underline{\beta})$	$\Psi((n+1)T, \underline{\beta})$	s_i	θ_{n+1}
π	8	$8\pi \frac{(t-nT)}{T} + \pi$	π	9π	$s_{24} = -s_8$	π
	9	$9\pi \frac{(t-nT)}{T} + \pi$	π	10π	$s_{25} = -s_9$	0
	10	$10\pi \frac{(t-nT)}{T} + \pi$	π	11π	$s_{26} = -s_{10}$	π
	11	$11\pi \frac{(t-nT)}{T} + \pi$	π	12π	$s_{27} = -s_{11}$	0
	12	$12\pi \frac{(t-nT)}{T} + \pi$	π	13π	$s_{28} = -s_{12}$	π
	13	$13\pi \frac{(t-nT)}{T} + \pi$	π	14π	$s_{29} = -s_{13}$	0
	14	$14\pi \frac{(t-nT)}{T} + \pi$	π	15π	$s_{30} = -s_{14}$	π
	15	$15\pi \frac{(t-nT)}{T} + \pi$	π	16π	$s_{31} = -s_{15}$	0

Bu tabloda θ_n başlangıç açısını, β_n katlamalı kodlayıcı çıkışını, $\psi(t, \beta)$ frekansın ne kadar kaydırılacağını, $\psi(nT, \beta)$ bulunulan fazı, $\psi((n+1)T, \beta)$ $\psi(t, \beta)$ ve $\psi(nT, \beta)$ 'ya göre bitiş fazının ne olacağını, s_i faz açılarını ve Şekil 3.64'e göre oluşacak 32 sinyali, θ_{n+1} de bitiş açısını göstermektedir.

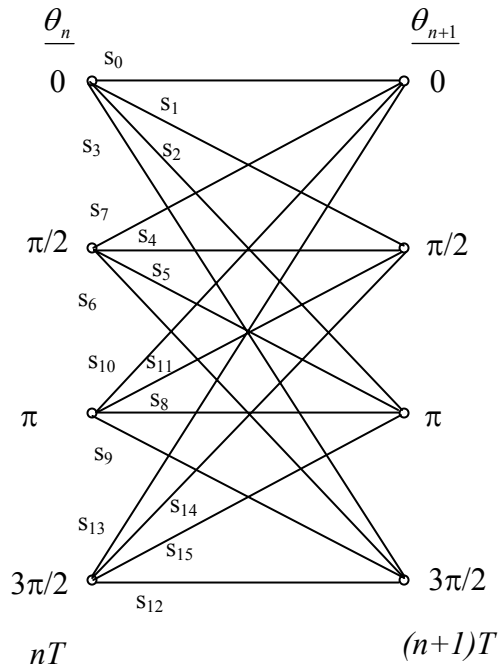
3.5.2.5. 4'lü CPFSK, $h=1/4$

Burada $h=1/4$ olduğundan dört ayrı faz değeri vardır, bu faz değerini saklamak için 2 bellek kullanmak gerekir. $M=4$ olduğundan toplam 16 sinyal oluşur.

$$s(t, \underline{\beta}) = \sqrt{\frac{2E}{T}} \cos \left[2\pi f_1 t + \frac{\pi}{2} \beta_n \frac{t - nT}{T} + \theta_n \right] \quad (3.55)$$

$$\alpha_n \in \{-3, -1, +1, +3\} \Rightarrow \beta_n \in \{0, 1, 2, 3\}$$

$$\theta_n \in \{0, \pi/2, \pi, 3\pi/2\}, R_s=2 \text{ bit/simge}, d_{serbest}^2 = 1.44E$$



Şekil 3.65 4'lü CPFSK, $h=1/4$ 'ün işaret kümesi.

Şekil 3.65'de görüldüğü gibi 16 sinyale ilişkin faz geçiş kafesi verilmektedir. Burada başlangıç açısı θ_n ve bitiş açısı da θ_{n+1} 'dir. Bu açı bilgisi 2 bellek kullanılarak bir sonraki sinyale geçirilebilmekte, belleklerin 00, 01, 10 ve 11 durumları açısal olarak 0, $\pi/2$, π ve $3\pi/2$ değerlerini göstermektedir.

Tablo 3.10 4'lü CPFSK, h=1/4 değerleri

θ_n	β_n	$\Psi(t, \underline{\beta})$	$\Psi(nT, \underline{\beta})$	$\Psi((n+1)T, \underline{\beta})$	s_i	θ_{n+1}
0	0	0	0	0	s_0	0
	1	$\frac{\pi}{2} \frac{(t-nT)}{T}$	0	$\frac{\pi}{2}$	s_1	$\frac{\pi}{2}$
	2	$\pi \frac{(t-nT)}{T}$	0	π	s_2	π
	3	$\frac{3\pi}{2} \frac{(t-nT)}{T}$	0	$\frac{3\pi}{2}$	s_3	$\frac{3\pi}{2}$
$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\pi}{2}$	s_4	$\frac{\pi}{2}$
	1	$\frac{\pi}{2} \frac{(t-nT)}{T} + \frac{\pi}{2}$	$\frac{\pi}{2}$	π	s_5	π
	2	$\pi \frac{(t-nT)}{T} + \frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{3\pi}{2}$	s_6	$\frac{3\pi}{2}$
	3	$\frac{3\pi}{2} \frac{(t-nT)}{T} + \frac{\pi}{2}$	$\frac{\pi}{2}$	2π	s_7	0

Tablo 3.10 4'lü CPFSK, $h=1/4$ değerleri (Devam)

θ_n	β_n	$\Psi(t, \underline{\beta})$	$\Psi(nT, \underline{\beta})$	$\Psi((n+1)T, \underline{\beta})$	s_i	θ_{n+1}
π	0	π	π	π	$s_8 = -s_0$	π
	1	$\frac{\pi}{2} \frac{(t - nT)}{T} + \pi$	π	$\frac{3\pi}{2}$	$s_9 = -s_1$	$\frac{3\pi}{2}$
	2	$\pi \frac{(t - nT)}{T} + \pi$	π	2π	$s_{10} = -s_2$	0
	3	$\frac{3\pi}{2} \frac{(t - nT)}{T} + \pi$	π	$\frac{5\pi}{2}$	$s_{11} = -s_3$	$\frac{\pi}{2}$
$\frac{3\pi}{2}$	0	$\frac{3\pi}{2}$	$\frac{3\pi}{2}$	$\frac{3\pi}{2}$	$s_{12} = -s_4$	$\frac{3\pi}{2}$
	1	$\frac{\pi}{2} \frac{(t - nT)}{T} + \frac{3\pi}{2}$	$\frac{3\pi}{2}$	2π	$s_{13} = -s_5$	0
	2	$\pi \frac{(t - nT)}{T} + \frac{3\pi}{2}$	$\frac{3\pi}{2}$	$\frac{5\pi}{2}$	$s_{14} = -s_6$	$\frac{\pi}{2}$
	3	$\frac{3\pi}{2} \frac{(t - nT)}{T} + \frac{3\pi}{2}$	$\frac{3\pi}{2}$	3π	$s_{15} = -s_7$	π

Bu tabloda θ_n başlangıç açısını, β_n katlamalı kodlayıcı çıkışını, $\psi(t, \beta)$ frekansın ne kadar kaydırılacağını, $\psi(nT, \beta)$ bulunulan fazı, $\psi((n+1)T, \beta)$ $\psi(t, \beta)$ ve $\psi(nT, \beta)$ 'ya göre bitiş fazının ne olacağını, s_i faz açılarına ve Şekil 3.65'e göre oluşacak 16 sinyali, θ_{n+1} de bitiş açısını göstermektedir.

3.5.3. Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama

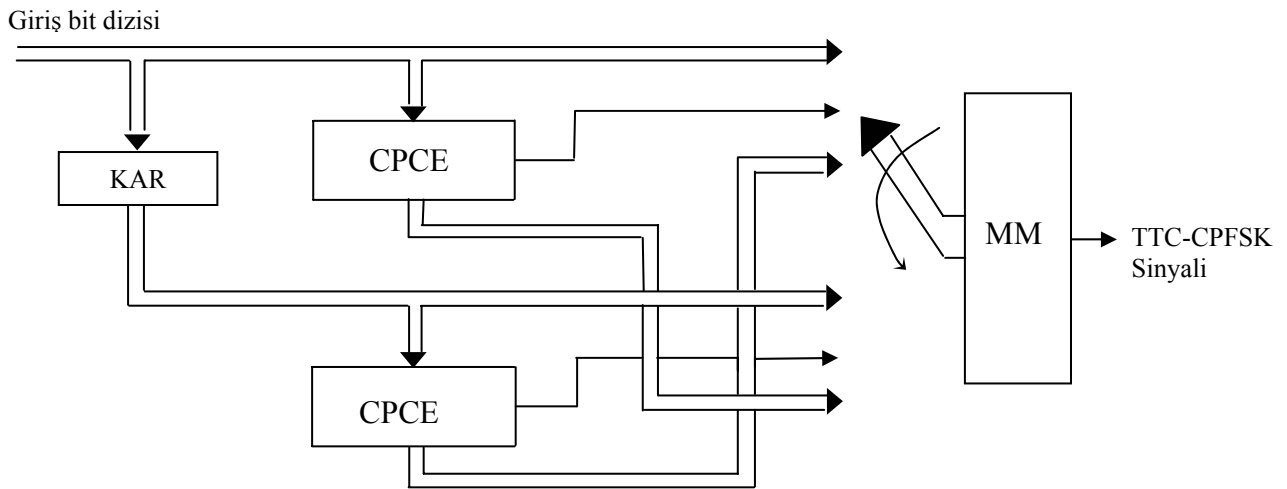
Bu bölümde CPFSK ve TTCM birleştirilerek Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama (Turbo Trellis Coded – Continuous Phase Frequency Shift Keying, TTC-CPFSK) adı verilen yeni bir kodlama tasarımı gerçekleştirilmiştir. Faz sürekliliği sağlanarak band verimliliği ve hata başarımı oldukça arttırılmıştır. Önerilen bu yeni kodlama yönteminde sürekli fazı sağlayabilmek için başlangıç faz bilgisinin tutulacağı belleklere ihtiyaç duyulmayıp bu iş için kodlayıcının belleklerinden yararlanılır. Bu sayede hem kodlama hem de kod çözme işlemleri sırasındaki karmaşıklıkta bir artma olmaz.

3.5.3.1. Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama Kodlayıcı Yapısı

TTC-CPFSK kodlayıcı yapısı, sürekli fazlı sinyaller gönderilebilecek şekilde tasarlanmıştır. CPFSK sinyalde ϕ_n o anki sinyalin başlangıç fazı ve bir önceki sinyalin bitiş fazını gösterir. Bunda dolayı, bir önceki sinyalin bitiş açısı sürekliliğin sağlanabilmesi için bilinmelidir. Eğer sonunda bir anahtar olmayan tek bir kodlayıcı yapısı olsaydı, birleşik kodun kafesi başlangıç ve bitiş fazlarını gösterecekti. Fakat kodlayıcı çıkışlarında bir anahtarın bulunmasından dolayı, bir önceki bilgiye ulaşılamayacağından, sürekliliği bu yapı ile sağlamak mümkün değildir. Çünkü bir sonraki kodlama adımında anahtar diğer kodlayıcının durumuna geçecek ve bir önceki bilgiler kullanılamayacaktır. Diğer bir engel de alt koldaki düzenleyicidir. Çünkü

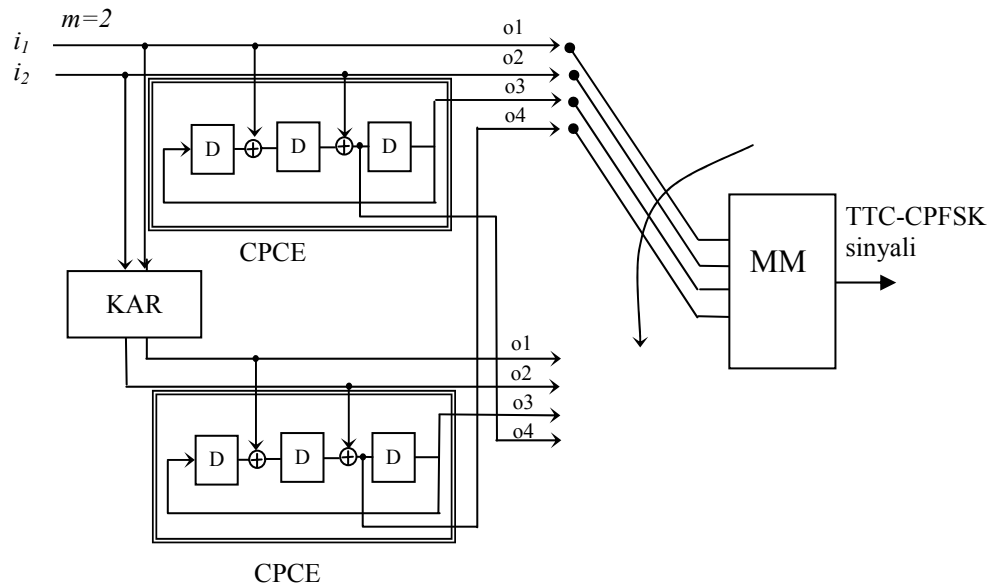
CPM’de CPE ve MM arasındaki herhangi bir düzenleyici (DZN) simgelerin dizilimlerini değiştirecek ve fazdaki sürekliliği ortadan kaldıracaktır.

Sistemi sürekli faz üretbilmesi için öncelikle vericideki kodlayıcı yapısında alt koldaki düzenleyici kaldırılır. Buna karşılık alıcıdaki karıştırıcı yapısı da değiştirilir. TTCM’de iki tane bir birinin aynı kodlayıcı vardır ve her adımda sadece birinin çıktısı kullanılır. TTC-CPFSK’da gönderilecek sinyalin bitiş fazının bilinmesi gerekir. Bu da diğer kodlayıcının bir sonraki adımda oluşturacağı sinyalin başlangıç fazının bilinmesi ve bu bilginin kullanılması ile olur. Bu nedenle, diğer kodlayıcının sürekli faz kodlayıcısı (Continuous Phase Encoder, CPE) belleklerinin girişi MM’e ek bir giriş olarak kullanılır ve bu şekilde süreklilik sağlanır. Şekil 3.66’da TTC-CPFSK’nın genel yapısı gösterilmiştir. CPE için katlamalı kodlayıcıdaki belleklerden istenildiği kadar kullanılabilir. Bu sayede diğer kodlayıcıdan gelen ek bağlantılarla CPFSK’nın modülasyon indeksi h ve sinyal boyutu değiştirilebilir.



Şekil 3.66 TTC-CPFSK’nın genel yapısı

Buradaki diğer bir yeni yaklaşım da CPE için ek bir bellek kullanılmasının gerekmediğidir. CE'deki belleklerden bazıları ortak bellek olarak kullanılır ve bu şekilde TTCM kodlayıcısının karmaşıklığı arttırılmamış olur. CE ve CPE nin oluşturduğu bu yeni yapıya Sürekli Faz Katlamalı Kodlayıcı (CPCE) adı verilmiştir. Bu yapı ile sadece giriş bitleri kodlanmayıp aynı zamanda sürekli fazlı bir sinyal de oluşturulur. Tablo 3.11'de örnek olarak 16-CPFSK'nın giriş ve çıkış tablosu verilmiştir. Giriş olarak iki bit dizisinin var olduğu düşünülmektedir. Şekil 3.67'de 16-CPFSK için TTC-CPFSK yapısı verilmiştir. Burada CPE için yalnız bir bellek kullanılmış ve MM'e ekstradan bir bağlantı yapılmıştır. Bu yüzden modülasyon indeksi $h=1/2$ dir. Tablo 3.11'de bunlara karşılık gelen giriş çıkış ve sinyal dağılımı verilmektedir. "o1" ve "o2" ilk iki sistematik biti göstermektedir. "o3" kodlanmış parite bitidir. "o4" ise sürekliliğin sağlanması için kullanılan diğer kodlayıcının üçüncü bellek girişinden gelen bilgidir. "o4" bize bir sonraki kodlama adımında sinyalin hangi faz ile başlayacağını göstermektedir. Şekil 3.68'de faz geçişleri verilmektedir. CPE için sadece bir bellek kullanıldığından, "0" ve " π " olmak üzere iki faz vardır.

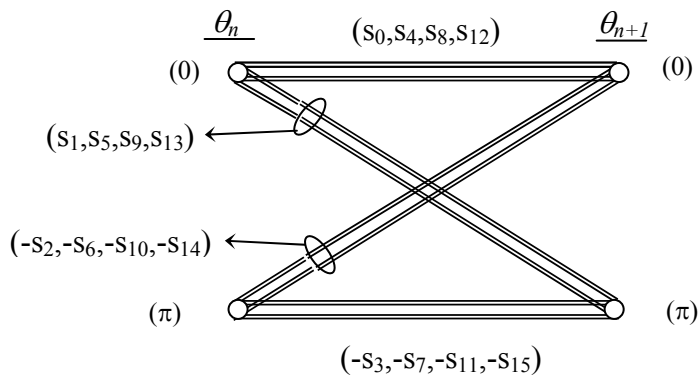


Şekil 3.67 16-CPFSK $h=1/2$ için TTC-CPFSK kodlayıcı yapısı.

Tablo 3.11 16-CPFSK için giriş çıkış ve sinyal dağılımı

$i1$	$i2$	$o1$	$o2$	$o3$	$o4$	β	16-CPFSK
0	0	0	0	0	0	0	s0
0	0	0	0	1	0	2	-s2
0	0	0	0	0	1	1	s1
0	0	0	0	1	1	3	-s3
0	1	0	1	0	1	5	s5
0	1	0	1	1	1	7	-s7
0	1	0	1	0	0	4	s4
0	1	0	1	1	0	6	-s6
1	0	1	0	0	0	8	s8
1	0	1	0	1	0	10	-s10
1	0	1	0	0	1	9	s9
1	0	1	0	1	1	11	-s11
1	1	1	1	0	1	13	s13
1	1	1	1	1	1	15	-s15
1	1	1	1	0	0	12	s12
1	1	1	1	1	0	14	-s14

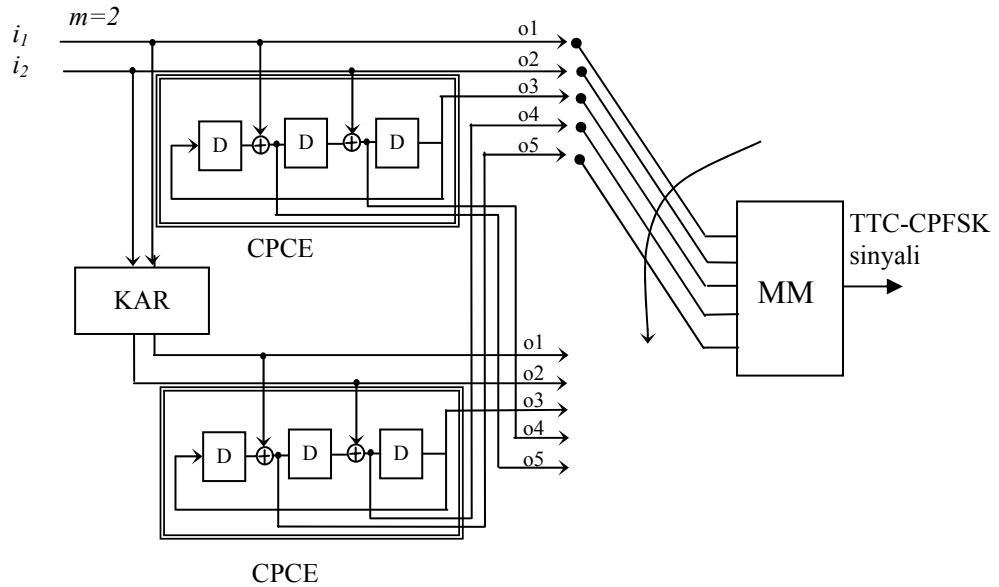
Bu tabloda $i1$ ve $i2$ kodlayıcı girişleri, $o1$ ve $o2$ sistematik data, $o3$ parite data ve $o4$ diğer kodlayıcının CPE belleğinin girişidir. Bu durumda 16-CPFSK sinyal, ikili kodlayıcı çıkışlarının onluk düzendeki hali β , başlangıç fazı θ_n ve sinyal işaretini belirleyen $o3$, bitiş fazı θ_{n+1} 'i belirleyen $o4$ değişkenleri ile oluşturulur.



Şekil 3.68 16- CPFSK için faz geçişleri

Şekil 3.69’da 36-CPFSK için TTC-CPFSK kodlayıcısının yapısı verilmiştir. Burada CPE için iki bellek kullanılmış, MM’e iki ek giriş yapılmış ve bundan dolayı $h = 1/4$ olmuştur. Tablo 3.12’de giriş çıkış ve sinyal dağılımı gösterilmektedir. Şekil 3.70’de de bu sisteme ilişkin faz geçişlerine ait kafes yapı gösterilmektedir. CPE için iki bellek kullanıldığından dört faz durumu oluşur bunlar sırasıyla; “0”, “ $\pi/2$ ”, “ π ” ve “ $3\pi/2$ ” dir.

Tablo 3.11’de “o1” ve “o2” ilk iki girişi , “o3” kodlanmış parite biti ve “o4” de diğer kodlayıcının üçüncü belleğinin girişini göstermektedir. Tablo 3.12’de “o1” ve “o2” ilk iki girişi , “o3” kodlanmış parite biti ve “o4” ile “o5” de diğer kodlayıcının ikinci ve üçüncü belleklerinin girişlerini göstermektedir. “o4” ve “o5” bir sonraki durumda sinyalin hangi fazla başlayacağını göstermektedir.



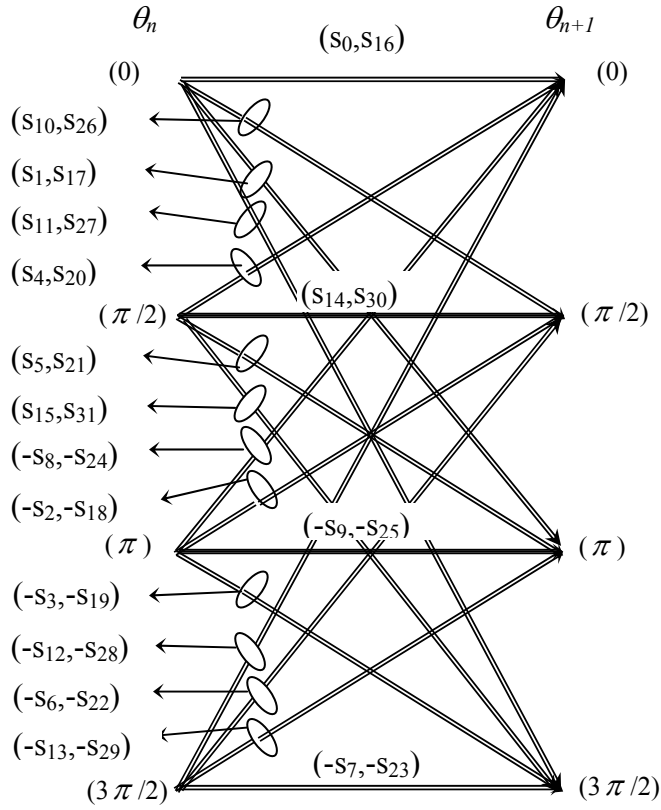
Şekil 3.69 32-CPFSK $h=1/2$ için TTC-CPFSK kodlayıcı yapısı

Tablo 3.12 32-CPFSK $h=1/4$ için TTC-CPFSK giriş çıkış tablosu

$i1$	$i2$	$o1$	$o2$	$o3$	$o4$	$o5$	β	32-CPFSK
0	0	0	0	0	0	0	0	s0
0	0	0	0	1	0	0	4	s4
0	0	0	0	0	1	0	2	-s2
0	0	0	0	1	1	0	6	-s6
0	0	0	0	0	0	1	1	s1
0	0	0	0	1	0	1	5	s5
0	0	0	0	0	1	1	3	-s3
0	0	0	0	1	1	1	7	-s7
0	1	0	1	0	1	0	10	s10
0	1	0	1	1	1	0	14	s14
0	1	0	1	0	0	0	8	-s8
0	1	0	1	1	0	0	12	-s12
0	1	0	1	0	1	1	11	s11
0	1	0	1	1	1	1	15	s15
0	1	0	1	0	0	1	9	-s9
0	1	0	1	1	0	1	13	-s13
1	0	1	0	0	0	1	17	s17
1	0	1	0	1	0	1	21	s21
1	0	1	0	0	1	1	19	-s19
1	0	1	0	1	1	1	23	-s23
1	0	1	0	0	0	0	16	s16
1	0	1	0	1	0	0	20	s20
1	0	1	0	0	1	0	18	-s18
1	0	1	0	1	1	0	22	-s22
1	1	1	1	0	1	1	27	s27
1	1	1	1	1	1	1	31	s31
1	1	1	1	0	0	1	25	-s25
1	1	1	1	1	0	1	29	-s29
1	1	1	1	0	1	0	26	s26
1	1	1	1	1	1	0	30	s30
1	1	1	1	0	0	0	24	-s24
1	1	1	1	1	0	0	28	-s28

Bu tabloda $i1$ ve $i2$ kodlayıcı girişleri, $o1$ ve $o2$ sistematik data, $o3$ parite data, $o4$ ve $o5$ diğer kodlayıcının CPE belleklerinin girişleridir. Bu durumda 32-CPFSK sinyal, ikili

kodlayıcı çıkışlarının onluk düzendeki hali β , başlangıç fazı θ_n ve sinyal işaretini belirleyen $o2$ ve $o3$, bitiş fazı θ_{n+1} 'i belirleyen $o4$ ve $o5$ değişkenleri ile oluşturulur.



Şekil 3.70 32-CPFSK h=1/4 faz geçişi

3.5.3.2. Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama Kodçözücü Yapısı

TTC-CPFSK kodlayıcı yapısında düzenleyici kullanmadığımız için alıcıdaki yapıda sadece üst kodlayıcı tarafından oluşturulan tek indeksli semboller karıştırılır, yani alıcıdaki karıştırıcı çift indekslilerin yerini değiştirmeyip tek indekslilerin yerini değiştirir. Alıcıdaki katlamalı kodlayıcının yapısında herhangi bir değişiklik olmamıştır. Bu nedenle kod çözme işleminde giriş dataları ve parite data ($o1$ $o2$ $o3$)

kullanılır, diğer kodlayıcıdan gelen bilgi ise sadece fazdaki sürekliliği sağlamak içindir. Tabloya bakıldığı zaman aynı (o1 o2 o3)’e sahip fakat farklı sinyallerin olduğu görülebilir. Örnek olarak, Tablo 3.11’deki ilk ve üçüncü satırlarda (o1 o2 o3) aynıdır, fakat o4 birinci satırı “s0”, üçüncü satırı da “s1” yapar. Kodlayıcıda bir karıştırıcı bulunduğundan, alt ve üst kollar arasında herhangi bir ilişki olmaması ve birlikte bir kafes yapı oluşturmaması “o4”ün kod çözme işlemi sırasında herhangi bir etkisinin olamayacağını gösterir.

CPM demodülatörü her alınan bir periyodluk sinyal için M tane ortonormal fonksiyon kullanarak M boyutlu bir sembol elde eder. Ölçekleyici hesaplamasında, X ve Y sırasıyla gönderilen ve alınan sinyalleri göstermek üzere M boyutlu sinyalin olasılığı Gauss dağılımı ile hesaplanır. M adet olası sinyal arasında aynı giriş ve parite datasına sahip birden fazla sinyalin olması, en olasılıklı olanı (most probable one MPO) işleminin yapılma gereğini doğurur. MPO, aynı giriş ve parite datasına sahip sinyallerin ölçekleyici değerleri arasından en yüksek olasılıklı olanını seçer. Bu tezde MPO’nun formülü aşağıdaki gibi önerilmiştir.

$$\mathbf{MPO} = [\max(\mathbf{MPO}_1), \max(\mathbf{MPO}_2), \dots, \max(\mathbf{MPO}_2^{m+1})] \quad (3.56)$$

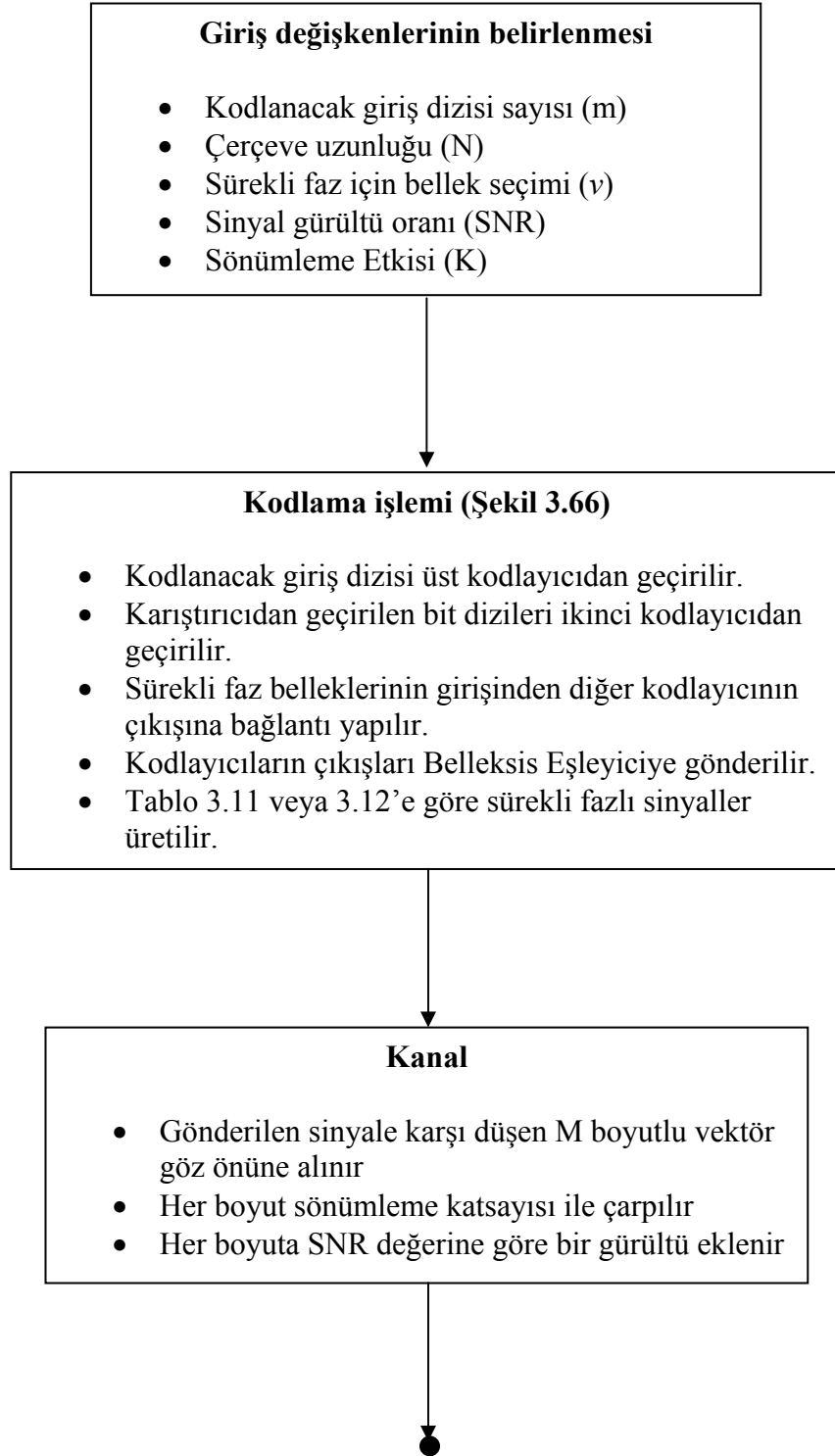
$\mathbf{MPO}$, her sinyal için 2^{m+1} elemandan oluşan bir vektördür. $\mathbf{MPO}_n$ ise aynı “o1 o2 o3” elemanlarına sahip sinyaller kümesidir.

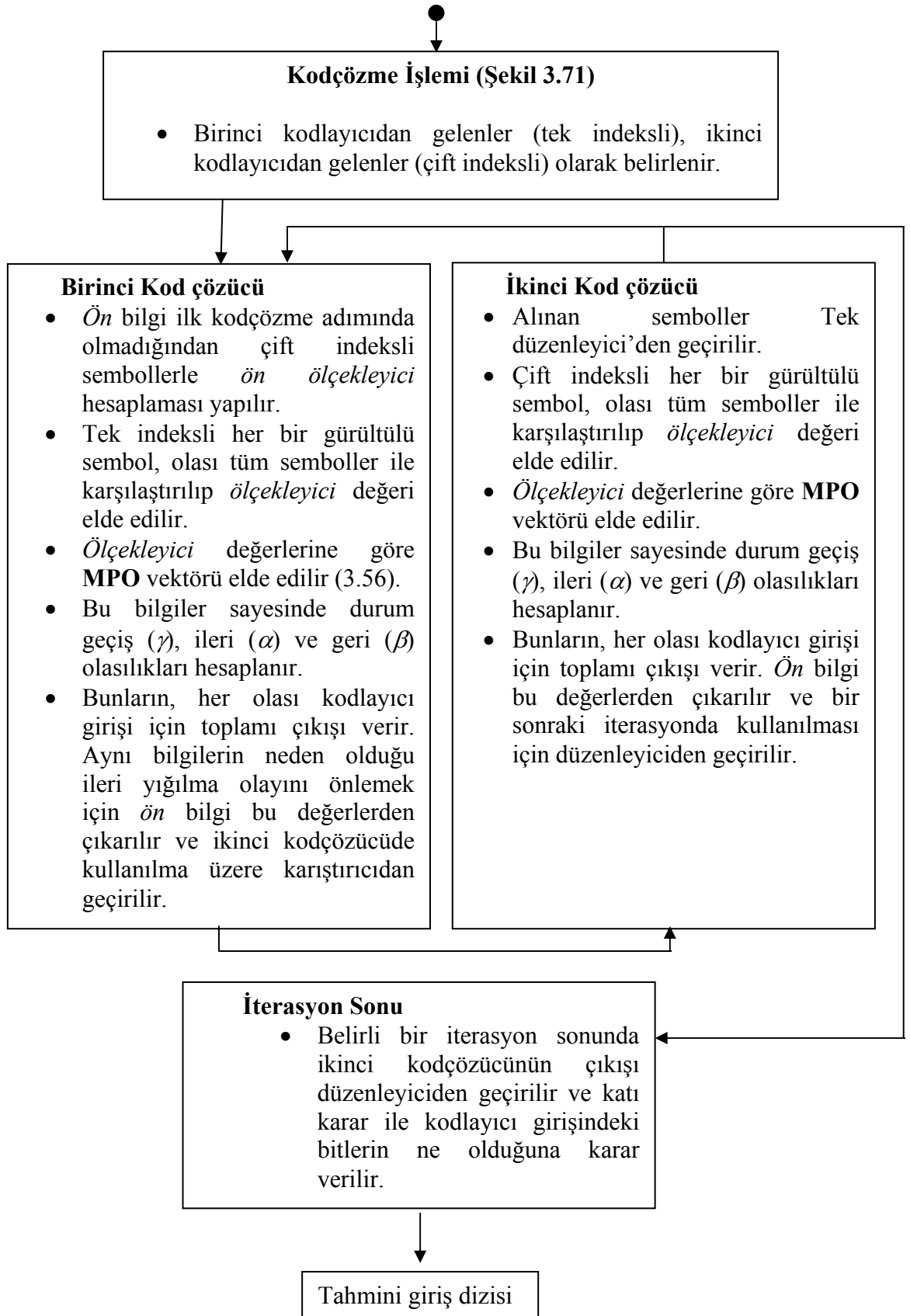
MPO işemini daha açık olarak anlatmak için, 16-CPFSK ve $m=2$ olduğu durumda logaritmik ölçekleyici sonuçlarının (-3.2; -41.4; -37.9; -48.5; -34.7; -39.5; -51.8; -44.1; -47.3; -41.5; -36.5; -42.6; -40.8; -42.7; -40.5; -48.8) olduğunu varsayalım. Aynı zamanda, 16-CPFSK ve $m=2$ için Tablo 3.11'e göre "s0" ve "s1"; "-s2" ve "-s3"; "s4" ve "s5"; "-s6" ve "-s7"; "s8" ve "s9"; "-s10" ve "-s11"; "s12" ve "s13"; "-s14" ve "-s15" sinyal kümelerinin aynı giriş ve parite datalarına sahip kümeler olduğu bilinmektedir. Buna göre **MPO** vektörü (-3.2; -37.9; -34.7; -44.1; -41.5; -36.5; -40.8; -40.5) şeklinde oluşur.

İlk bakışta bu seçme işlemi katı karar işlemi gibi görünsede, hiç bir (o1 o2 o3) girişi veya durumu ortadan kaldırılmamış, sadece diğer kodlayıcıdan gelen bilgi nedeniyle artan sinyal kümesi eski boyutu 2^{m+1} 'e döndürülmüştür. Başka bir taraftan da, aslında yapılan bu işlem diğer kodlayıcıdan gelen faz bilgisine karar verilmektedir, ama bu karar (o1 o2 o3) için bir kısıtlama yaratmamaktadır. Şekil 3.71'de TTC-CPFSK kod çözücüsünün genel yapısı verilmektedir. Ölçekleyici & MPO çıktısı, SISO kod çözücüsünün girişi olarak alınır ve burada 2. bölümde anlatılan TTCM için kullanılan MAP algoritması uygulanır.

Simülasyonda sistemin verici, kanal ve alıcı blok diyagramlarının nasıl gerçekleştirildiğini ve hangi değişkenlerin nasıl ve nerede belirlendiğini göstermek amacı ile aşağıdaki gibi bir akış diyagramı verilmiştir.

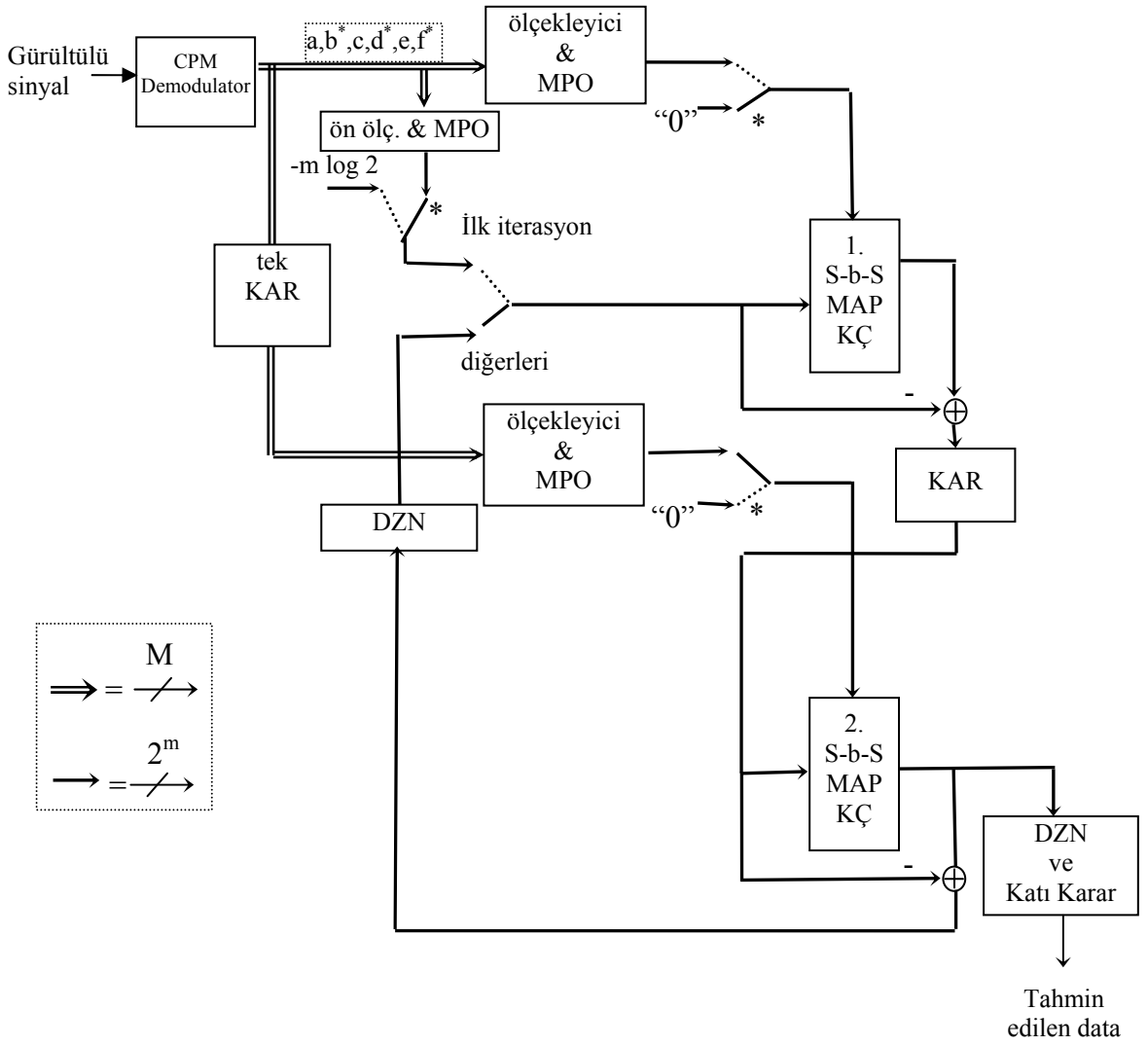
Tablo 3.13 TTC-CPFSK akış şeması





3.5.3.4. TTC-CPFSK simülasyon sonuçları

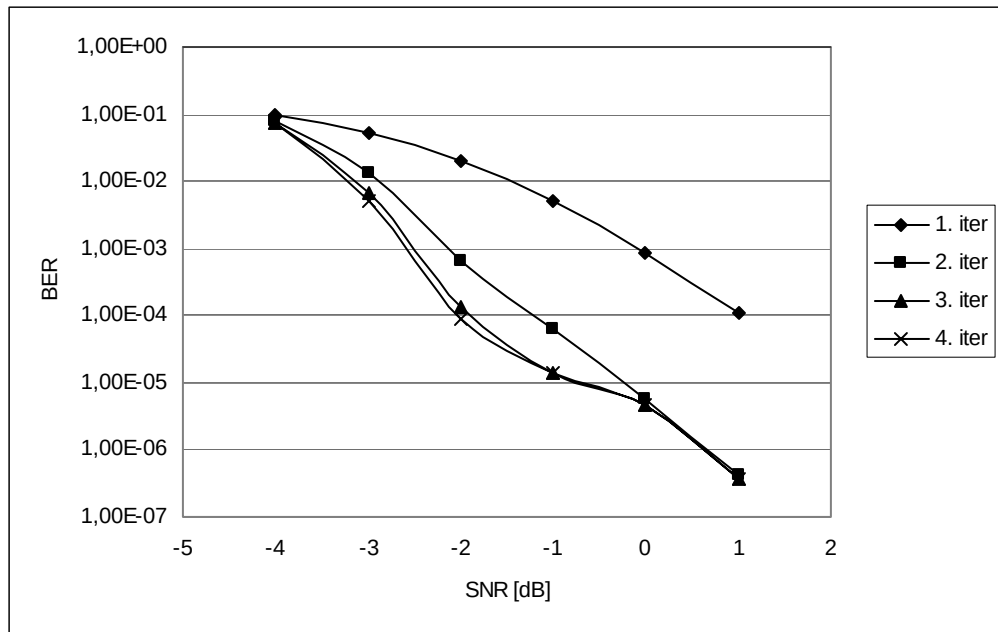
Simülasyonda TTC-CPFSK'nın iki girişi olduğu kabul edilip Şekil 3.69 ve Şekil 3.70 kodlayıcıları göz önünde bulundurulacaktır. Buna göre modülasyon, 16-CPFSK $h=1/2$ ve 32-CPFSK $h=1/4$ olacaktır.



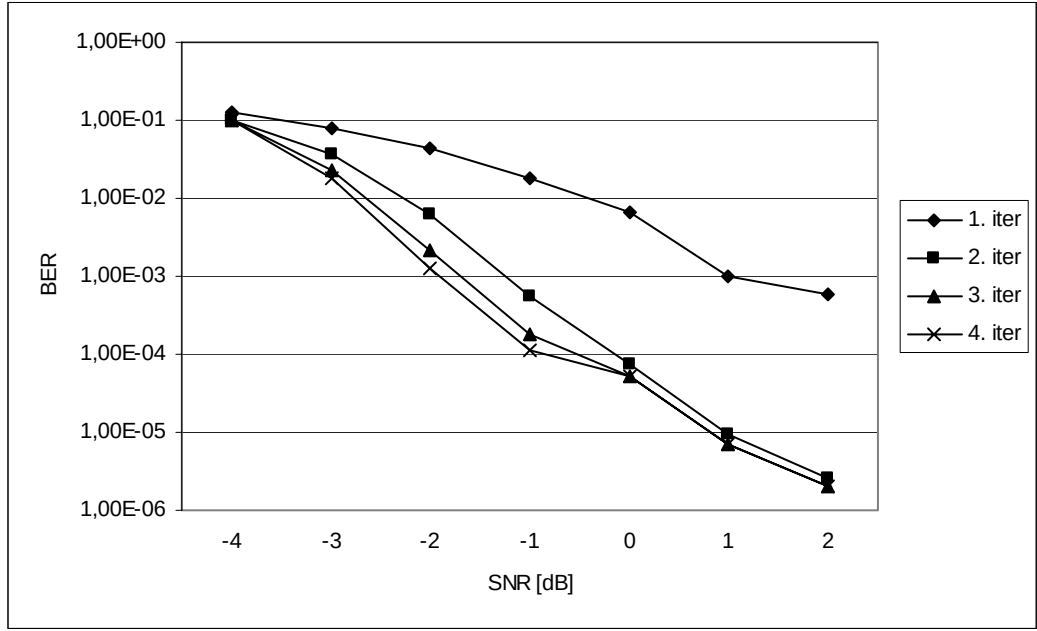
Her iki kodlayıcı yapısı için girişlerin bit uzunluğunun 256, 1024 ve 5000 olacağı düşünülmüş, kanal yapısının da AWGN, $K=10\text{dB}$ ve $K=4\text{dB}$ Rician olduğu varsayılmıştır. Buna göre kod çözücüde dört iterasyon uygulanacaktır.

Simülasyon MATLAB programı kullanılarak hazırlanmıştır. PIV 1.5 GHz bir bilgisayarda 4 iterasyonda, başlangıçta 256 bit uzunluklu bir çerçeve için 7.8 sn, 1024 bit uzunluklu bir çerçeve için 20 sn ve 5000 bit uzunluklu bir çerçeve için de 145 sn sürmektedir. Çerçeveler gönderilirken sistemin bellek özelliklerine ve işletim sistemine göre bu süre yavaş yavaş uzamaktadır. Belirli bir bit hata olasılığını istatistiksel olarak doğru yakalamak için gönderilmesi gereken bit sayısı şu şekilde hesaplanmaktadır. Örneğin belirli bir SNR değerinde 10^{-x} bit hata olasılığı yakalanıyor ise gönderilmiş bit sayısının minimum değerinin 10^{x+1} olması gerekmektedir. Bu da yaklaşık olarak o durumda en az 10 bitte hata olması gerektiğini söyler. Hatalı bit sayısı arttıkça bit hata olasılığı daha doğru olarak elde edilmiş olur.

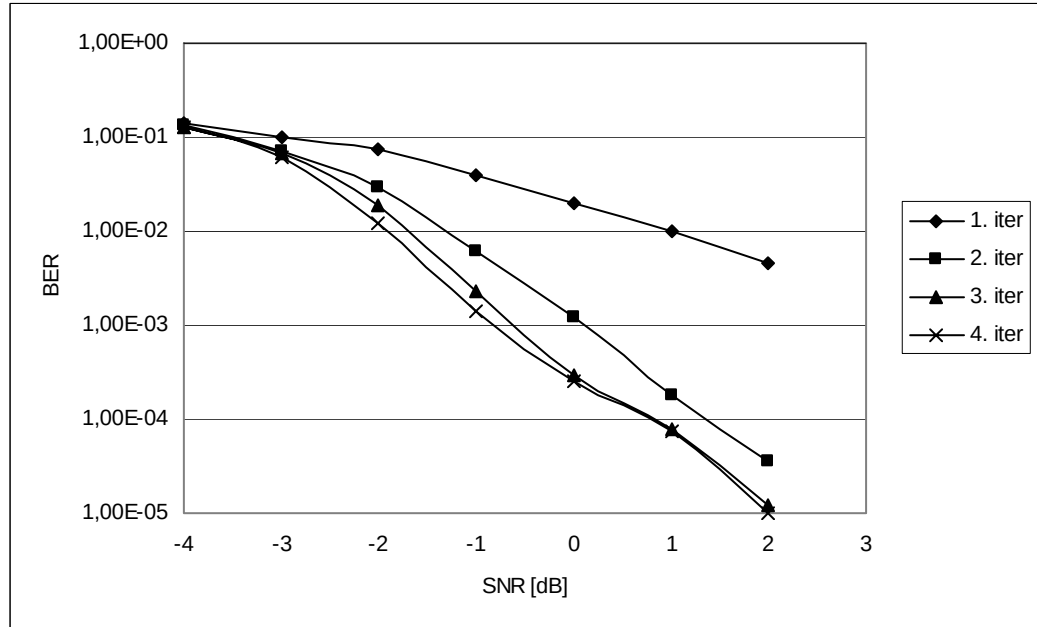
İlk üç şekil, Şekil 3.72, Şekil 3.73 ve Şekil 3.74 çerçeve boyutu $N=256$ olması durumunda 16-CPFSK modülasyonunda sırasıyla AWGN, $K=10\text{dB}$ ve $K=4\text{dB}$ Rician kanallar için bit hata oranını göstermektedir.



Şekil 3.72 16-CPFSK, $h=1/2$, $N=256$ için AWGN ortamda TTC-CPFSK başarımı



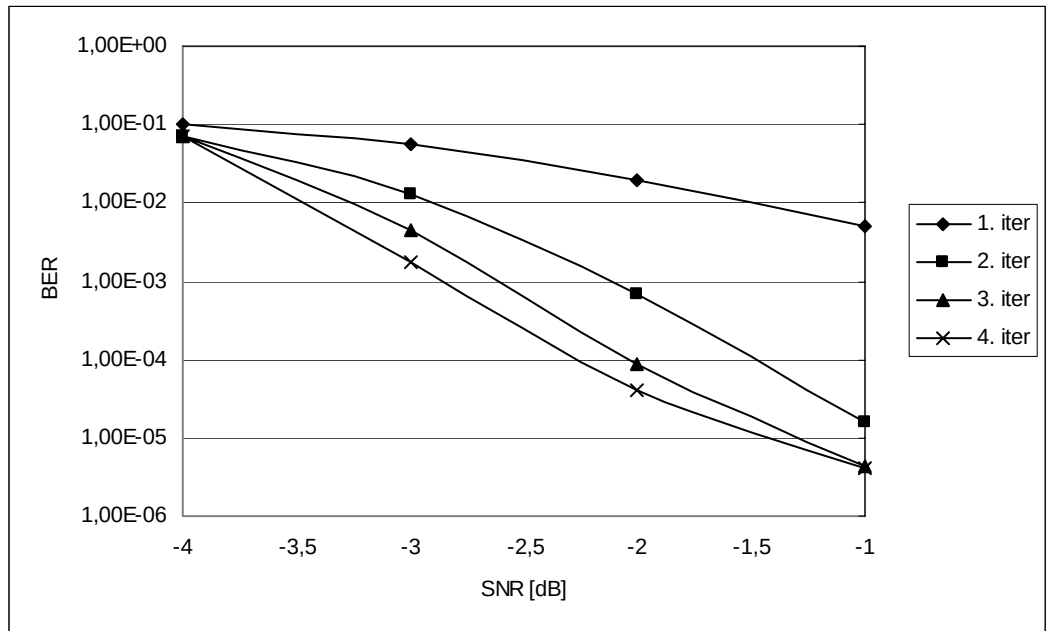
Şekil 3.73 16-CPFSK, $h=1/2$, $N=256$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı



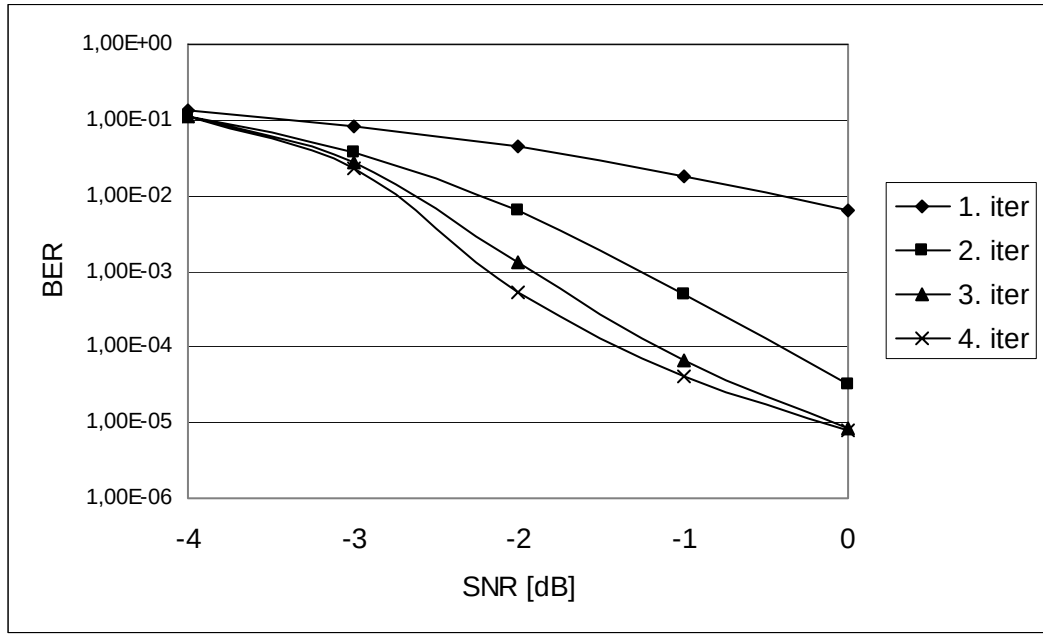
Şekil 3.74 16-CPFSK, $h=1/2$, $N=256$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı

Bu sonuçlara bakıldığında, bit hata oranının çerçeve boyutunun 265 olmasına rağmen çok iyi olduğu görülmektedir. Aynı zamanda, sönümleme etkisi arttıkça iterasyona gerek duyulduğu, AWGN ve K=10dB Rician kanal için ikinci iterasyonun yeterli olduğu, ancak K=4dB olduğunda üçüncü iterasyona gerek duyulabileceği anlaşılmaktadır.

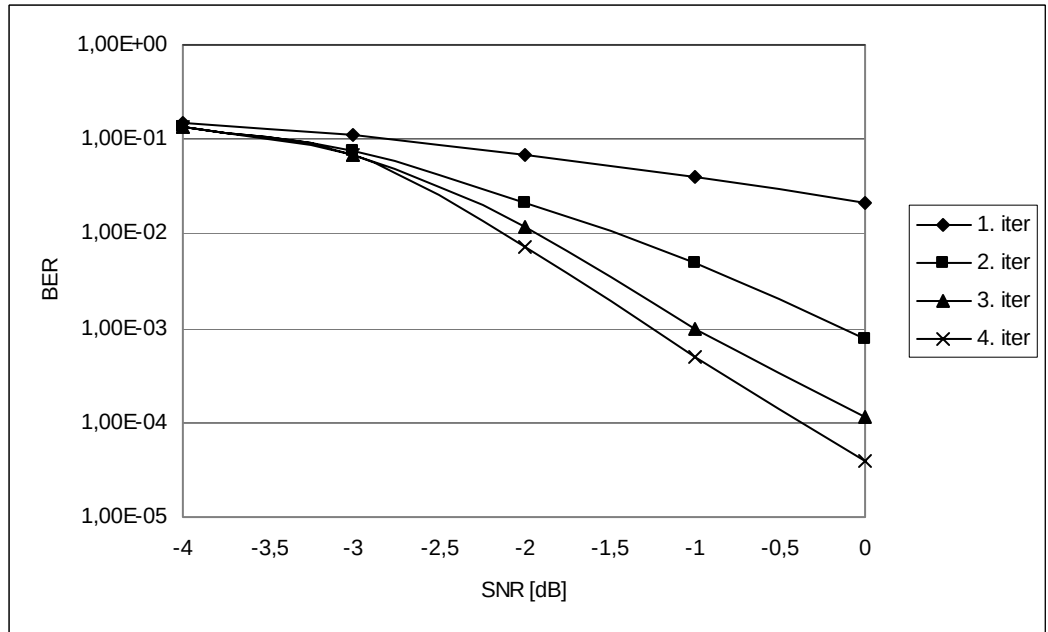
Bundan sonraki üç şekil, Şekil 3.75, Şekil 3.76 ve Şekil 3.77 çerçeve boyutu N=1024 olmak üzere 16-CPFSK modülasyonunda sırasıyla AWGN, K=10dB ve K=4dB Rician kanallar için bit hata oranını göstermektedir.



Şekil 3.75 16-CPFSK, $h=1/2$, N=1024 için AWGN ortamda TTC-CPFSK başarımları



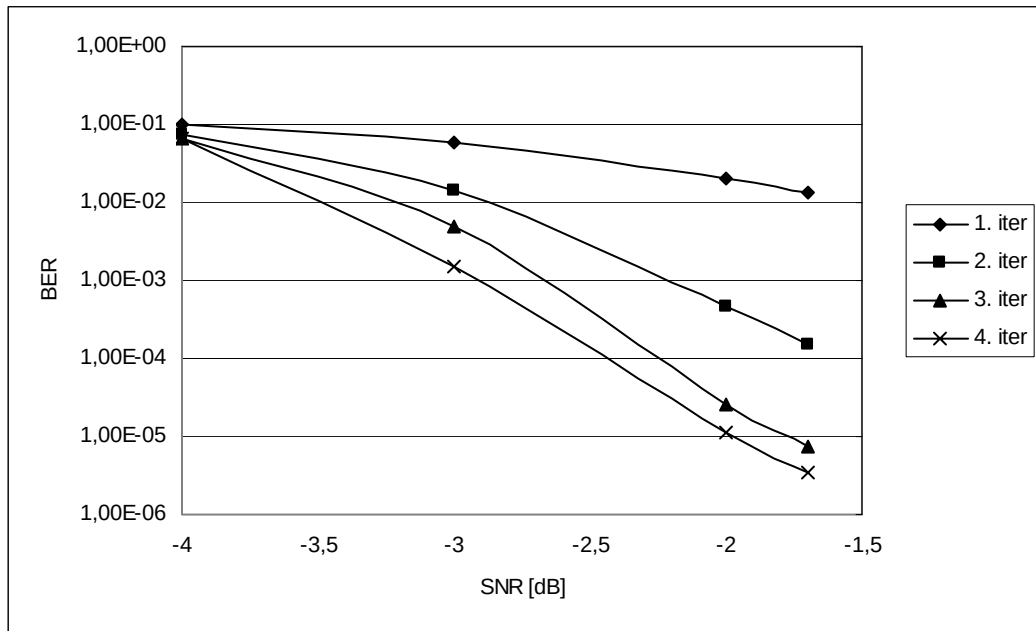
Şekil 3.76 16-CPFSK, $h=1/2$, $N=1024$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı



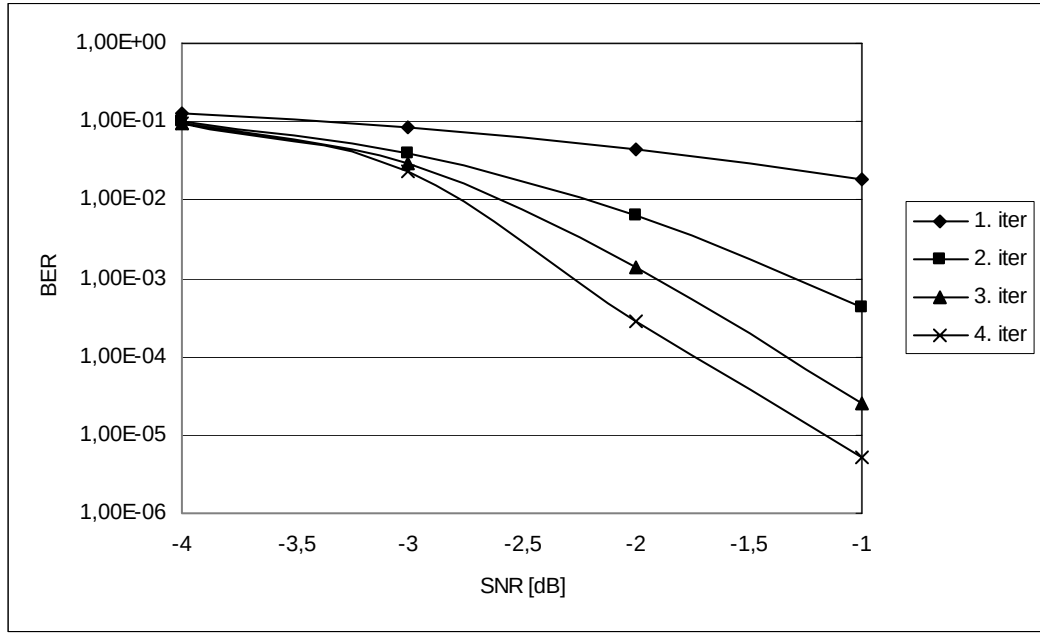
Şekil 3.77 16-CPFSK, $h=1/2$, $N=1024$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı

Bu sonuçlara bakıldığında, bit hata oranının çerçeve boyutunun 1024 olması durumunda, 256 olması durumuna göre daha iyi olduğu görülmektedir. Aynı zamanda, bir önceki durum gibi sönümleme etkisi arttıkça iterasyona gerek duyulduğu, AWGN ve $K=10\text{dB}$ Rician kanal için üçüncü iterasyonun yeterli olduğu, ancak $K=4\text{dB}$ olduğunda dördüncü iterasyona gerek duyulabileceği anlaşılmaktadır.

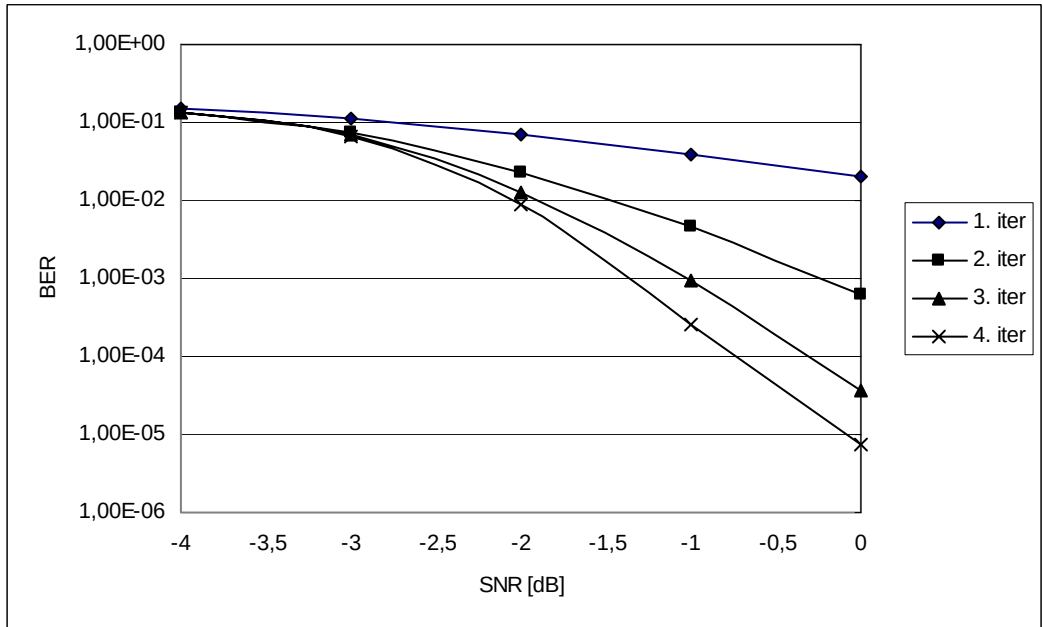
Bundan sonraki üç şekil, Şekil 3.78, Şekil 3.79 ve Şekil 3.80 çerçeve boyutu $N=5000$ olmak üzere 16-CPFSK modülasyonunda sırasıyla AWGN, $K=10\text{dB}$ ve $K=4\text{dB}$ Rician kanallar için bit hata oranını göstermektedir.



Şekil 3.78 16-CPFSK, $h=1/2$, $N=5000$ için AWGN ortamda TTC-CPFSK başarımı



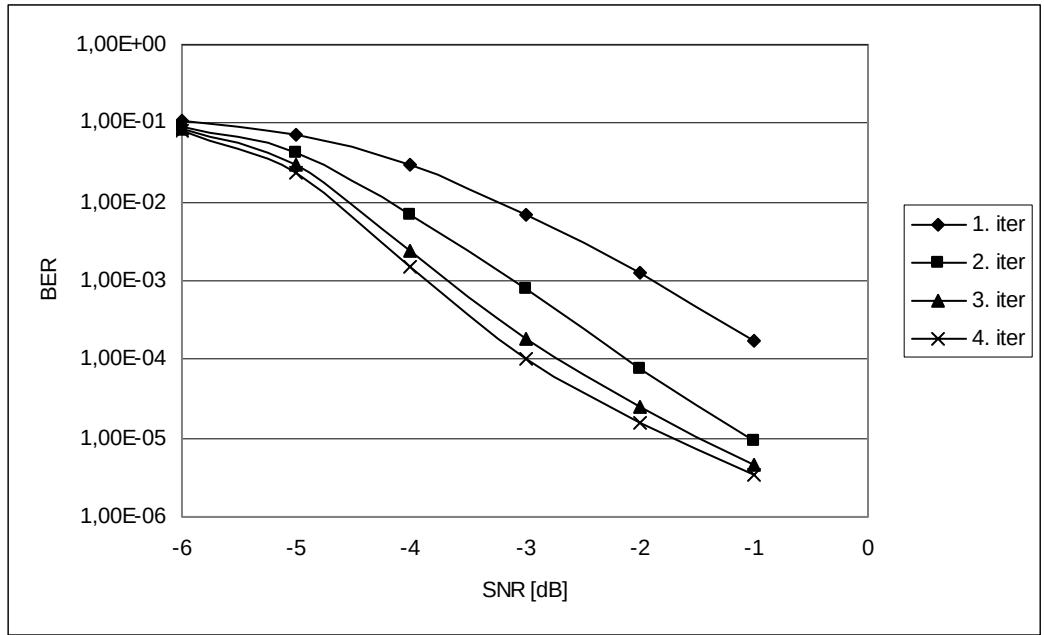
Şekil 3.79 16-CPFSK, $h=1/2$, $N=5000$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı



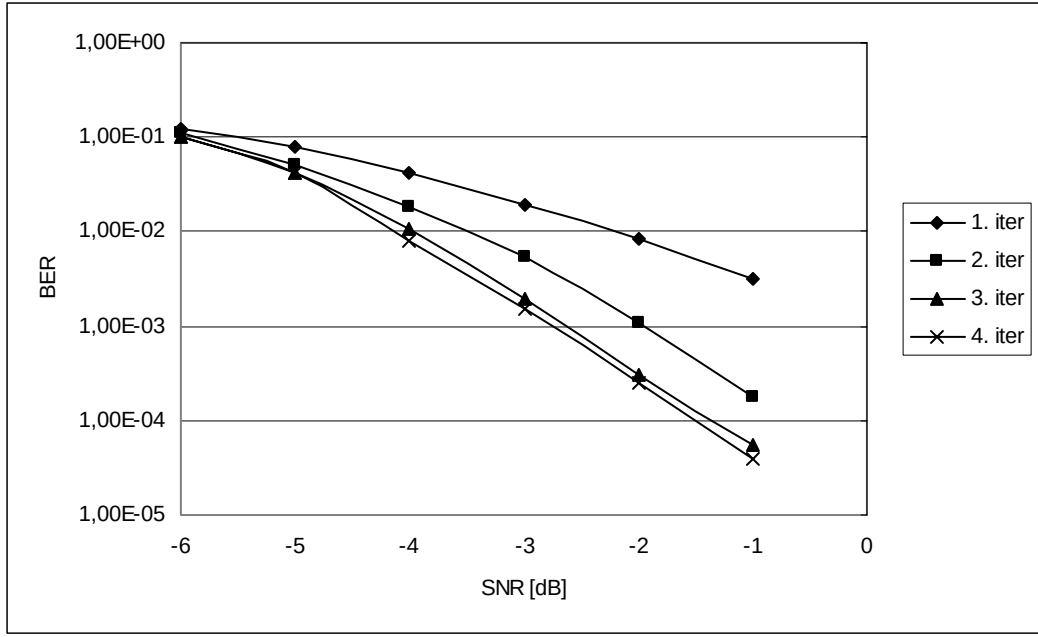
Şekil 3.80 16-CPFSK, $h=1/2$, $N=5000$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı

Bu sonuçlara bakıldığında, bit hata oranının çerçeve boyutunun 5000 olması durumunda, 1024 ve 256 olması durumlarına göre daha iyi olduğu görülmektedir. Aynı zamanda, daha önceki durumlarda olduğu gibi sönümleme etkisi arttıkça iterasyonun etkinliğinin arttığı görülmektedir.

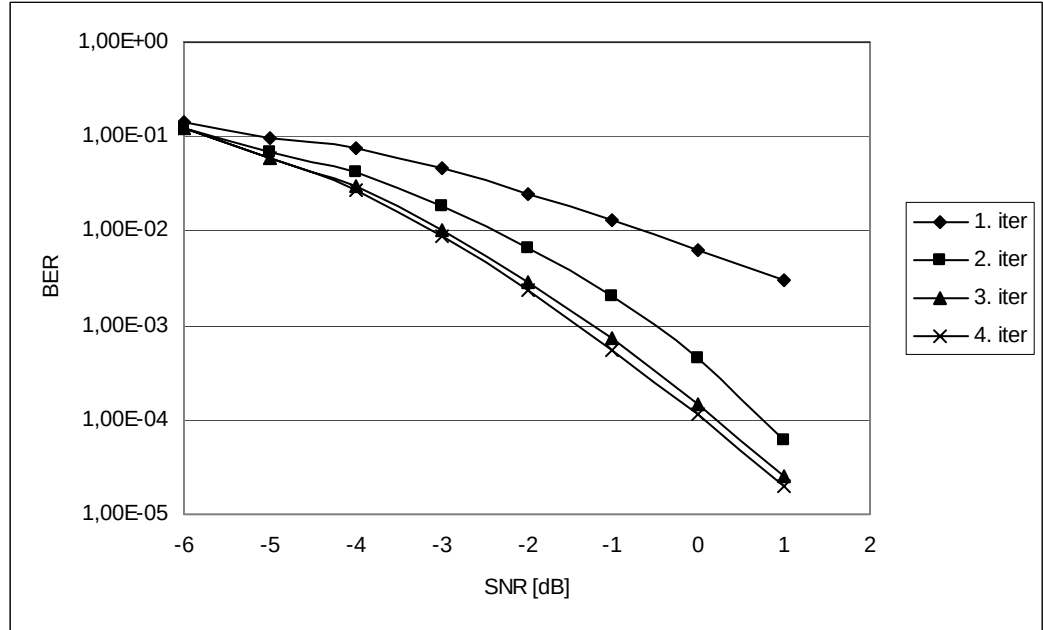
Bundan sonraki üç şekil, Şekil 3.81, Şekil 3.82 ve Şekil 3.83 çerçeve boyutu $N=256$ olmak üzere $h=1/4$ için 32-CPFSK modülasyonunda sırasıyla AWGN, $K=10\text{dB}$ ve $K=4\text{dB}$ Rician kanallar için bit hata oranını göstermektedir.



Şekil 3.81 32-CPFSK, $h=1/4$, $N=256$ için AWGN ortamda TTC-CPFSK başarımı



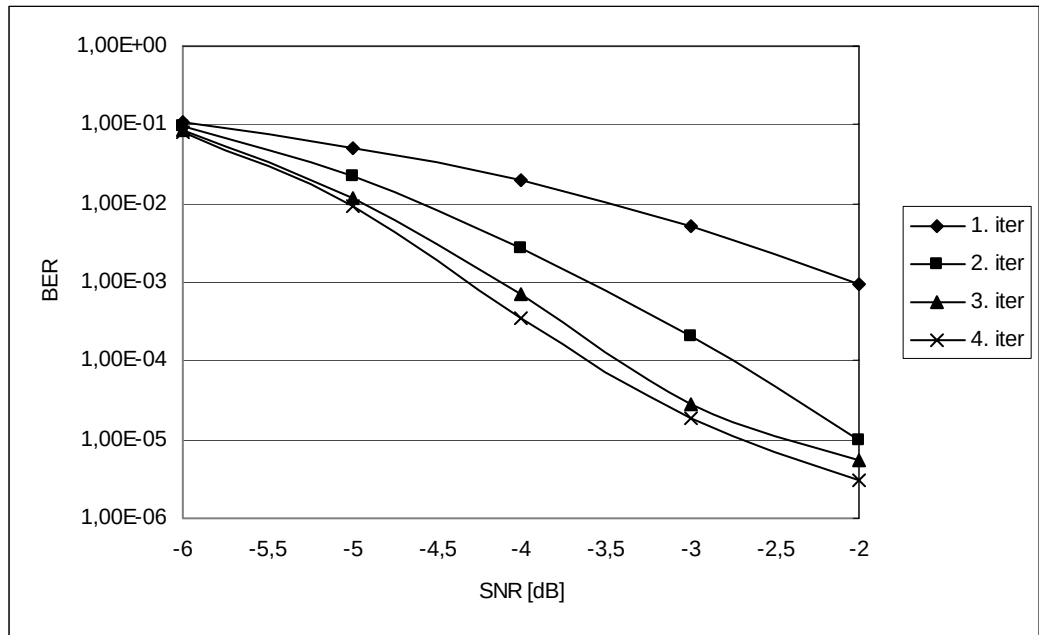
Şekil 3.82 32-CPFSK, $h=1/4$, $N=256$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı



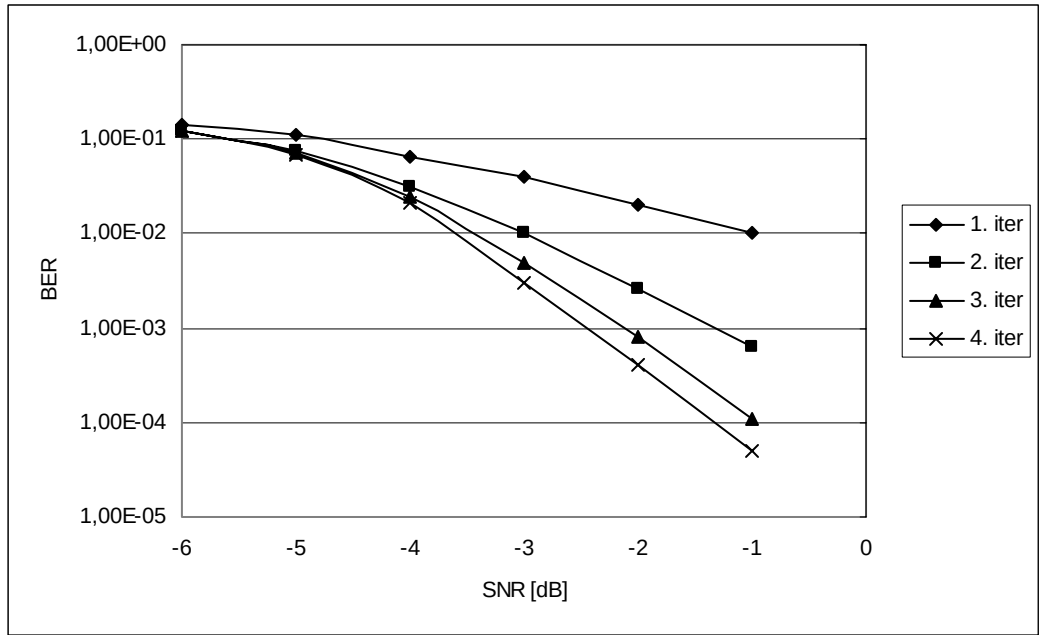
Şekil 3.83 32-CPFSK, $h=1/4$, $N=256$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı

Bu sonuçlara bakıldığında, bit hata oranının çerçeve boyutunun 265 olması ve $h=1/4$ için 32-CPFSK durumunda 16-CPFSK ya oranla daha iyi olduğu görülmektedir. Bunun sebebi, sinyal boyutunun ve bundan dolayı sinyaller arası serbest uzaklığın artmış olmasıdır. Her üç kanal ortamı için, üçüncü iterasyonun yeterli olduğu görülmektedir.

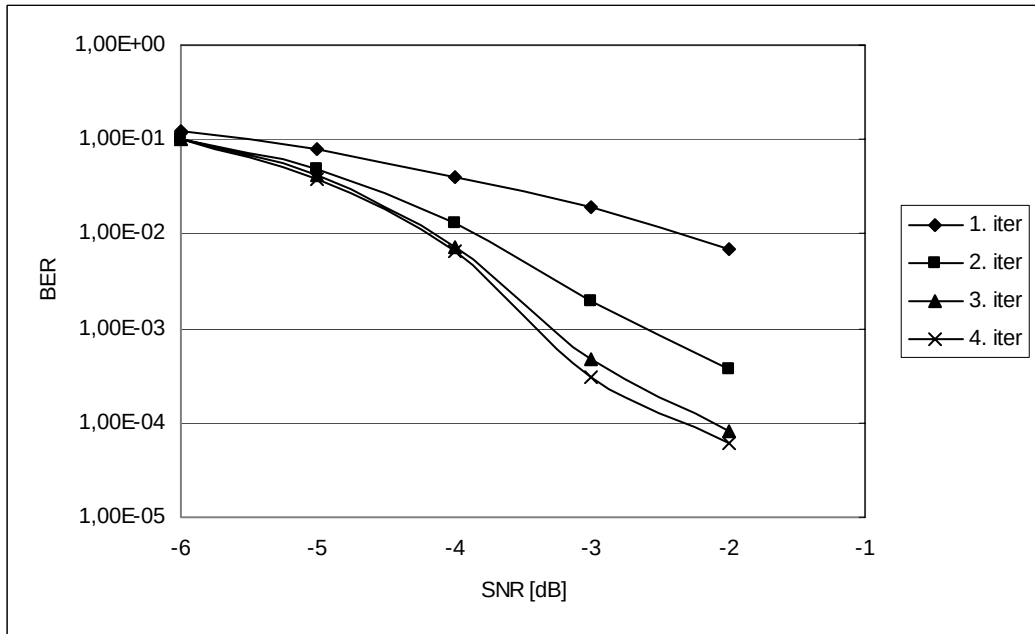
Bundan sonraki üç şekil, Şekil 3.84, Şekil 3.85 ve Şekil 3.86 çerçeve boyutu $N=1024$ olmak üzere $h=1/4$ için 32-CPFSK modülasyonunda sırasıyla AWGN, $K=10\text{dB}$ ve $K=4\text{dB}$ Rician kanallar için bit hata oranını göstermektedir.



Şekil 3.84 32-CPFSK, $h=1/4$, $N=1024$ için AWGN ortamda TTC-CPFSK başarımı



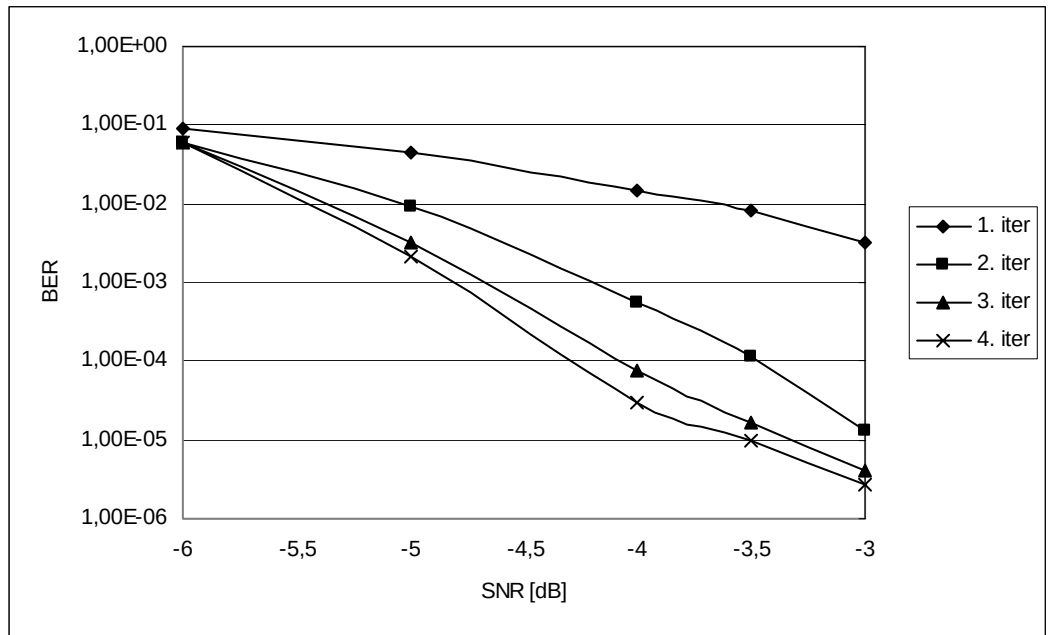
Şekil 3.85 32-CPFSK, $h=1/4$, $N=1024$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı



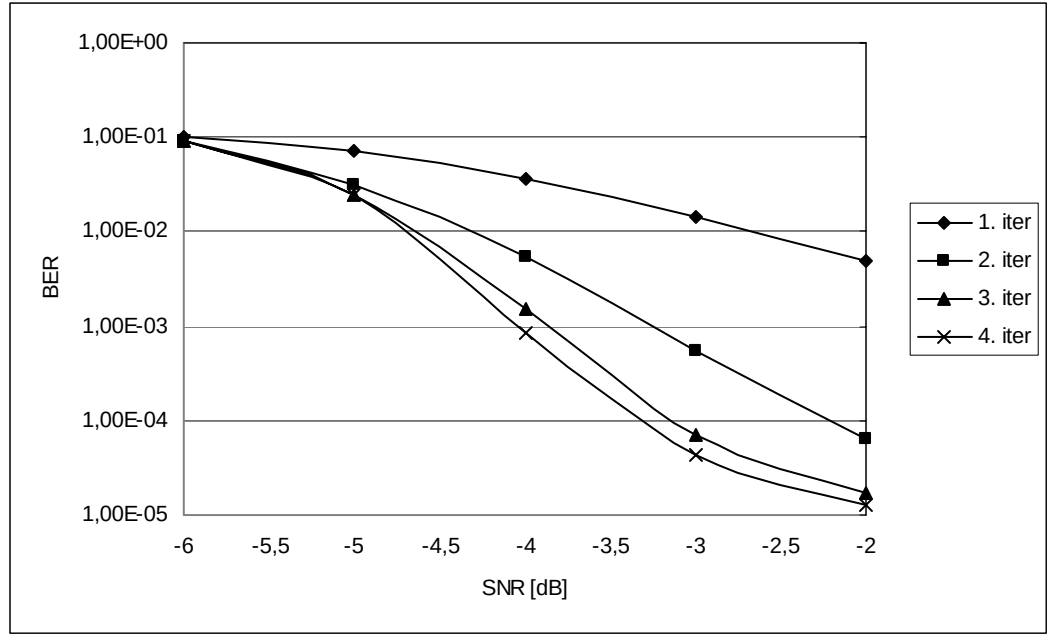
Şekil 3.86 32-CPFSK, $h=1/4$, $N=1024$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı

Bu sonuçlara bakıldığında, bit hata oranının, $h=1/4$ için 32-CPFSK ve çerçeve boyutunun 1024 olması durumunda, 256 olması durumuna ve $h=1/2$ için 16-CPFSK modülasyon kullanılma durumlarına göre daha iyi olduğu görülmektedir. Burada da ikinci ve üçüncü iterasyonlar yeterli görülmekte ve dördüncü iterasyona aslında gerek duyulmamaktadır.

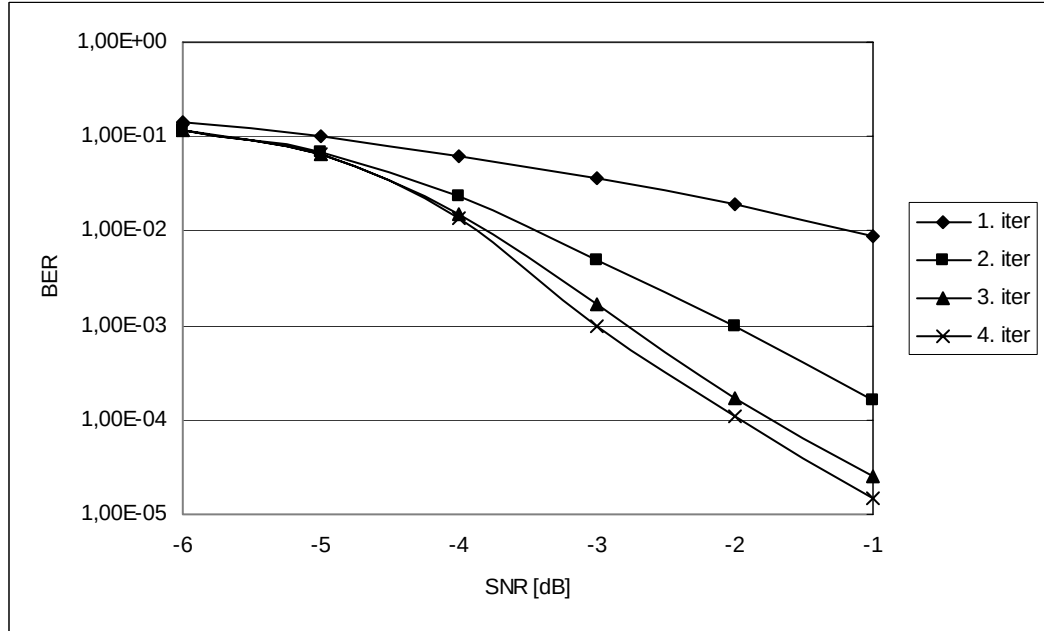
Bundan sonraki üç şekil, Şekil 3.87, Şekil 3.88 ve Şekil 3.89 çerçeve boyutu $N=5000$ olmak üzere $h=1/4$ için 32-CPFSK modülasyonunda sırasıyla AWGN, $K=10\text{dB}$ ve $K=4\text{dB}$ Rician kanallar için bit hata oranını göstermektedir.



Şekil 3.87 32-CPFSK, $h=1/4$, $N=5000$ için AWGN ortamda TTC-CPFSK başarımı



Şekil 3.88 32-CPFSK, $h=1/4$, $N=5000$ için $K=10$ dB Rician ortamda TTC-CPFSK başarımı



Şekil 3.89 32-CPFSK, $h=1/4$, $N=5000$ için $K=4$ dB Rician ortamda TTC-CPFSK başarımı

Simülasyon sonuçları incelendiğinde, çerçeve ve CPFSK'nın boyutu ile kanal sönümlleme etkisinin TTC-CPFSK'ya çok büyük etkisi vardır. Üçten fazla iterasyonun yapılmasının birçok durumda hemen hemen hiç bir yararının olmadığı görülmektedir.

TTC-CPFSK yöntemi ile klasik TCM yöntemi arasında hata başarımı bakımından oldukça büyük farklar vardır. TTC-CPFSK'nın en az 7-8 dB lik bir kazancının olduğu ve bu kazancın gerekirse CPFSK'nın boyutunu belirli bir oranda büyütürken daha da arttırılabileceği görülmektedir. Bunun sisteme getireceği yük ise sadece modülasyon ve demodülasyondaki karmaşıklık artmasıdır. Fakat karmaşıklıkta bu artış modülasyon ve demodülasyon süresinde herhangi bir gecikmeye neden olmaz.

Diğer bir taraftan band verimliliği, simülasyon sonuçlarına yansımamıştır. CPFSK PSK ve QAM gibi modülasyon yöntemlerindeki band verimliliği ve band sınırlı alıcılardan doğan bilgi kaybı bakımından da oldukça önemli avantajlara sahiptir. Bu modülasyon yöntemini Turbo kafes kodlama ile birleştirerek her ikisinin sağlamış olduğu yarardan faydalanılmış ve sonuçta hata başarımı ve band verimliliği bakımından çok iyi bir kodlama yöntemi ortaya konulmuştur. Bu sistemin her durumda ve ortamdaki performansının bu zamana kadar bulunan tüm kodlama yöntemlerinden çok daha iyi olduğu açıkça görülmektedir.

4. TARTIŞMA VE SONUÇ

Bu tezde Turbo Kodlama ve Turbo Kafes Kodlamalı Modülasyon yöntemleri açıklanmış olup Turbo Kafes Kodlamalı Modülasyon yöntemine ilişkin çeşitli kanal yapılarında kanal bilgisinin tam olarak bilindiği ve hiç bilinmediği durumlar için simülasyon sonuçları elde edilmiştir. Yine bir çok kanal için kanal parametreleri Baum-Welch (BW) algoritması ile kestirilmiş ve kestirilen kanal parametreleri kullanılarak kod çözme işlemi gerçekleştirilmiştir. Simülasyon sonuçları BW algoritması ile elde edilmiş sonuçların ideal sonuçlara çok yakın olduğunu göstermiştir. Buna karşın kod çözme işleminin karmaşıklığını arttırdığı için kod çözme sürecini uzattığı belirtilmiştir.

TTCM bu tezde OFDM ile birlikte kullanılmış çeşitli kanal yapıları, band genişliği ve alt-taşıyıcı sayısına göre simülasyon sonuçları elde edilmiş kanal parametreleri BW algoritması ile kestirilip Doppler kaymasının önemine değinilmiştir.

Bilindiği üzere sönümleme ve faz bozulması performansı en çok etkileyen sebeplerdir. Bu etkiler, çok anten kullanılarak en aza indirilebilmektedir. Bunun için en çok bilinen Uzay Zaman Blok Kodları (STBC) ve buna karşın önerilen Tekrarlamalı Uzay Zaman Kodları'nın (RSTC) her ikisi TTCM yapısı ile birlikte kullanılarak hata başarımları elde edilmiştir. Sonuç olarak önerilen yöntemin 0.6 dB daha iyi sonuç verdiği buna karşılık iletişim hızını azalttığı görülmektedir.

Tezin en önemli bölümü Turbo Kafes Kodlamalı - Sürekli Faz Frekans Kaydırmalı Anahtarlama (Turbo Trellis Coded - Continuous Phase Frequency Shift Keying, TTC-CPFSK) isimli yeni bir kod çözme yönteminin önerildiği bölümdür. TTCM çok etkin bir kodlama yöntemi olmasına karşın PSK ve QAM gibi sadece iki boyutlu işaretler oluşturabilir. Bu işaretlerin kanal verimliliği sürekli fazlı olan CPFSK'lı işaretlerden oldukça kötüdür. Ayrıca CPFSK çok boyutlu olduğu için simgeler arası uzaklık da fazladır bu da hata başarımında oldukça etkilidir. Tezde önerilen TTC-CPFSK isimli bu yeni kodlama yönteminin hata başarımı çeşitli kanallar için çok iyi sonuçlar vermiştir. Bu sebepten dolayı TTC-CPFSK'nın bit hata başarımı şu ana kadar bilinen sistemlerin hepsinden çok daha iyidir.

İleriki çalışmalarda, hata başarımlarını arttırmak için sürekli faz modülasyonunu kullanmak ve bunu başta çok antenli sistemler olmak üzere çeşitli sistemlere uygulamak çok iyi sonuç verecektir. Bunun yanında çok aşamalı kodlanmış sürekli faz modülasyonlu sistemlerin TTC-CPFSK'ya uygulanması performansın daha da iyileştirilmesini sağlayabilecektir.

KAYNAKLAR

- [1] SHANNON C. E., "A mathematical theory of communication," *Bell Sys. Tech. J.*, vol.27, Sayfa 379-423 ve 623-656, 1948.
- [2] HAMMING R.W., "Error detecting and correcting codes," *Bell Sys. Tech. J.*, vol. 29, Sayfa 147-160, 1950.
- [3] FORNEY G.D., "The Viterbi algorithm," *Proc. IEEE*, vol.61, Sayfa 268-278, Mart 1973.
- [4] UNGERBOECK G., "Channel coding with multi-level/phase signals", *IEEE Trans. IT-28*, Sayfa 56-67, 1982.
- [5] UNGERBOECK G., "Trellis coded modulation with redundant signal set part I," *IEEE Commun. Mag.*, Sayfa 5-11, 1987.
- [6] UNGERBOECK G., "Trellis coded modulation with redundant signal set part II," *IEEE Commun. Mag.*, Sayfa 15-21, 1987
- [7] WEI L.F., "Rotationary invariant convolutional channel coding with expanded signal space- Part 1:180°," *IEEE J. Selected Areas Commun.*, vol.SAC-2, no.5, Sayfa 659-671, 1984.
- [8] WEI L.F., "Rotationary invariant convolutional channel coding with expanded signal space- Part 2: Nonlinear codes," *IEEE J. Selected Areas Commun.*, vol.SAC-2, no.5, Sayfa 672-685, 1984.
- [9] WILSON S.G., et al., "Rate $\frac{3}{4}$ convolutional coding of 16-PSK: Code design and performance study," *IEEE Trans. Commun.*, vol. COM-32, Sayfa.1308-1315, 1984.
- [10] DIVSALAR D., SIMON M.K., "Multiple trellis coded modulation," *IEEE Trans. Inf. Theory*, vol.IT-36, no.4, Sayfa 410-419, 1988.
- [11] PITETROBON S.S., et al., "Trellis-coded multidimensional phase modulation," *IEEE Trans. Inf. Theory*, vol.IT-36, no.1, Sayfa 63-89, 1990.

- [12] DIVSALAR D., SIMON M.K. "Trellis coded modulation for 4800-9600 bits/s transmission over a fading mobile satellite channel," *IEEE J. Selected Areas Commun.*, vol.SAC-5, no.2, Sayfa 162-175, 1987.
- [13] SCHLEGEL C., COSTELLO D.J.Jr., "Bandwidth efficient coding for fading channels: Code construction and performance analysis," *IEEE J. Selected Areas Commun.*, vol.SAC-7, no.9, Sayfa 1356-1368, 1989.
- [14] DU J., VUCETIC B., "Trellis coded 16-QAM for fading channels," *ETT*, vol.4, no.3, Sayfa 101-107, 1993.
- [15] DU J., VUCETIC B., AHANG L., "Construction of new MPSK trellis codes for fading channels," *IEEE Trans. Commun.*, vol.43, no.2/3/4, Sayfa 776-784, February/March/April 1995
- [16] LEONARDO E.J., ZHANG L., VUCETIC B.S., "Multidimensional M-PSK trellis codes for fading channels," *IEEE Trans. Inf. Theory*, vol.IT-42, no.4, Sayfa 1093-1108, 1996.
- [17] BERROU C., GLAVIEUX A., THITIMASJSHIMA P., "Near Shannon-limit error correcting coding and decoding: Turbo codes," *Proc. ICC'93*, Sayfa 1064-1070, 1993.
- [18] BAHL L. R., COCKE J., JELINEK F., RAVIV J., "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Trans. Info. Theory*, vol IT-20, Sayfa 284-287, Mart 1974.
- [19] ERFANIAN J. A., PASUPATHY S., "Low-Complexity Parallel-Structure Symbol-by-Symbol Detection for ISI Channels," *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Sayfa 350-353, Haziran 1-2 1989.
- [20] BLACKERT W., WILSON S., "Turbo Trellis Coded Modulation," *Proc. CISS'96*
- [21] ROBERTSON P., WORZ T., "Bandwidth-efficient Turbo Trellis-coded modulation using punctured component codes," *IEEE JSAC*, vol.16, no.2, Sayfa 206-218, 1998.
- [22] BARBULESCU S. A., *Iterative Decoding of Turbo Codes and Other Concatenated Codes*, Doktora Tezi, 1996.

- [23] VALENTI M. C., *Iterative Detection and Decoding for Wireless Communications*, Doktora Tezi, 1999.
- [24] WU Y., *Design and Implementation of Parallel and Serial Concatenated Convolutional Codes*, Doktora Tezi, 1999.
- [25] GLOVER I.A., GRANT P.M., *Digital Communications*, Prentice Hall, 1998.
- [26] RODEN M.S., *Digital and Data Communication Systems*, Prentice Hall, 1982.
- [27] UÇAN O.N., *Kafes Kodlamalı-Dik Kısmi Yanıtlı Sistemlerin (QPR-TCM) Hata Başarım Analizi*, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 1994.
- [28] ODABAŞIOĞLU N., *Sürekli Faz Modülasyonlu Kaynak/Kanal Kodlamalı Sistemlerin Sönümlemeli Ortamlarda Hata Başarımı*, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 2002.
- [29] LI Y., VUCETIC B., SATO Y., “Optimum soft-output detection for channels with intersymbol interference,” *IEEE Trans. Inf. Theory*, no.41, Sayfa 704-713, Mayıs 1995.
- [30] ROBERTSON P., HOEHER P., VILLEBRUN E., “Optimum and sub-optimum a posteriori algorithms suitable for turbo decoding,” *European Trans. On Telecommunications*, no.8, Sayfa 119-125, 1997.
- [31] UÇAN O.N., OSMAN O., PAKER S., “Turbo Coded Signals Over Wireless Local Loop Environment,” *Int. Journal of Electronics and Commun.*, vol.56 no.2, Sayfa 1-6, 2002.
- [32] UÇAN O.N., OSMAN O., GÜMÜŞ A., “Performance of Turbo coded signals over partial response fading channels with imperfect phase reference,” in *Proc. IEEE 43.Midwest Symposium on Circuits and Systems, MWSCS'2000*, 04000-026, Michigan, Aug. (2000)..
- [33] UCAN O.N., OSMAN O., “Performance of turbo coded signals over partial response fading channel with imperfect phase reference”, *Istanbul University Journal of Electrical & Electronics Engineering*, vol:1, no:2, Sayfa 147-168, 2001.

- [34] UÇAN O.N., OSMAN O., “Turbo Trellis Coded Modulation (TTCM) with Imperfect Phase Reference”, *Istanbul University Journal of Electrical and Electronics*, vol:1, no.1, Sayfa 1-5, 2001
- [35] ERKURT M., Proakis J.G., “Joint data detection and channel estimation for rapidly fading channels,” in *Proc. IEEE Globecom*, Sayfa 910-914, Aralık 6-9, 1992.
- [36] BAUM L.E., PETRIE T., SOULES G., Weiss N., “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains,” *The Annals of Mathematical Statistics*, vol.41, Sayfa 164-171, 1970.
- [37] RABINER L.R., “A tutorial on hidden Markov models and selected applications in speech recognition,” in *Proceeding of the IEEE*, vol.77, no.2, Sayfa 257-274, 1989.
- [38] KALEH G. K., VALLET R., “Joint parameter estimation and symbol detection for linear or non linear unknown channels,” *IEEE Trans. on Communications*, vol.42, no.7, Sayfa 2406-2413, 1994.
- [39] UÇAN O.N., CIRPAN H.A., “Blind Equalisation of Quadrature Partial Response-Trellis Coded Modulated Signals,” Chapter of Third Generation Mobile Telecommunications. *Springer Verlag*, 2001.
- [40] VAN de BEEK J., SANDELL M., BORJESSON P., “ML Estimation of time and frequency offset in OFDM systems,” *IEEE Trans. Signal Processing*, vol.45, no.7, 1800-1805, 1997.
- [41] LI Y., “Pilot-symbol-aided channel estimation for OFDM in wireless systems,” *IEEE Trans. Vehicular Technology*, vol.49, no.4, Sayfa 1207-1215, 2000.
- [42] BENEDIX M., ERTEL J., FINGER A., “Turbo Coding for an OFDM-based wireless LAN at 17 GHz,” *7th International OFDM-Workshop*, Sayfa 137-141, Hamburg Eylül 2002.
- [43] LU B., WANG X., LI Y., “Iterative Receiver for Space-Time Block-Coded OFDM Systems in Dispersive Fading Channels,” *IEEE Trans. on Wireless Comm.*, vol.1, no.2, Sayfa 213-225, Nisan 2002.
- [44] BAHAI A. R., SALTZBERG B. R., *Multi-carrier Digital Communications Theory and Applications of OFDM*, Kluwer Academic/Plenium Publishers, Newyork 1999.

- [45] OCHIAI H., IMAI H., "Performance of the Deliberate Clipping with Adaptive Symbol Selection for Strictly Band-Limited OFDM Systems," *IEEE Journal Selected Areas*, vol. 18, no.11, Sayfa 2270-2277, 2000.
- [46] WEINSTEIN S. B., ELBERT P.M., "Data transmission by frequency division multiplexing using the discrete fourier transmission," *IEEE Trans. Comm.*, vol.COM-19, Sayfa 628-634, Ekim 1971.
- [47] ZOU W. Y., WU Y., "COFDM: an overview," *IEEE Trans. Comm.*, vol.41, no.1, Sayfa 1-8,Mart 1995.
- [48] CHOW P. S., TU J. C., CIOFFI J. M., "A Discrete Multitone Transceiver System for HDSL Applications," *IEEE J. Selected Areas in Comm.*, vol.SAC-9, no.6, Sayfa 895-908, Ağustos 1991.
- [49] CHOW P. S., TU J. C., CIOFFI J. M., "Performance Evaluation of a Multichannel Transceiver System for ADSL and VHDSL services," *IEEE J. Selected Areas in Comm.*, vol. SAC-9, no. 9, Sayfa 909-919, Ağustos 1991.
- [50] PAIEMENT R. V., "Evaluation of Single Carrier and Multicarrier Modulation Techniques for Digital ATV Terrestrial Broadcasting," *CRC Report*, no. CRC-RP-004, Ottawa, Canada, Aralık 1994.
- [51] SARI H., KARMA G., JEANCLAUDE I., "Transmission Techniques for Digital Terrestrial TV Broadcasting," *IEEE Comm. Mag.*, vol 33, Sayfa 100-109, Şubat 1995.
- [52] OPPENHEIM A. V., SCHAFFER R. W., *Discrete-time Signal Processing*, Prentice-Hall International, ISBN 0-13-216771-9, 1989.
- [53] HARA S., MOURI M., OKADA M., MORINAGA N., "Transmission Performance Analysis of Multi-Carrier Modulation in Frequency Selective Fast Rayleigh Fading Channel," *Wireless Personal Comm.*, Kluwer Academic Publishers, vol.2, Sayfa 335-356, 1996.
- [54] ALAMOUTİ S.M., "A Simple Transmit Diversity Technique for wireless Communication" *IEEE Journal of Selected Areas in Communications*,no.8, Sayı 16, Ekim 1998.

- [55] TAROKH V., SESHADRI N., CALDERBANK A., "Space-time codes for high data rate wireless communication: Performance criterion and code construction" *IEEE Transaction on Information Theory*, no.44, sayfa 744-765, Mart 1998.
- [56] TAROKH V., JAFARKHANI H., CALDERBANK A., "Space-time codes from orthogonal design" *IEEE Transaction on Information Theory*, Haziran 1999.
- [57] BAUCH G., "Concatenation of Space-Time Block Codes and "Turbo"-TCM", *Proc. IEEE International Conference on Communications*, sayfa 1202-1206, Vancouver, Canada, 1999.
- [58] OSMAN O., "Performance of Antenna Diversity with Turbo Trellis Coded Modulation Over Nakagami Channels," *Istanbul University Journal of Electrical & Electronics Engineering*, vol.2, no.1, Sayfa 369-375, 2002.
- [59] UÇAN O.N., "Concatenation of Space-Time Block Codes and Turbo Trellis Coded Modulation (ST-TTCM) Over Rician Fading Channels With Imperfect Phase Reference," *Istanbul University Journal of Electrical & Electronics Engineering*, vol.2, no.1, Sayfa , 2002.
- [60] RIMOLDI B.E., "A decomposition approach to CPM," *IEEE Trans. Inform. Theory*, vol. 34, Sayfa 260–270, Mart 1988.
- [61] RIMOLDI B.E., "Design of coded CPFSK modulation systems for band-width and energy efficiency," *IEEE Trans. Commun.*, vol. 37, Sayfa 897–905, Eylül 1989.
- [62] YANG R. H. H., TAYLOR D.P., "Trellis-coded continuous-phase frequency-shift keying with ring convolutional codes," *IEEE Trans. Inform. Theory*, vol. 40, Sayfa 1057–1067, Temmuz 1994.
- [63] RIMOLDI B.E., LI Q., "Coded continuous phase modulation using ring convolutional codes," *IEEE Trans. Commun.*, vol. 43, Sayfa 2714–2720, Kasım 1995.
- [64] ABRISHAMKAR F., BIGLIERI E., "Suboptimum detection of trellis-coded CPM for transmission on bandwidth- and power-limited channels," *IEEE Trans. Commun.*, vol. 39, Sayfa 1065–1074, Temmuz 1991.

- [65] KERR R.W., MCLANE P.J., "Coherent detection of interleaved trellis encoded CPFSK on shadowed mobile satellite channels," *IEEE Trans. Veh. Technol.*, vol. 41, Sayfa 159–169, Mayıs 1992.
- [66] NARAYANAN K.R., "Iterative demodulation and decoding of trellis coded CPM," in *Proc. of the 1999 IEEE Military Communications Conf.*, Sayfa 1302–1306, Ekim/Kasım 1999
- [67] BRUTEL C., BOUTROS J., "Serial concatenation of interleaved convolutional codes and M-ary continuous phase modulations," *Annales des T' el'e communications*, vol. 54, Sayfa 235–242, Mart/Nisan 1999.
- [68] SZETO V.F., PASUPATHY S., "Iterative decoding of serially concatenated convolutional codes and MSK," *IEEE Commun. Lett.*, vol. 3, Sayfa 272– 274, Eylül 1999.
- [69] SHANE M. R., WESEL R. D., "Parallel concatenated turbo codes for continuous phase modulation," in *Proc. of WCNC*, Sayfa 147-152, Eylül 2000.
- [70] PIZZI S. V., WILSON S. G., "Convolutional Coding Combined With Continuous Phase Modulation," *IEEE Transaction on Communications*, vol. 33, Ocak, 1985.
- [71] AULIN T., SUNDBERG C. E., "Continuous phase modulation- Part2: Partial response signaling," *IEEE Trans. Comm.* vol. COM-9, no.3, Sayfa 210-225, Mart 1981.
- [72] ALTUNBAŞ İ., AYGÖLÜ Ü., "Design and performance analysis of multilevel coded M-ary CPFSK," *Proc. Of Int. Telecom. Conf.*, ICT'96 Sayfa 269-272, İstanbul Nisan 1996

ÖZGEÇMİŞ

Onur Osman 25 Eylül 1973 yılında İstanbul Türkiye’de doğdu. 1994 yılında İstanbul Teknik Üniversitesi (İTÜ) Elektrik Mühendisliği Güç Elektronikliği paketinden mezun oldu. Aynı yıl Yüksek Lisans programına giren Onur Osman “Relüktans Motor Benzetimi ve TMS370 Mikrokontrolör ile Kontrolü” tezi ile 1998 yılında Yüksek Mühendis ünvanını aldı. Bu tarihten itibaren İstanbul Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği bölümünde doktora yapmaya başladı. 1998-1999 yılları arasında Universal İletişim ve Bilgisayar Hizmetleri ve Tic. Ltd. Şti.’de araştırma-geliştirme mühendisi olarak çalıştı. 1999-2001 arasında Yeditepe Üniversitesi’nde araştırma görevlisi ve daha sonra öğretim görevlisi olarak çalıştı. Şu anda İstanbul Ticaret Üniversitesi Meslek Yüksek Okulu Bilgisayar Teknolojisi ve Programlama bölümünde öğretim görevlisi olarak çalışmaktadır.

Onur Osman’ın çalışma konuları; haberleşme sistemleri, kablosuz haberleşme, Turbo kodlama, kanal yapıları ve etkileri, kanal parametrelerinin kestirimi ve Saklı Markov Model (HMM), uzay-zaman kodlaması, mikrodalga kanallar, biomedical ve jeofizik uygulamaları, ağ yapıları, yapay zeka, resim işleme, ses işleme, mikrokontrolörler ve servo motor kontrolü.