



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

**SÜREKLİ FAZ MODÜLASYONUNUN ÇOK DÜZEYLİ
TURBO KODLANMASI**

Yük. Müh. Niyazi ODABAŞIOĞLU
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman
Prof. Dr. Osman Nuri UÇAN

Aralık, 2005

İSTANBUL



**İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

**SÜREKLİ FAZ MODÜLASYONUNUN ÇOK DÜZEYLİ
TURBO KODLANMASI**

Yük. Müh. Niyazi ODABAŞIOĞLU
Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman
Prof. Dr. Osman Nuri UÇAN

Aralık, 2005

İSTANBUL

Bu alıřma, 21.02.2006 tarihinde ařaęıdaki jüri tarafından Elektrik Elektronik Mühendislięi Anabilim Dalı Elektrik Elektronik Mühendislięi programında Doktora Tezi olarak kabul edilmiřtir.

Tez Jürisi

Danıřman
Prof. Dr. Osman Nuri UAN
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof. Dr. Mahmut ÜN
İstanbul Üniversitesi
Mühendislik Fakültesi

Prof. Dr. Serhat řEKER
İstanbul Teknik Üniversitesi
Elektrik Elektronik Fakültesi

Prof. Dr. Metin YÜCEL
Yıldız Teknik Üniversitesi
Elektrik Elektronik Fakültesi

Do. Dr. Hakan Ali IRPAN
İstanbul Üniversitesi
Mühendislik Fakültesi

ÖNSÖZ

Öncelikle, yüksek lisans ve doktora öğrenimim boyunca bana her zaman yol gösteren, desteğini ve yardımlarını hiç esirgemeyen değerli hocam Sayın Prof. Dr. Osman Nuri Uçan’a sonsuz teşekkürlerimi sunuyorum. Tezin oluşum aşamasında çoğu zaman birlikte çalıştığım İstanbul Ticaret Üniversitesi Öğretim Görevlisi Sayın Dr. Onur Osman’a teze yaptığı büyük katkılarından dolayı teşekkür ederim. Ayrıca tez çalışmam boyunca bana hep yardımcı olan, mesai arkadaşlarım, İstanbul Üniversitesi Elektrik Elektronik Mühendisliği Araştırma Görevlileri Yük. Müh. Niyazi Kılıç ve Yük. Müh. Yeşim Hekim’e teşekkürü bir borç biliyorum. Son olarak hayatımın her aşamasındaki manevi desteklerinden dolayı aileme teşekkür ederim.

İstanbul, Aralık 2005

Niyazi Odabaşoğlu

*Bu çalışma İstanbul Üniversitesi Bilimsel Araştırma Projeleri Yürütücü Sekreterliği tarafından T-580/17032005 numaralı proje kapsamında desteklenmiştir.

İÇİNDEKİLER

ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİL LİSTESİ.....	iv
TABLO LİSTESİ	vii
ÖZET	viii
SUMMARY	x
1. GİRİŞ	1
1.1 TEZDE İZLENEN YOL.....	6
2. GENEL KISIMLAR	8
2.1. KANAL KAPASİTESİ VE KANAL KODLAMA.....	8
2.1.1. Blok Kodlar	9
2.1.1.1 Kod Uzaklığı	9
2.1.1.2 Kod Uzunluğu	10
2.1.2. Katlamalı Kodlar	10
2.1.3. Geri Beslemeli SistematiK Katlamalı Kodlar.....	13
2.2. KODLAMALI MODÜLASYON TEKNİKLERİ	15
2.2.1. Kafes Kodlamalı Modülasyon	16
2.2.1.1. Kafes Kodlu Modülasyonda Kodçözme	20
2.2.2. Çok Düzeyli Kodlama	22
2.3. TURBO KODLAMA	25
2.3.1. Turbo Kodlayıcı.....	25
2.3.2. Turbo Kod Çözme	27
2.3.2.1. Yumuşak-Giriş Yumuşak-Çıkış (Soft-Input Soft-Output) Kodçözme.....	27
2.3.2.2. En Büyük İleriye Dönük Algoritma.....	30
2.3.2.3. Max-log-MAP ve log-MAP Algoritmaları	33
2.3.3. Turbo Kafes Kodlamalı Modülasyon	35
2.3.3.1. Turbo Kafes Kodlayıcı	35

2.3.3.2. Turbo Kafes Kodçözücü.....	37
2.4. SÜREKLİ FAZ MODÜLASYONU.....	44
2.4.1. CPM Tekniğinin Tanıtılması	46
2.4.2. Sürekli Faz Modülasyonu İçin Ayırıştırma Tekniği	48
2.4.3. Sürekli Fazlı Frekans Kaydırmalı Anahtarlama (CPFSK)	49
2.4.3.1. 4'lü CPFSK , $h=1/2$	54
2.4.3.2. 8'li CPFSK , $h=1/2$	56
2.4.3.3. 16'lı CPFSK , $h=1/2$	
3. MALZEME VE YÖNTEM	62
3.1. SÜREKLİ FAZ MODULASYONUN ÇOK DÜZEYLİ TURBO KODLANMASI.....	62
3.1.1. Kodlayıcı Yapısı	62
3.1.2. Kodçözücü Yapısı.....	63
3.1.3. İki Düzeyli Turbo Kodlanmış 4'lü CPFSK Sistemi	66
3.1.4. Üç Düzeyli Turbo Kodlanmış 8'li CPFSK Sistemi	71
4. BULGULAR	76
4.1. SÖNÜMLEMELİ KANALLAR	76
4.2. ÖRNEK SİSTEMLERİN HATA BAŞARIMLARI	78
5. TARTIŞMA VE SONUÇ	89
KAYNAKLAR	92
ÖZGEÇMİŞ	97

ŞEKİL LİSTESİ

Şekil 1.1	: Sayısal bir iletişim sisteminin blok diyagramı	2
Şekil 2.1	: Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcı.....	11
Şekil 2.2	: Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcının durum diyagramı	12
Şekil 2.3	: Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcının kafes diyagramı	13
Şekil 2.4	: Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcı.....	14
Şekil 2.5	: Kodlama oranı $r = 1/2$ ve $K_c=3$ için RSC kodlayıcı durum diyagramı.	15
Şekil 2.6	: Kodlama oranı $r = 1/2$ ve $K_c=3$ için RSC kodlayıcı kafes diyagramı...	15
Şekil 2.7	: Kafes kodlamalı modülasyon blok şeması	17
Şekil 2.8	: 8-PSK işaret kümesinin ayrıştırılması	17
Şekil 2.9a	: 4 Durumlu kafes diyagramı	18
Şekil 2.9b	: Bitlerin eşlenmesi	18
Şekil 2.10	: Olası minimum uzaklıklı iki yol.....	19
Şekil 2.11	: Kafes kodlamalı modülasyon şeması	20
Şekil 2.12	: Genel bir çok düzeyli kodlayıcı yapısı	22
Şekil 2.13	: Genel bir çok aşamalı kodçözücü yapısı	23
Şekil 2.14	: Turbo Kodlayıcı.....	25
Şekil 2.15	: 2 bellekli Turbo Kodlayıcı.....	26
Şekil 2.16	: $1/2$ oranlı RSC kodlayıcı için yumuşak-giriş yumuşak-çıkış (SISO) kod çözücü	28
Şekil 2.17	: BPSK Modülatör	28
Şekil 2.18	: Turbo Kod Çözücü	30
Şekil 2.19	: Turbo Kafes Kodlayıcısı.....	36
Şekil 2.20	: Kodlayıcı girişi ve eşleyici çıktısı	37
Şekil 2.21	: TTCM Kod Çözücü.....	37
Şekil 2.22	: Faz modülasyon fonksiyonu	45
Şekil 2.23	: Faz kafesi.....	46

Şekil 2.24	: CPM işaretlerinin üretilmesi.....	47
Şekil 2.25	: CPE'nin blok diyagramı	48
Şekil 2.26a	: CPFSK'ya ait $f(t)$	49
Şekil 2.26b	: CPFSK'ya ait $f(t)$ 'nin integrali	49
Şekil 2.27	: İkili CPFSK faz ağacı.....	50
Şekil 2.28	: FFSK' nın fiziksel faz kafes diyagramı	51
Şekil 2.29	: M'li CPFSK, $h=J/P$ 'nin fiziksel eğik faz kafesi (işaret kümesi)	53
Şekil 2.30	: 4'lü CPFSK, $h=1/2$ 'nin işaret kümesi	55
Şekil 2.31	: 8'li CPFSK, $h=1/2$ 'nin işaret kümesi	57
Şekil 2.32	: 16'lı CPFSK $h=1/2$ ' nin işaret kümesi	61
Şekil 3.1	: Sürekli faz modülasyonunun çok düzeyli turbo kodlanmasına ilişkin blok diyagram	63
Şekil 3.2	: 2 düzeyli turbo kodlanmış 4'lü CPFSK sistemi için kodlayıcı yapısı..	67
Şekil 3.3	: 2 düzeyli turbo kodlanmış 4'lü CPFSK sistemi için kodçözücü yapısı	67
Şekil 3.4	: 4'lü CPFSK modülasyonuna ilişkin küme bölmeleme işlemi.....	68
Şekil 3.5	: Üç düzeyli turbo kodlanmış 8'li CPFSK sistemi için kodlayıcı yapısı	72
Şekil 3.6	: Üç düzeyli turbo kodlanmış 8'li CPFSK sistemi için kodçözücü yapısı	73
Şekil 3.7	: 8'li CPFSK modülasyonuna ilişkin küme bölmeleme işlemi.....	74
Şekil 4.1	: İki düzeyli turbo kodlanmış 4PSK sisteminde $N=100$ için hata başarımlar eğrileri	79
Şekil 4.2	: İki düzeyli turbo kodlanmış 4'lü CPFSK sisteminde $N=100$ için hata başarımlar eğrileri	79
Şekil 4.3	: İki düzeyli turbo kodlanmış 4PSK sisteminde $N=1024$ için hata başarımlar eğrileri	80
Şekil 4.4	: İki düzeyli turbo kodlanmış 4'lü CPFSK sisteminde $N=1024$ için hata başarımlar eğrileri	80
Şekil 4.5	: Üç düzeyli turbo kodlanmış 8PSK sisteminde $N=100$ için hata başarımlar eğrileri	82
Şekil 4.6	: Üç düzeyli turbo kodlanmış 8'li CPFSK sisteminde $N=100$ için hata başarımlar eğrileri	82
Şekil 4.7	: Üç düzeyli turbo kodlanmış 8PSK sisteminde $N=1024$ için hata başarımlar eğrileri	83

Şekil 4.8	: Üç düzeyli turbo kodlanmış 8’li CPFSK sisteminde $N=1024$ için hata başarım eğrileri	83
Şekil 4.9	: 2,4,8 ve 16’lı CPFSK modülasyonları için kanal kapasitesi eğrileri.....	87

TABLO LİSTESİ

Tablo 2.1	: Durum Sayısı-Kodlama Kazancı	20
Tablo 2.2	: 4'lü CPFSK, $h=1/2$ değerleri	55
Tablo 2.3	: 8'li CPFSK, $h=1/2$ değerleri	56
Tablo 2.3	: 8'li CPFSK, $h=1/2$ değerleri (Devam).....	57
Tablo 2.4	: 16'lı CPFSK, $h=1/2$ değerleri	58
Tablo 2.4	: 16'lı CPFSK, $h=1/2$ değerleri (Devam).....	59
Tablo 2.4	: 16'lı CPFSK, $h=1/2$ değerleri (Devam).....	60
Tablo 2.4	: 16'lı CPFSK, $h=1/2$ değerleri (Devam).....	61
Tablo 3.1	: İki düzeyli turbo kodlanmış 4'lü CPFSK sistemi için kodlayıcı çıkışı- 4'lü CPFSK işaret eşlemeleri	69
Tablo 3.2	: Üç düzeyli turbo kodlanmış 8'li CPFSK sistemi için kodlayıcı çıkışı- 8'li CPFSK işaret eşlemeleri	75
Tablo 4.1	: Örnek olarak tasarlanan çok düzeyli turbo kodlanmış CPFSK sistemlerinin kafes kodlanmış CPFSK sistemlerine olan kodlama kazançları	85

ÖZET

SÜREKLİ FAZ MODÜLASYONUNUN ÇOK DÜZEYLİ TURBO KODLANMASI

Bu tezde, turbo kodların çok düzeyli kodlanmasına ilişkin bir yöntem geliştirilerek sürekli faz modülasyonunun geliştirilen bu yöntemle kodlanması önerilmiştir. Önerilen yöntem için örnek sistemler tasarlanmış ve bu örnek sistemlerin hem toplamsal beyaz gaus gürültülü (AWGN) kanallar hem de sönmülemeli kanallardaki hata başarımları incelenmiştir.

Turbo kodlar literatürde şu ana kadar bulunmuş en verimli hata düzeltme kodlarından biridir. Turbo kodlar büyük boyutlu karıştırıcılar ve oldukça basit alt kodlayıcılar kullanarak Shannon limitine çok yakın bir performans gösterirler. Turbo kodların en önemli özelliği Claude Shannon tarafından öngörülen teorik limite oldukça yakın, güç verimliliği yüksek güvenilir bir iletişim sağlamasıdır. Büyük boyutlu serpiştirici kullanma ve kodçözmede çok sayıda iterasyon gerekliliği ise turbo kodların en önemli dezavantajlarıdır. Tezde, turbo kodlara ait bu dezavantajların giderilmesi için, turbo kodların çok düzeyli kodlama tekniğiyle kodlanması üzerinde durulmuştur. Çok düzeyli kodlama tekniği, kolay kodlayıcı tasarımlarına imkan veren ve kod çözme karmaşıklığını arttırmadan güç kazancı sağlayan önemli bir kanal kodlama tekniğidir. Çok düzeyli kodlayıcı, birkaç hata düzeltme kodunun kombinasyonundan oluşmaktadır. Çok düzeyli kodlamada kanala iletilecek işareti oluşturmak için her bir işaretleme aralığında hata denetleme kodlayıcılarının bir ya da daha fazla çıkış biti kullanılabilir. Tezde çok düzeyli kodlayıcının her bir düzeyinde bir turbo kodlayıcı kullanılmış böylece hem kodlayıcı tasarımları kolayca yapılmış hem de kod çözme karmaşıklığı arttırılmadan turbo kodların kodlama kazancı daha da arttırılmıştır.

Sürekli Faz Modülasyonu (SFM) sabit zarf, band ve güç verimliliği özelliklerinden dolayı güç ve band sınırlı kanallar için oldukça uygun bir modülasyon yöntemidir.

SFM bilinen faz modülasyonunun sürekli fazlı biçimidir ve faz sürekliliğinden dolayı radyo spektrumunu oldukça verimli kullanır. Bütün bu üstün özelliklerinden dolayı bu tezde sürekli faz modülasyonunun tasarlanan kodlama kazancı yüksek çok düzeyli turbo kodlarla kodlanmasına ilişkin bir yöntem geliştirilmiştir. Sürekli Faz modülasyonunun kodlanmasında en önemli kısıtlamayı kanala iletilecek işaretlerin sürekli faza sahip olma zorunluluğu oluşturur. Tezde önerilen yöntemde sürekli faz şartının sağlanması için çok düzeyli kodlayıcının son düzeyinde faz bilgisini tutan sürekli faz kodlayıcısı kullanılarak bu şart sağlanmıştır. Böylece sürekli faz modülasyonu, sürekli faz özelliği bozulmadan çok düzeyli turbo kodlarla kodlanarak hem iletim bandını verimli kullanan hem de kodçözme karmaşıklığını arttırmadan yüksek kodlama kazancına sahip sistemler oluşturulmuştur.

Ayrıca önerilen sisteme örnek olması açısından iki adet sistem tasarımı yapılmıştır. Bu sistemler iki düzeyli turbo kodlanmış 4'lü CPFSK sistemi ve üç düzeyli turbo kodlanmış 8'li CPFSK sistemidir. Bu sistemlerin AWGN ve sönmülemeli kanallardaki hata başarımlar eğrileri çıkarılmış literatürde bulunan benzer sistemlerle karşılaştırılarak hata başarımlar üstünlükleri ortaya konulmuştur.

SUMMARY

MULTILEVEL TURBO CODED CONTINUOUS PHASE MODULATION

In this thesis, multilevel turbo codes are first introduced and then applied to continuous phase modulation. Thus, multilevel turbo coded continuous phase modulation systems (MLTC-CPM) are also introduced. Performance issues of the proposed schemes are explored based on the evaluation of Bit-Error rate (BER) bounds for both additive white Gaussian channel (AWGN) and fading channels as well.

Turbo codes are one of the most efficient error control codes that has ever been proposed in the literature till now. The performance of these codes is close to the Shannon-limit with relatively simple component codes and large size interleavers. The importance of turbo codes is that they enable reliable communications with power efficiencies close to the theoretical limit predicted by Claude Shannon. Classical turbo codes require large frame sizes and high number of iterations at receiver side to obtain better bit error rates. These requirements bring two major disadvantages of turbo codes together. To eliminate these disadvantages, multilevel coding (MLC) technique is used together with turbo codes. Multilevel coding is one of the most important channel coding technique which makes more simple encoder design process possible and provides power efficiency without increasing decoding complexity. Multilevel encoder consists of several error control encoder combinations. Multilevel coding scheme employs, at each signalling interval, one or more output bits of each of several error control encoders. In this study, one turbo code is used in every level of multilevel encoder, thus encoder is simply designed without increasing the complexity of decoding and the coding gain of turbo codes are more increased.

Continuous Phase Modulation (CPM) is very convenient for mobile radio and satellite communication because of its constant envelop property, power and spectral efficiency. CPM is the continuous phase version of known phase modulation and via its phase

continuity it uses radio spectrum highly efficiently. Due to its superior properties mentioned above, in this thesis, a new coding technique for continuous phase modulation by using multilevel turbo code is designed and the whole system is called “Multivel Turbo Coded Continuous Phase Modulation”. The most important constraint in encoding of continuous phase modulation is the fact that the signals which are transmitted to the channel must have continuous phase. In the method proposed in this study, this constraint is fullfilled by using continuous phase encoder which keeps phase information at the last level of multilevel encoder. Hence, continuous phase modulation is encoded with multilevel turbo codes without loosing its continuous phase properties to get systems which not only uses trasmission band efficiently but also has higher coding gain without increasing the decoding complexity.

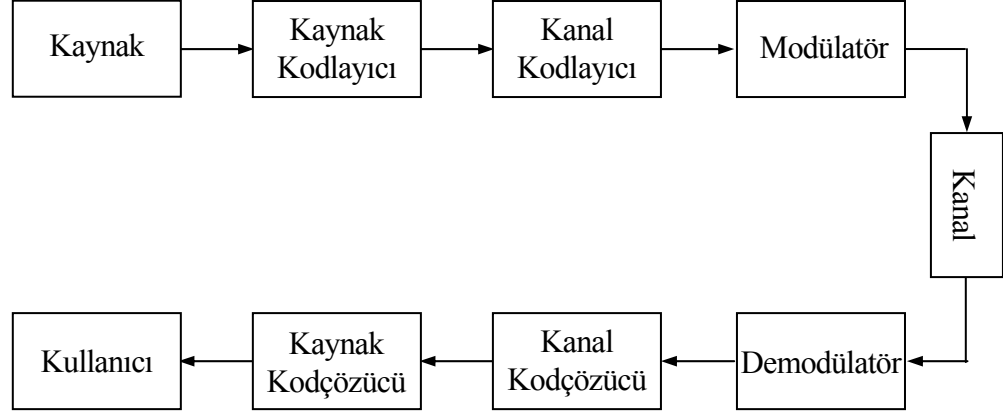
In order to study the performance of the proposed scheme, two different system are considered. These systems are two level turbo coded 4-ary CPFSK and three level turbo coded 8-ary CPFSK. Error performance curves of these systems in AWGN and fading channels are obtained and compared with other systems which are used in the literature and performace superiorities of the proposed system are presented.

I. GİRİŞ

Son yıllarda kablosuz haberleşme sistemlerini kullanan kullanıcı sayısı ve iletilen verilerin büyüklüğü hızlı bir şekilde artmaktadır. Bu artışı kısıtlayan en önemli neden radyo spektrumunun kısıtlı olmasıdır. Bundan dolayı, tüm ağ kapasitesini arttırmak için radyo spektrumunu etkin olarak kullanan kodlama, modülasyon ve sinyal işleme yöntemlerinin geliştirilmesi, günümüzde bir çok araştırmaya konu olmaktadır. Sayısal bir haberleşme sisteminde amaçlanan temel hedef en kısa sürede alıcı tarafa en doğru şekilde en çok verinin aktarılmasıdır. Bu amaç gerçekleştirilirken karşılaşılan en büyük problemler ise iletim ortamının sınırlı bir banda sahip olması ve işaretlerin düşük güç kullanılarak iletilmek istenmesidir. Yani sayısal haberleşme sistemlerinin verimi için iki temel ölçüt; işaretlerin ne kadar dar bantta iletilebildiği ve ne kadar düşük güçte iletilebildiğidir.

Sayısal bir haberleşme sistemi genel olarak Şekil 1.1'deki bloklardan oluşur. Bilgi kaynağında üretilen bilgiler öncelikle elektriksel işarete çevrilerek kaynak kodlayıcısı yardımıyla sayısal işaretlere dönüştürülür. Kaynak kodlayıcısında elde edilen sayısal bilgiler, kanaldan en az şekilde etkilenmesi için kanal kodlayıcıda kodlanır. Kanal kodlayıcıda veriye kontrol bitleri ekleneceğinden verinin büyüklüğü de artmış olur. Kanal kodlayıcıdan çıkan bitler, kullanılacak sayısal modülasyon işaretlerine eşlenerek kanala gönderilir. Kanalda, kanalın tipine göre modülasyonlu işaretin genlik, faz gibi özelliklerinde birtakım bozulmalar olur.

Alıcıya gelen bozulmuş işaretler öncelikle kullanılan modülasyon türüne ilişkin demodülatör kullanılarak demodüle edilirler. Demodüle edilen işaretler, kanal kodçözücü yardımıyla önce kodçözme işlemine tabi tutulur. Ardından kaynak kodlamanın tersi olan kaynak kodçözme işlemi yapılarak kullanıcının kullanımına sunulur.



Şekil 1.1 Sayısal bir iletişim sisteminin blok diyagramı

Shannon'un sayısal haberleşme ve bilgi kuramının temellerini atan 1940'ların sonlarında yaptığı çalışmalardan [1] sonra kanal kodlama ve modülasyon teknikleri üzerindeki çalışmalar önemli ölçüde ivme kazanmıştır. Shannon bu çalışmasında gürültülü bir kanalda birim zamanda iletilebilecek maksimum bilgi miktarını (Kanal kapasitesi) belirlemiş, iletim hızının kanal kapasitesinden düşük olduğu sürece güvenilir bir iletişimin yapılabileceğini göstermiştir. 1950-1980 yılları arasında blok kodlama, katlamalı kodlama vb. gibi çeşitli kanal kodlama teknikleri tanımlanmış ve üzerlerinde yoğun çalışmalar yapılmıştır. Yine bu yıllarda modülasyon yöntemleri üzerinde de çeşitli çalışmalar yapılmış fakat kodlama ve modülasyon hep ayrı ayrı düşünülmüştür. 1980'li yıllardaki çalışmalarıyla [2-4] Ungerboeck, bu düşüncenin yanlışlığını, kodlama ve modülasyonun beraber düşünülerek tasarım yapılması gerekliliğini ortaya koymuş ve kafes kodlamalı modülasyon (TCM) tanımını yapmıştır. Kafes kodlamalı modülasyona göre kanal kodlayıcı, kodlayıcı çıkışındaki bit dizileri arasındaki Hamming uzaklığını maksimum yapmak yerine kanala iletilecek modülasyonlu işaretler arasındaki Öklid uzaklığını maksimum yapmak için tasarlanmalıdır. Böylece kanalda iletilen işaretler birbirlerinden uzaklaştırılmış ve birbirine karışması engellenmiş olur. TCM tekniğinde Öklid uzaklığına göre karar veren yumuşak kararlı Viterbi kodçözücü [5] yapısı kullanılarak iletim hızı düşürülmeden önemli ölçüde güç kazancı elde edilebileceği [2-4]'te gösterilmiştir. Ungerboeck'in çalışmalarından sonra, TCM üzerine araştırmalar yoğunlaşmış, hem toplamsal beyaz Gauss gürültülü (AWGN) kanallar [6-10] hem de

sonümlmeli kanallar [11-15] için TCM tekniğine dayalı yeni sistemler tasarlanmıştır.

Önemli bir başka güç verimli kodlama tekniği de Imai-Hirakawa tarafından [16]'da tanımlanan çok düzeyli kodlama/çok aşamalı kodçözme yöntemidir. Çok düzeyli kodlama tekniğinde kodlanmış bit dizisi tek bir kanal kodlayıcısı yerine birbirine paralel birden çok kodlayıcı tarafından belirlenir. Bu yapıda her bir kodlayıcı bir düzey olarak isimlendirilir. Çok düzeyli kodlayıcıya paralel olarak çok aşamalı kodçözücüde de birbirine paralel olarak birden çok kodçözücü bulunur. Her bir kodçözücü ise bir aşama olarak tanımlanır. Yani sistemin her düzeyinde bulunan kodlayıcı için her aşamasında o kodu çözecek bir kodçözücü bulunur. Kodçözme işlemi birinci aşamadan başlar. Birinci aşama hariç her aşamadaki kodçözücü ilgili düzeye ilişkin kodu çözerken kendinden bir önceki kodçözücünün vermiş olduğu kararı bir yan bilgi olarak kullanır. Imai ve Hirakawa'nın yaptıkları çalışmada, düzeylerde blok kodlar kullanmasına karşın daha sonraları, düzeylerde katlamalı kodlar kullanan sistem tasarımları da yapılmış [17,18], ve bu sistemlerin önemli kodlama kazançları sağladığı da gösterilmiştir. Bu çalışmalardan sonra konu üzerindeki çalışmalar iyice artmış ve çok düzeyli kodlama temeline dayanan çok sayıda yeni sistem tasarımı gerçekleştirilmiştir [19-30]. Ayrıca Imai-Hirakawa tarafından önerilen çok düzeyli kodlama tekniği, Pottie ve Taylor [31] tarafından geliştirilmiş ve Imai-Hirakawa tip çok düzeyli kodlamanın aslında Pottie-Taylor tipi çok düzeyli kodlamanın özel bir biçimi olduğu gösterilmiştir.

1990'lı yıllara gelindiğinde ise, güç yönünden bilinen kodlama yöntemlerinden çok daha etkili bir kodlama yöntemi olan Turbo kodlar [32], Berrou, Glavieux ve Thitimajshima tarafından ortaya atılmıştır. Turbo kodlamada 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. Kodçözme işlemi *en büyük ileriye dönük* (Maksimum a posteriori, MAP) algoritma yöntemi ile iteratif olarak yapılmakta ve burada da genellikle Bahl Cocke Jelinek Raviv (BCJR) [33] algoritması kullanılmaktadır. BCJR algoritması çok fazla çarpma ve üssel işlemlerden oluşmaktadır. İlk defa Erfanian ve Pasupathy [34], Jacobian logaritması ile basitleştirilmiş MAP algoritmasını

geliştirmişlerdir. Bu algoritmada çarpma işlemleri toplama işlemine, toplama işlemleri de maksimumu seçme işlemine dönüşmüştür.

1993'te Turbo kodların ortaya atılmasından sonra konu üzerinde araştırmalar yoğunlaşmış ve 1996 yılına gelindiğinde, Blackert ve Wilson, Turbo Kafes Kodlamalı Modülasyon (TTCM) [35] yöntemini ortaya atmışlardır. Bu yöntemde ilk 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 Dik Genlik Modülasyonu (Quadrature Amplitude Modulation, QAM) gibi modülasyon yöntemleri ile modüle edilerek gönderilmektedir. Bu yöntemle, Turbo kodlama kafes kodlara uygulanmış olmaktadır. 1998 yılında Robertson ve Wörz [36] delikli kodlar için TTCM'i incelemiş ve ikili olmayan kodlar için MAP algoritmasını ayrıntılı olarak vermişlerdir.

Haberleşme sistemlerinde, sistemin iyi hata başarımının yanında band verimli olarak tasarlanması da oldukça önemlidir. Band verimliliği, band verimli modülasyon yöntemlerinin kullanılmasıyla sağlanabilir. Sürekli faz modülasyonu (CPM) [37,38], band verimliliği için oldukça uygun bir modülasyon yöntemidir. CPM bilinen faz modülasyonunun sürekli faza sahip şeklidir. Modüle edilen işaretler arasında faz atlamaları olmadığından daha dar band kullanılarak iletim yapılmasına olanak sağlar. Sürekli faz modülasyonunda faz sürekliliğinin sağlanması için bir önceki faz bilgisini tutan bellek kullanılmaktadır. Bu bellek sayesinde CPM, band veriminin yanında güç verimi de sağlar [39]. Sürekli fazlı frekans kaydırmalı anahtarlama (CPFSK), CPM modülasyonunun en çok bilinen özel bir biçimidir. CPFSK, sürekli faz modülasyonuna ilişkin frekans darbe yanıtının dikdörtgen olarak seçilmiş durumudur. Sürekli faz modülasyonunun tasarımı, band ve güç verimliliği üzerine 1980'li yıllarda çok sayıda araştırma yapılmıştır [40-43].

Sürekli faz modülasyonunun tasarımı iletilecek işaretler arasında faz atlamaması olmaması gerekliliğinden dolayı PSK, QAM gibi diğer modülasyonlara göre daha güç olmaktadır. Ayrıca faz bilgisini tutan belleklerden yani dahili kodlayıcıdan dolayı kodçözme karmaşıklığı da diğer modülasyonlu sistemlere göre daha yüksek olmaktadır. Sürekli faz modülasyonunun kodçözme işlemi dahili kodlayıcının kafesine göre viterbi kodçözücü ile yapılır. Hata başarımının daha da arttırılmak istenmesiyle dış bir kodlayıcı da kullanılırsa kodçözücünün kodçözme karmaşıklığı daha da artacaktır. Böyle bir sistemin kodçözme karmaşıklığı dış

kodlayıcının kodçözme karmaşıklığı ve sürekli faz kodlayıcısının kodçözme karmaşıklığının çarpımı kadar olabilmektedir. Bu nedenle, sürekli faz modülasyonlu sistemler tasarlanırken faz sürekliliğinin yanında kodçözme karmaşıklığı olabildiğince düşük sistem tasarımları yapılması oldukça önemlidir. Sürekli faz modülasyonunun kodlanmasına ilişkin şimdiye kadar üç ana yöntem önerilmiştir. Bu yöntemler klasik yöntem [38], [44,45], uyumlu kodlama yöntemi [46], ve ayrıştırma yöntemidir [47,48].

Tezde, sürekli faz modülasyonunun kodlanabilmesi için ayrıştırma yöntemi kullanılmıştır. Ayrıştırma yöntemine göre sürekli faz modülatörü, zamanla değişmeyen bir sürekli faz kodlayıcısı ve belleksiz bir modülatöre ayrıştırılır. Sürekli faz kodlayıcısı, modülasyonun bellek kısmını simgelemektedir ve dış kodlayıcı ile sürekli faz kodlayıcısının birleşiminden elde edilen ortak kafese ilişkin hata başarımları ölçütü maksimum yapılmaya çalışılır. Bu yöntemde, katlamalı kodlayıcının doğrusal sistematik yapıda seçilmesi durumunda, kodlayıcı ile sürekli faz kodlayıcısının bazı bellek elemanları ortak olabilmekte ve böylece birleşik yapının durum sayısı, kodlayıcının durum sayısı ile sürekli faz kodlayıcısının durum sayısının çarpımından küçük olmaktadır. Rimoldi [48], ayrıştırma yöntemi ile 4'lü CPFSK modülasyonunun indisinin 1/4 olduğu durum için 1/2 oranlı en iyi katlamalı kodlayıcıları elde etmiştir. Karam vd. [49] ise, 2/3 oranlı katlamalı kodlayıcılar ile 8'li CPFSK modülasyonu için en uygun çözümleri vermişlerdir. Huber ve Liu [50], Rimoldi'den bağımsız olarak CPM'in ayrıştırılarak kodlanmasını ele almış ve 4'lü CPFSK, 8'li CPFSK ve diğer modülasyon türleri için kodlar tasarlamışlardır.

Kodlanmış CPM'in alıcı kısmındaki kodçözme karmaşıklığını azaltmak amacıyla, kodlayıcıyı sürekli faz modülatörüne uyumlu tasarlayarak Viterbi kodçözücünün durum sayısını olabilecek en küçük değere getirmenin yanında, başka yöntemler de kullanılmaktadır. Bu amaçla Huber ve Liu [51], birleşik kafes üzerinde karmaşıklığı azaltan kimi algoritmaların kullanılmasını incelemiştir. Bu konudaki bir diğer yaklaşım ise kodlayıcı-sürekli faz modülatörü birleşik kafesini ayırarak, alınan işaretlerin önce CPM kafesi üzerinden demodüle edilmesi ve sonra CPM işaretlere ilişkin elde edilen faz durumu bilgisinin kodlayıcının kafesini çözen Viterbi kod çözücüyü aktarılmasıdır. Abrishamkar ve Biglieri [52] ve Sarı vd. [53] tarafından incelenen bu yöntemin sakıncası, alıcı kısımda birleşik kafes kullanılmadığı için hata başarımlarındaki düşüştür.

Sürekli faz modülasyonunun çok düzeyli kodlanması fikri ilk olarak Altunbaş ve Aygözü [54] tarafından ortaya atılmıştır. Daha sonra, Altunbaş [55]'teki çalışmasıyla konuyu çok daha detaylı olarak ele almıştır. Bu çalışmada CPFSK modülasyonunun çok düzeyli kodlama yöntemi kullanılarak kodlanmasına ilişkin yöntemler geliştirilmiş ve bu yöntemlerin AWGN ve sönümlemeli kanallardaki hata başarımları analizleri yapılmıştır. [55]'te verilen sonuçlara göre sürekli faz modülasyonunun çok düzeyli kodlanmasıyla önemli ölçüde güç ve band verimi elde edilmiştir. [55]'te kullanılan çok düzeyli kodlayıcı yapısında kodlama düzeylerinde kullanılan kodlayıcılar klasik katlamalı kodlardır. Alıcı tarafta ise viterbi kodçözücü kullanılmaktadır. Güç veriminin daha da artırılması amacıyla çok düzeyli kodlayıcının düzeylerinde katlamalı kodlar kullanmak yerine güç verimi daha yüksek olan turbo kodların kullanılması ve buna paralel olarak alıcı tarafta iteratif bir yapıya sahip olan turbo kodçözme algoritmalarının kullanılması ise bu tezde detaylı olarak verilmektedir.

Çok düzeyli turbo kodlanmış sistemlerin hata başarımının daha da iyileştirilmesi amacıyla kodlama düzeylerinde blok yada katlamalı kodlar yerine turbo kodların kullanılmasına ilişkin yöntem ise AWGN ve sönümlemeli kanallar için Osman, Uçan ve Odabaşoğlu tarafından [56,57]'de verilmiştir. Aynı sistemin çok girişli-çok çıkışlı (MIMO) kanallardaki hata başarımı da [58]'de detaylı olarak incelenmiştir. Çok düzeyli turbo kodlarda kanal denkleştirmeye ilişkin çalışma [59]'da, Sürekli faz modülasyonunun çok düzeyli turbo kodlanmasına ilişkin ilk çalışma ise [60]'ta yapılmıştır.

Sürekli faz modülasyonunun etkin bir başka kanal kodlama yöntemi olan turbo kafes kodlama ile kodlanmasına ilişkin çalışmalar ise [61,62]'de detaylı olarak yapılmıştır.

1.1 TEZDE İZLENEN YOL

Bu tezde, sürekli faz modülasyonunun çok düzeyli turbo kodlanması ve çok aşamalı turbo kodçözülmesine ilişkin bir yöntem geliştirilmiş ve bu yöntemin haberleşme sistemleri açısından verimi detaylarıyla incelenmiştir.

Tezin 2. bölümünde Genel kısımlar ana başlığı altında 2.1'de kanal kapasitesi ve kanal kodlama hakkında temel bilgiler verilmektedir. 2.2'de kodlamalı modülasyon teknikleri (Kafes kodlamalı modülasyon ve Çok düzeyli kodlama) detaylı olarak tanıtılmıştır. Bölüm

2.3'te ise turbo kodlama ve turbokodçözme teknikleri detaylarıyla anlatılmaktadır. Bölüm 2.4'te sürekli faz modülasyonu ve onun özel bir biçimi olan sürekli faz frekans kaydırmalı anahtarlama tanıtılmaktadır.

3. bölümde, malzeme ve yöntem olarak sürekli faz modülasyonunun çok düzeyli turbo kodlanmasına ilişkin önerilen yöntem tanıtılarak bu yöntem için örnek olarak tasarlanan iki düzeyli turbo kodlanmış 4'lü CPFSK sistemi ve üç düzeyli turbo kodlanmış 8'li CPFSK sistemi detaylarıyla açıklanmıştır.

Tezin 4. bölümünde, önerilen yöntemle ilişkin tasarlanan örnek sistemlerin hata başarımları analizleri verilmiş ve önerilen sistemlerin literatürdeki benzer sistemlere olan üstünlükleri gösterilmiştir.

Son bölüm olan 5. bölümde ise, tezde elde edilen sonuçlar tartışılarak konu hakkında ileride yapılabilecek çalışma önerileri verilmiştir.

2. GENEL KISIMLAR

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 [63,64] 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 KANAL KAPASİTESİ VE KANAL KODLAMA

Shannon Kanal Kapasitesi Teoremine göre; bir kanalın taşıyabileceği en büyük 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,

$$C = B \log_2(1 + S/N) \quad (2.1)$$

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 önlenir. 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 spektrumunun $N_0/2$ olduğu düşünülürse, kanal kapasitesi

$$C = B \log_2 \left(1 + \frac{S}{N_0 B} \right) \quad (2.2)$$

şeklinde olur. Maksimum kanal kapasitesini bulmak için B yi sonsuza götürürsek,

$$\begin{aligned} C &= \lim_{B \rightarrow \infty} B \log_2 \left(1 + \frac{S}{N_0 B} \right) \\ &= \lim_{B \rightarrow \infty} \frac{S}{N_0} \log_2 \left(1 + \frac{S}{N_0 B} \right)^{N_0 B / S} \\ &= \frac{S}{N_0} \log_2 e = 1.44 \frac{S}{N_0} \end{aligned} \quad (2.3)$$

elde edilir.

2.1.1 Blok Kodlar

Blok kod, belli sayıdaki mesaj parçalarının kabul edilebilir birleşimlerine kod kelimelerini atama kuralları kümesidir. Gönderilen bitlerin alıcıda tek çözülebilir (*unique 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 olsa da 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.

2.1.1.1 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 uzaklık $D_{\min}$ ile gösterilir. Alıcıda $D_{\min}-1$ bite 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.

2.1.1.2 Kod Uzunluğu

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

$$M \leq \frac{2^n}{\sum_{i=0}^E \binom{n}{i}} \quad (2.4)$$

Burada $\binom{n}{i}$ n 'in i 'li kombinasyonudur. Pratik açıdan bu eşitlikle 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 değerinin kaç olabileceği bulunabilir.

2.1.2 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 dizisi 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 m , k diziden oluşmaktadır. $m^{(p)} = (m_p, m_{p+k}, m_{p+2k}, \dots)$ $p \in (0, \dots, k-1)$. k adet giriş dizisi katlamalı kodlayıcıya girer ve n adet çıkış dizisi oluşur $x^{(q)}$, $q \in (0, \dots, n-1)$.

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

$$M_c = \sum_{p=0}^{k-1} M_p \quad (2.5)$$

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

$$K_c = 1 + \max_p M_p \quad (2.6)$$

olarak tanımlanır.

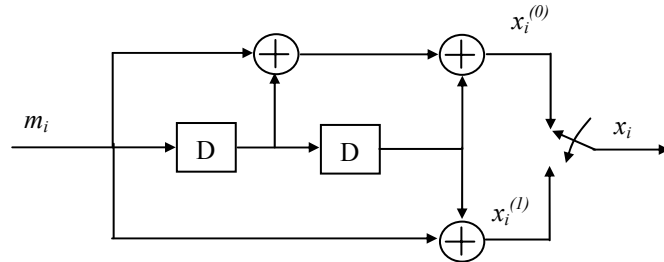
Kodlama işlemi, her oluşturma polinomunun katsayıları ile ilgili giriş arasında konvolüsyon yapılarak devam eder.

$$v_i^{(p,q)} = \sum_{j=0}^{M_p} m_{i-j}^{(p)} g_j^{(p,q)} \quad (2.7)$$

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

$$\mathbf{x}^{(q)} = \sum_{p=0}^{k-1} \mathbf{v}^{(p,q)} \quad (2.8)$$

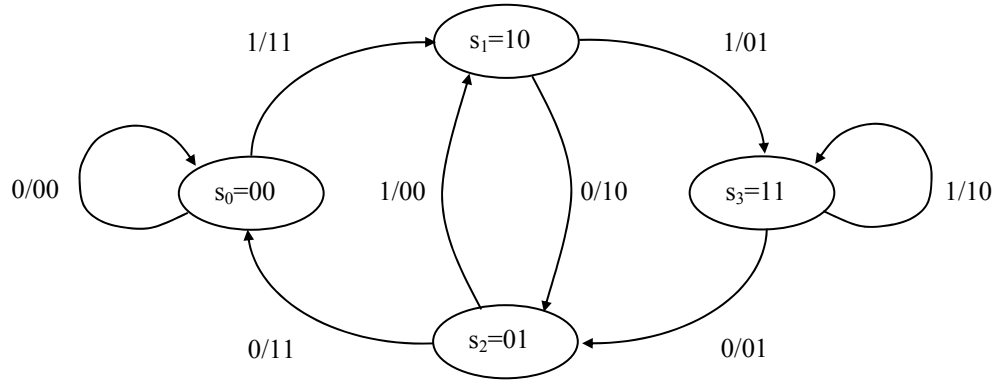
Girişte tek bir dizi olduğu düşünüldüğünde yani $k = 1$ durumu için kodlayıcı oranı $r = 1/n$ olur. Oran $1/n$ olunca $p = 0$ olur ve ihmal edilir. Bu durumda $1/2$ oranlı oluşturma polinomu $g^{(0)} = 1 + D + D^2$ ve $g^{(1)} = 1 + D^2$ olan bir katlamalı kodlayıcı Şekil 2.1'de verilmiştir. Her oluşturma polinomunun katsayıları genellikle sekiz tabanında (*octal*) verilir. Şekil 2.1'deki kodlayıcı için bu değer $(7,5)_{\text{octal}}$ dir.



Şekil 2.1 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcı

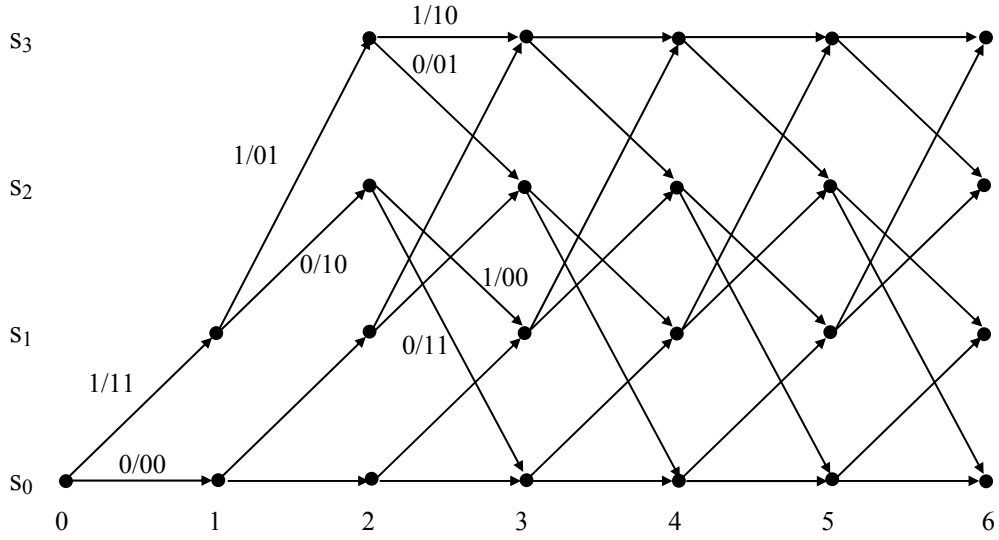
Katlamalı kodlayıcılar durum diyagramı ve kafes yapı ile gösterilebilen, sonlu durumlu bir Markov sürecidir. Katlamalı kodlayıcının durumu belleklerin içerikleri ile belirlenir. M_c adet bellek için toplam 2^{M_c} adet durum söz konusudur. Belleklerin durumları, giriş bitinin “1” veya “0” olmasına bağlı 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 söz konusudur ve bunlar s ile gösterilir. Tüm durumlar dallarla birbirlerine bağlıdır ve bu dalların üzerlerinde herhangi bir durumdayken m giriş yapıldığında x çıkışı olacağını gösteren m/x ifadeleri vardır. Şekil 2.1 için durum diyagramı Şekil 2.2’de verilmiştir.



Şekil 2.2 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 2.3’te 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 $s = (s_0, \dots, s_L)$ ile gösterilir. Kafes yapı belleklerin sıfır durumu ile başlar. Her yolun durum zinciri s , o yol ile ilgili kod kelimesi x ve o kod kelimesini oluşturan mesaj m , birbirleriyle birebir ilişkilidirler.



Şekil 2.3 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcının kafes diyagramı

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

$$d_{free} = \min\{w(\mathbf{x}) : x_0 = 1\} \quad (2.9)$$

Bu bilgi en önemli başarımlı ölçütüdür.

2.1.3 Geri Beslemeli Sistemik Katlamalı Kodlar

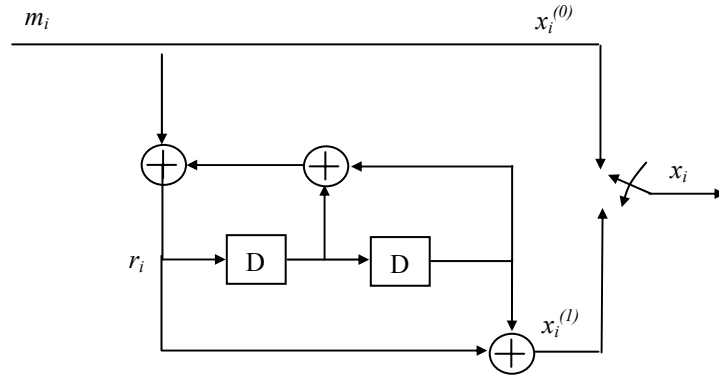
Döngüsel kodların Hamming uzaklığını azaltmadan sistemik kod yapılabildiği gibi katlamalı kodlar da Hamming uzaklığını azaltmadan sistemik yapılabilir. $1/2$ oranlı bir katlamalı kod $m(D)/g^{(0)}(D)$ polinom bölümünün kalanı $r(D)$ 'yi hesaplayarak yapılır. Parite çıkış polinomu $x^{(1)}(D)=r(D)g^{(1)}(D)$ polinom çarpımı ile bulunur. Sistemik çıktı ise $x^{(0)}=m(D)$ 'dir. $r(D)$ kalanı geri beslemeli bir kaydırmalı kaydedici 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 sistemik katlamalı (Recursive Systematic Convolutional, RSC) kodlar denir. Kodlama işlemi geri besleme değişkenini hesaplamakla başlar,

$$r_i = m_i + \sum_{j=1}^{M_c} r_{i-j} g_j^{(0)} \quad (2.10)$$

ve daha sonra parite çıktısı hesaplanır,

$$x_i^{(1)} = \sum_{j=1}^{M_c} r_{i-j} g_j^{(1)} \quad (2.11)$$

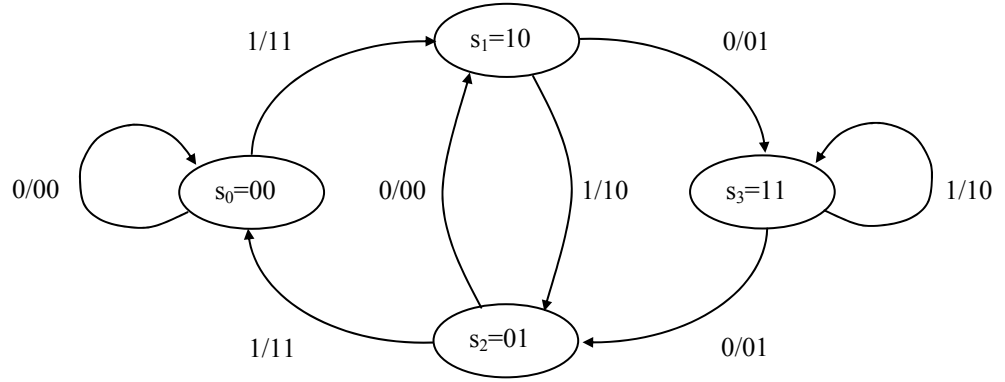
Şekil 2.4'te geri beslemeli sistematik katlamalı kodlayıcı verilmiştir. RSC kodlayıcıları sonlu durumludur. Durum ve kafes diyagramları ile gösterilirler. Şekil 2.5 ve Şekil 2.6'da durum ve kafes diyagramları verilmiştir. Dikkat edilirse geri beslemeli sistematik katlamalı kodlayıcıların durum ve kafes diyagramları katlamalı kodlayıcıların durum ve kafes diyagramlarıyla aynıdır. Tek fark Şekil 2.2'deki s_1 ve s_2 'den çıkan kollar için sadece giriş değerinin farklı olmasıdır. Bu durumda serbest uzaklık da değişmez.



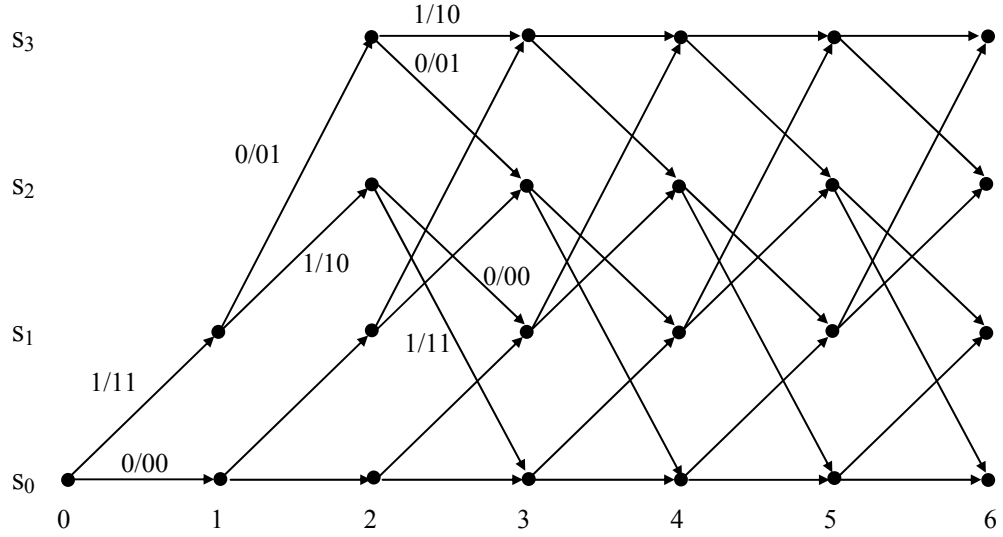
Şekil 2.4 Kodlama oranı $r = 1/2$ ve $K_c=3$ için katlamalı kodlayıcı

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.

$$m_i = \sum_{j=1}^{M_c} r_{i-j} g_j^{(0)} \quad (2.12)$$



Şekil 2.5 Kodlama oranı $r = 1/2$ ve $K_c=3$ için RSC kodlayıcı durum diyagramı



Şekil 2.6 Kodlama oranı $r = 1/2$ ve $K_c=3$ için RSC kodlayıcı kafes diyagramı

2.2 KODLAMALI MODÜLASYON TEKNİKLERİ

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. İzleyen alt bölümlerde kodlama ile modülasyon işlemlerini ayrı ayrı değil de tek bir süreç olarak ele alan yöntemler tanıtılmaktadır.

2.2.1 Kafes Kodlamalı Modülasyon

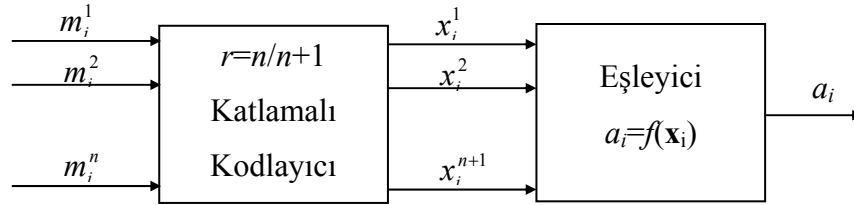
Kodlama ve modülasyonun birlikte düşünüldüğü kafes kodlamalı modülasyon (Trellis Coded Modulation, TCM) tekniği ilk defa Ungerboeck tarafından 1982 yılında önerilmiştir. TCM tekniği ile hata başarımını iyileştirmek için kanal işaretlerinin gereğinden fazla seçilmesi yeterli olmaktadır. Kafes kodlamalı 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

$$d_{serbest}^2 = \min \sum_{a_n \neq a'_n} d^2(a_n, a'_n) \quad (2.13)$$

şekil 2.7’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 bölmeleyerek eşleme” (*mapping by set 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 kümeler 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

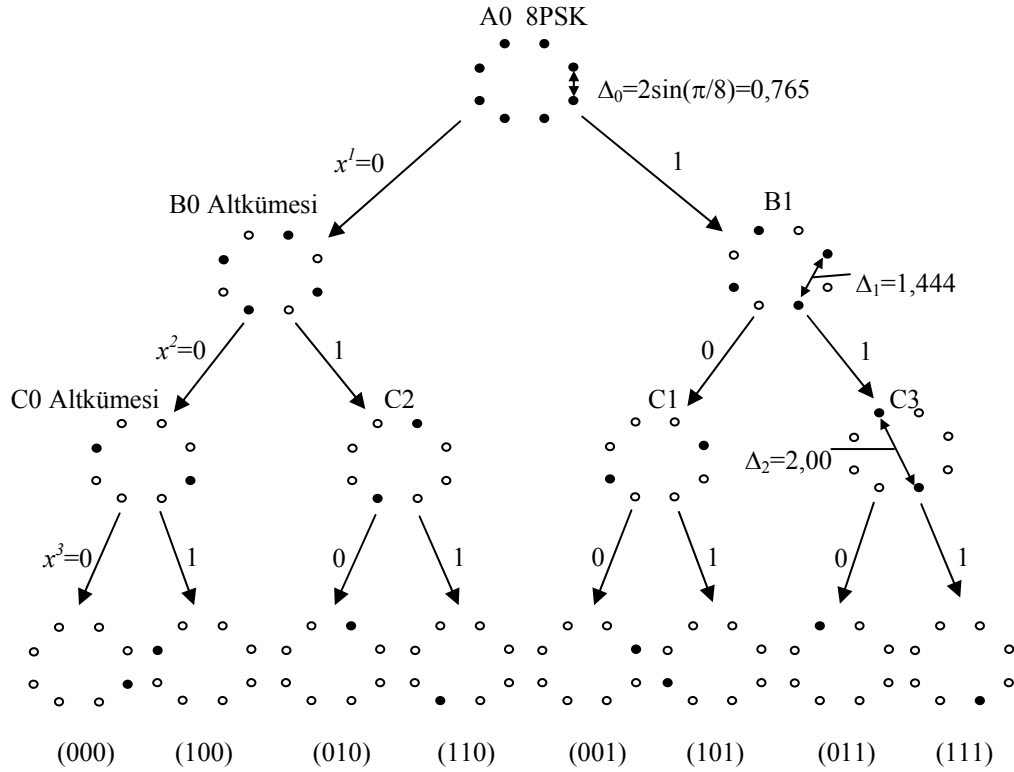
işaretlerinin eşit sayıda kullanılmasına, kalkış ve varış durumlarına atanan alt kümelerin maksimum olmasına dikkat edilmelidir.

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



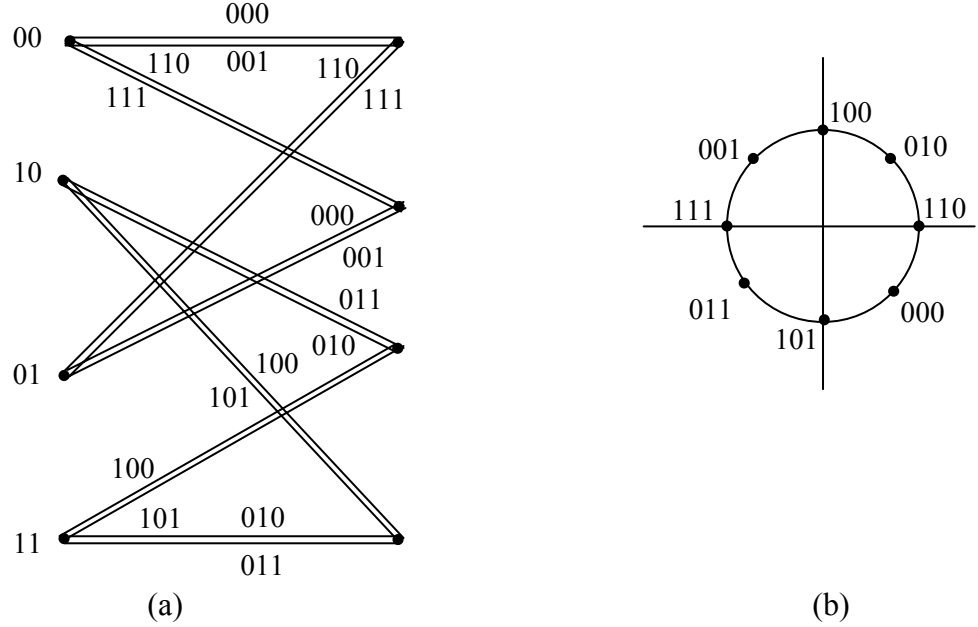
Şekil 2.7 Kafes kodlamalı modülasyon blok şeması

8PSK modülasyonu için kanal işaretlerinin ayrıştırılması Şekil 2.8’de verilmiştir. Burada Δ_i , i . Alt kümeye ilişkin en yakın iki işaret arasındaki karesel uzaklığı vermektedir. Küme bölmeleyerek eşleme mantığına uygun olarak $\Delta_0 < \Delta_1 < \Delta_2$ olmaktadır.



Şekil 2.8 8PSK işaret kümesinin ayrıştırılması

Şekil 2.9'daki kafes diyagramı, sıradan bir katlamalı kodun kafes diyagramıdır. Buradaki ana fark; iki durumu birbirine bağlayan yalnızca iki yolun olmasıdır. Her bir alt kümeden herhangi bir nokta seçen m_2 biti bu durum sonrasında oluşur. İki durumu birbirine bağlayan, paralel bir yol bir bölmeye; herhangi tek bir yol ise bir noktaya aittir.



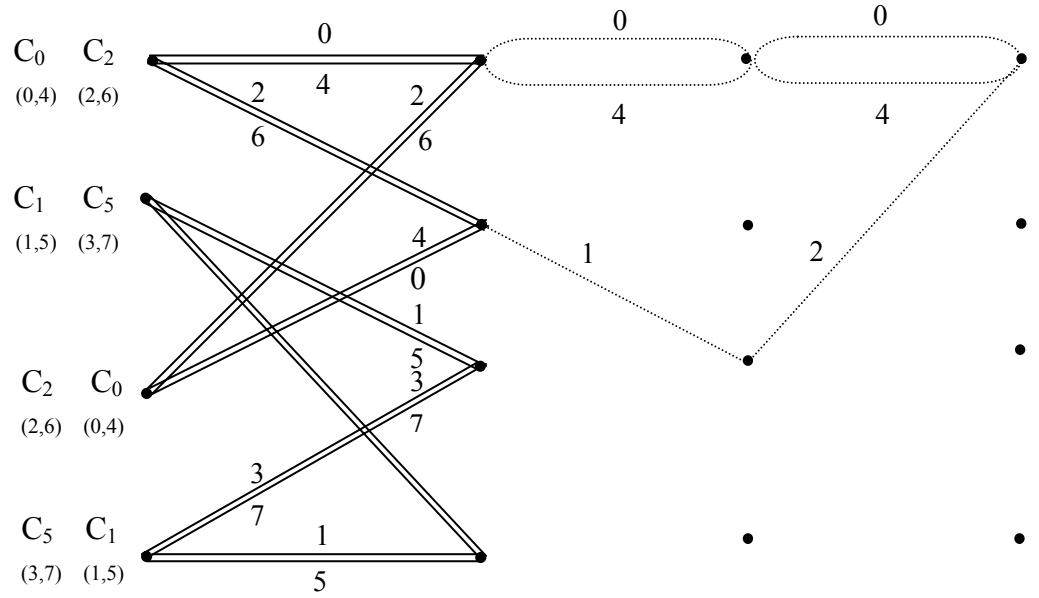
Şekil 2.9 (a) 4 Durumlu kafes diyagramı ve (b) Bitlerin eşlenmesi

Katlamalı kodlayıcının durum geçişleri ile işaretler arasındaki en iyi eşleme şu şekilde yapılmalıdır:

- a) Mevcut olan paralel geçişler aynı alt kümenin elemanlarıyla eşlenmelidir.
- b) Aynı durumdan başlayan ve aynı durumda sonlanan geçişler bir büyük alt kümenin elemanlarına eşlenmelidir.
- c) Tüm işaretler aynı sıklıkta (sayıda) kullanılmalıdır.

Kafes kodlamalı modülasyon yapısının nasıl işlediğini incelemek için öncelikle, bir düğümden başlayan ve bir başka düğümde sonlanan iki yol arasındaki minimum Öklid uzaklığı tanımlanmalıdır. TCM'de önemli bir karakteristik olan bu uzaklık, serbest Öklid uzaklığı olarak tanımlanır ve D_{fed} şeklinde gösterilir. D_{fed} bazen iki paralel geçiş arasındaki Öklid uzaklığı olarak da ifade edilebilir ve aşağıdaki gibi gösterilir.

$$d_2 = 2\sqrt{E_s} \quad (2.14)$$



Şekil 2.10 Olası minimum uzaklıklı iki yol

Şekil 2.10'da iki yol arasındaki Öklid uzaklığı $d^2 = d_0^2 + 2d_I^2 = 4.58E_s$ 'dir. Açıkça görülüyor ki bu uzaklık, iki paralel geçiş arasındaki uzaklıktan daha fazladır. Bu kod için serbest Öklid uzaklığı kolaylıkla $D_{fed} = d_2 = 2\sqrt{E_s}$ olarak tanımlanabilir. R birim zaman içinde iletilen ortalama bilgi biti sayısı (bilgi hızı, bit/sn) ve R kullanılabilir güç olmak üzere;

Kodlanmamış yapıda serbest Öklid uzaklığı

$$d^2_{kodlanmamış} = 4E_b = 4\frac{P}{R} \quad (2.15)$$

Kodlanmış yapıda serbest Öklid uzaklığı

$$d^2_{kodlanmış} = 4E_s = 8E_b \quad (2.16)$$

Bu durum sonucunda oluşan kodlama kazancı

$$G_{kodlama} = d^2_{kodlanmış} / d^2_{kodlanmamış} = 2 \sim 3 \text{ dB} \quad (2.17)$$

şeklinde olur. Kafes Kodlamalı Modülasyon yöntemiyle bant genişliğini arttırmadan 3dB kodlama kazancı elde edilebilir. Kafesteki durum sayısı arttıkça kodlama

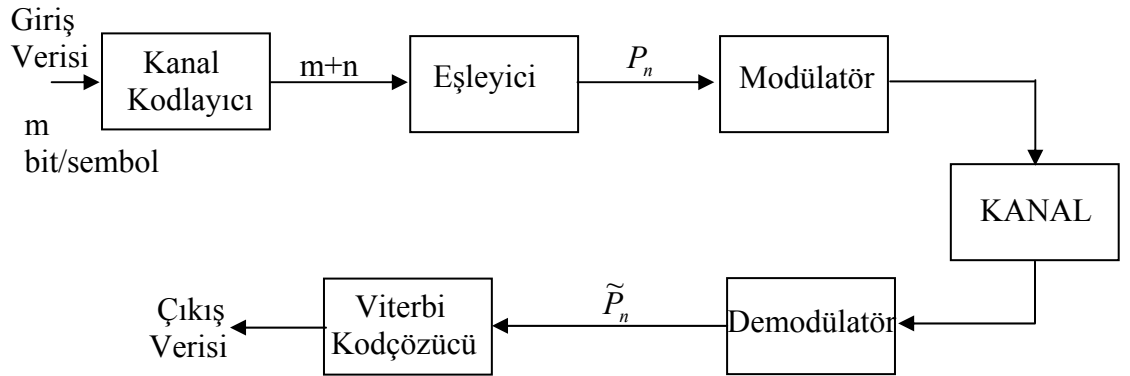
kazancının da artacağı, Ungerboeck'in yaptığı yoğun bilgisayar çalışmaları sonucunda kanıtlanmıştır. 8, 16, 32, 64, 128 ve 256 farklı giriş durumunda 3.6 dB ile 5.75 dB arasında kodlama kazancı sağlanabileceği ortaya çıkmıştır. Bu durum Tablo 2.1'de özetlenmiştir.

Tablo 2.1 Durum Sayısı-Kodlama Kazancı

Durum Sayısı	4	8	16	32	64	128	256
Kodlama kazancı (dB)	3	3.6	4.1	4.6	4.8	5.4	5.75

2.2.1.1 Kafes Kodlu Modülasyonda Kodçözme

Kafes kodlamalı modülasyonun kodunun çözülmesi iki farklı adımda gerçekleştirilir. İlk aşamada yapılacak olan, her bölmedeki birbirine en benzer noktaları bulmaktır. Bu işlem her bölmedeki alıcı noktasına en yakın Öklid uzaklığındaki noktaları bularak başarılabılır. Kodçözme işlemindeki bu adıma alt küme çözme denir. Bu aşamadan sonra sadece bir tane nokta seçilmiş olur, dolayısıyla sadece bir tane Öklid uzaklığı belirlenmiştir. İkinci aşamada ise, bir önceki aşamada bulunan Öklid uzaklığı yardımıyla kafes içindeki alınan işarete en yakın Öklid uzaklığına sahip yolu bulmaktır. Bu işlem Viterbi Algoritması yardımıyla yapılır.



Şekil 2.11 Kafes kodlamalı modülasyon şeması

Katlamalı kodların çözümünde bir çok yöntem vardır. (Yığın Algoritması, Geri beslemeli Kodçözme, Ardışıl Kodçözme). Bu yöntemler dışında en yaygın olarak kullanılan temeli 'maksimum benzerlik' teoremine dayanan Viterbi algoritmasıdır.

Verilen herhangi bir $d_1, \dots, d_{j_0}$, giriş bit dizisi için kodlanmış sembol dizisi $u_1, \dots, u_{2(j_0+2)}$, beyaz Gauss gürültüsü eklenmiş ve modüle edilmiş dizi, kodu çözüldükten sonra $v_1, \dots, v_{2(j_0+2)}$ olarak ifade edilir. Kodlama işlemi bu kodlar arasındaki maksimum benzerliğe göre yapılmaktadır. Kodçözücü, giriş bitlerinin kafeste izlediği yola, en benzer yolu bulmak kaydıyla giriş bilgi bitlerini geri elde etmeye çalışır. Bu işlemi yaparken öncelikle kafesteki muhtemel yolları araştırır ve bu yollar ile giriş dizinin metriklerini karşılaştırır.

Herhangi bir yol için j anındaki dal metriği; alınan n -bit ile alınması beklenen n -bit arasındaki birleşik olasılık olarak (2.18) eşitliği ile tanımlanır.

$$m_j^\alpha = \ln \prod_{i=1}^n P(r_{ji} | c_{ji}^\alpha) = \sum_{i=1}^n P(r_{ji} | c_{ji}^\alpha) \quad (2.18)$$

Herhangi bir yol için yol metriği ise, bu yol boyunca j anındaki dal metriklerinin toplamı olarak (2.19) eşitliği ile tanımlanır.

$$M^\alpha = \sum_{j=1}^J m_j^\alpha \quad (2.19)$$

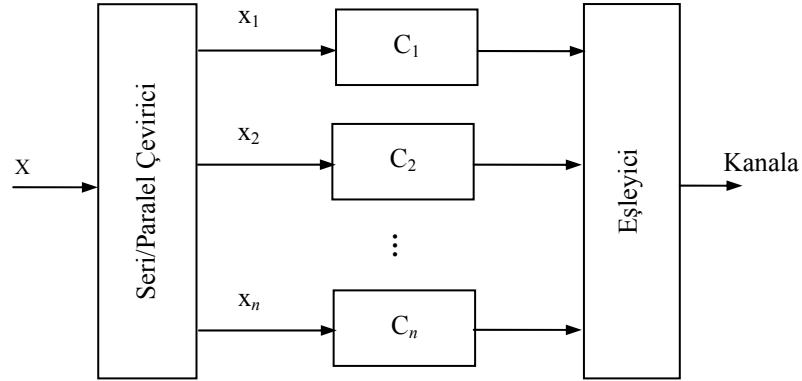
Kafesteki her bir düğüm için bu işlemler tekrar edildikten ve, kafesteki son düğümlere gelen alt ve üst dallardan en küçük metrikli dallar da seçildikten sonra, her düğümün metrikleri karşılaştırılır ve bunlardan en küçük olanı çıkış yolu olarak atanır.

$$\min(M^{\alpha_1}, M^{\alpha_2}, \dots, M^{\alpha_k}) \quad (2.20)$$

Genellikle (n, k, L) {çıkış, giriş, bellek sayısı} şeklinde tanımlanan bir katlamalı kodlayıcı $2^{(L-1)k}$ tane olası duruma sahiptir. Kafesteki her bir düğüm için 2^k tane birleşik yol vardır. Kodçözücü birleşik yollardan en kısa mesafeli olanı seçer ve kafesteki her adım sonunda $2^{(L-1)k}$ tane en kısa mesafeli yol bellekte tutulur. Tüm giriş bitleri işlendikten sonra kodçözücü minimum metrikli yolu ve bu yol üzerindeki çıkış bitlerini seçerek kodçözücü çıkışı olarak atar. Giriş dizilerinin çok uzun olduğu uygulamalarda kafes derinliğini (karar adım sayısı) $>(5 \times L)$ seçmek yeterlidir.

2.2.2 Çok Düzeyli Kodlama

Çok düzeyli kodlama, modülasyonlu işaretlerin sadece bir kodlayıcı yerine birbirine paralel birden çok sayıda kodlayıcı tarafından belirlenmesi düşüncesine dayanan bir kodlamalı modülasyon tekniğidir. Şekil 2.12’de genel bir çok düzeyli kodlayıcı yapısı görülmektedir. Şekilde 2.12’deki gibi n düzeyli bir yapıda giriş bit dizisi önce seri/paralel çevirici tarafından n adet paralel diziye çevrilir. Birinci düzeyde C_1 kodlayıcısı x_1 dizisini, ikinci düzeyde C_2 kodlayıcısı x_2 dizisini, n . Düzeyde ise C_n kodlayıcısı x_n dizisini kodlar. Düzeylerde kullanılan bileşen kodlar blok, katlamalı kodlayıcı gibi istenildiği gibi seçilebilir. Ayrıca hiç kodlama yapılmadan direk geçişler şeklinde de düzeyler tanımlanabilir. Kodlayıcı çıkışları kafes kodlamalı modülasyonda olduğu gibi küme bölmeleyerek eşleme tekniğiyle modülasyonlu işaretlere eşlenir. Herbir modülasyonlu işaret herbir düzeydeki kodlayıcı çıkışından birer bit alınarak belirlenebileceği gibi her düzeyden farklı sayıda bit de alınabilir. Herbir düzeydeki kodlayıcı çıkışından birer bit alınması durumu Imai-Hirakawa tip çok düzeyli kodlama olarak bilinir. Tezde kullanılan çok düzeyli kodlama tekniği de bu tip çok düzeyli kodlamadır.

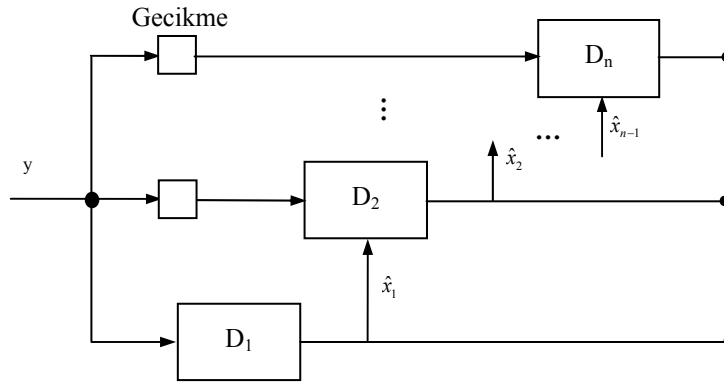


Şekil 2.12 Genel bir çok düzeyli kodlayıcı yapısı

Çok düzeyli kodlamada kodlayıcı tasarımı yapılırken daha üst düzeylerde daha güçlü kodlayıcıların kullanılmasına dikkat edilmesi gerekmektedir. Çünkü aşağıda da anlatıldığı gibi, çok aşamalı kodçözücünde sonraki kodçözme aşamaları hep önceki kodçözme aşamalarının bulduğu bilgileri bir ön bilgi kabul ederek kodçözme işlemi

yapar. Üst düzeylerde yapılan bir hatanın tüm kodçözme aşamalarını etkileyeceği ve işaretin tamamen yanlış çözüleceği açıktır.

Çok aşamalı kodçözme tekniği ise çok düzeyli kodlanmış bilgilerin kodçözümü için önerilmiş en iyiye yakın (suboptimum) kodçözme tekniğidir. Bu tekniğe göre çok düzeyli kodlayıcının her bir düzeyinde bulunan kodlayıcıların hepsi için ayrı ayrı olmak üzere o düzeydeki kodlayıcıya ilişkin bir kodçözücü kullanılır. Bu kodçözücülerin her biri ise bir kodçözme aşaması olarak isimlendirilir. Birçok aşamadan meydana gelen tüm kodçözücü yapısına ise çok aşamalı kodçözücü denilmektedir. Genel bir çok aşamalı kodçözücü yapısı Şekil 2.13'te görülmektedir.



Şekil 2.13 Genel bir çok aşamalı kodçözücü yapısı

Çok düzeyli kodlanmış bir işaret belirlenirken her düzeyden en az bir adet bit kullanıldığı için her modülasyon aralığında alınan işaret tüm kodçözme aşamalarından geçerek çözülmelidir. Alınan kodlanmış işaret öncelikle birinci düzey kodlayıcı olan C_1 kodlayıcısına ilişkin kodun çözülmesi için birinci aşama kodçözücü olan D_1 kodçözücüsüne girer. D_1 kodçözücüsü C_1 kodlayıcısına ilişkin kodu çözerek $\hat{x}_1$ 'yi bulur. $\hat{x}_1$ biti seçilen küme bölmeleme yöntemine göre işaret kümesini ikiye bölmüştür. D_1 kodçözücüsü tarafından bulunan $\hat{x}_1$ ile bu iki kümeden biri seçilerek olası işaret sayısı yarıya düşürülür. D_2 kodçözücüsü bu bilgiyi kullanarak C_2 kodlayıcısının kodladığı bilgiyi çözer. Böylece olası işaret sayısı bir kez daha yarıya düşürülmüş olur. D_2 kodçözücüsünün çıkışı olan $\hat{x}_2$ kullanılarak D_3 kodçözücüsü C_3 koduna ilişkin bilgiyi çözer. Bu işlem son kodçözme aşamasına kadar bu şekilde devam eder. Son

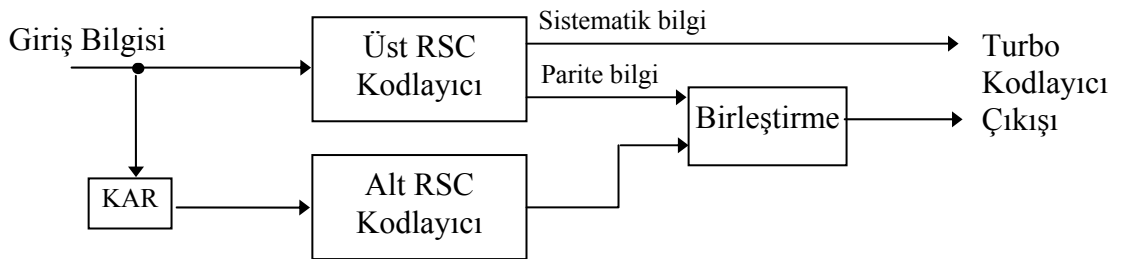
aşama kodçözücü kalan iki işaretten birini seçerek iletilen işareti kestirmiş olur. Çok düzeyli kodlanmış bilgilerin kodçözümü çok aşamalı kodçözücünden başka tüm kodlayıcıların kafesleri birleştirilip oluşan ortak kafesin kullanılarak çözülmesiyle de yapılabilir. Bu yöntem çok düzeyli kodlama için en iyi (optimum) kodçözme yöntemi olmasına karşın, ortaya çıkan ortak kafes çok fazla durum sayısı içereceğinden kodçözme karmaşıklığı çok yüksek olmaktadır. Burada kodçözme karmaşıklığı her bir kodçözücünün kodçözme karmaşıklıklarının çarpımı kadardır. Çok aşamalı kodçözmede ise toplam kodçözme karmaşıklığı her bir kodçözücünün karmaşıklıklarının toplamı kadardır. Bu açıdan bakıldığında çok aşamalı kodçözme tekniği çok düzeyli kodlama için kullanılabilecek en ideal kodçözme tekniğidir.

2.3 TURBO KODLAMA

Turbo kodlama [32], iki basit kodlayıcının paralel olarak birleştirilmesiyle oluşturulan güçlü bir kanal kodlama tekniğidir. Turbo kodlama 1993 yılında Beurro, Glavieux ve Thitimasjshima tarafından tanımlanmış ve güçlü hata düzeltme özelliğinden dolayı o tarihten günümüze literatürde üzerinde en çok araştırma yapılan kanal kodlama tekniklerinden biri olmuştur. Turbo kodlamanın hata başarımındaki üstünlüğü [32]'den kolayca görülebilir. Bu çalışmada 1/2 oranlı Turbo kodlar için 4 bellek elemanlı alt kodlayıcılar kullanılmış ve karıştırıcı uzunluğu 65536 olarak seçilmiştir. Bu durumda sistemin bit hata olasılığının kanal kapasitesine uzaklığı 0.7 dB olmaktadır.

2.3.1 Turbo Kodlayıcı

Turbo kodlayıcının genel yapısı Şekil 2.14'te verilmiştir. Turbo kodlayıcı, iki tane genellikle aynı yapıdaki Geribeslemeli Sistemik Katlamalı (Recursive Systematic Convolutional, RSC) kodlayıcıdan oluşur. Her iki kodlayıcı da aynı bilgiyi alır fakat ikinci kodlayıcı giriş bilgisi karıştırıcıdan geçtikten sonra oluşan yeni dizilişli bilgiyi alır. Turbo kodlanmış bilgilerin rastgele 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 2.14 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 kodlayıcının da düşük ağırlıklı bir çıkış yaratacak giriş almaları durumunda turbo kodların minimum uzaklıkları pek yüksek olmaz. İyi tasarlanmış bir turbo kodlayıcı ile bu gibi düşük ağırlıklı kodlara az rastlanır.

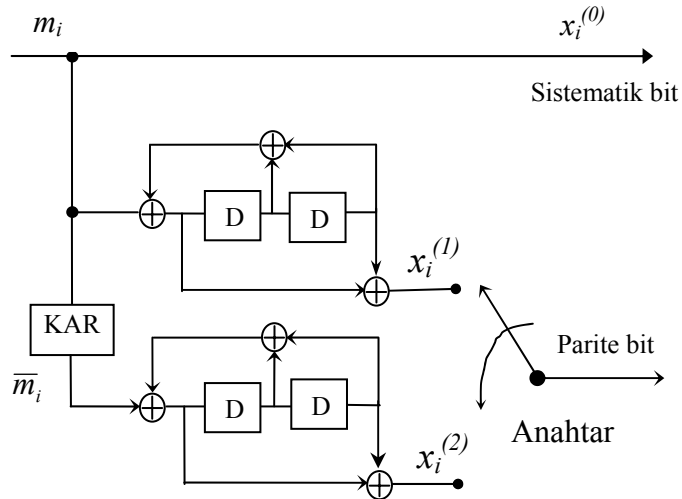
Turbo kodların performansı, düşük ağırlıklı kodların sayısının az olmasından dolayı, düşük sinyal gürültü oranlarında (Signal-to-Noise, SNR) iyidir. Buna rağmen, turbo kodların yüksek sinyal gürültü oranlarındaki başarımları minimum uzaklığın göreceli olarak daha az olmasından dolayı sınırlıdır. Geleneksel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

RSC kodlayıcılı bir turbo kodlayıcı Şekil 2.15'te gösterilmektedir. Karıştırıcının i . girişe uyguladığı fonksiyon $\alpha(i)$ 'dir, çıkışı $\bar{m}_i$ ile gösterilmektedir. Bu durumda giriş çıkış ilişkisi,

$$\bar{m}_{\alpha(i)} = m_i \quad (2.21)$$

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

$$m_{\alpha^{-1}(i)} = \bar{m}_i \quad (2.22)$$



Şekil 2.15 2 bellekli Turbo Kodlayıcı

Turbo kodlayıcının sistematik çıkışı $x^{(0)}$, giriş verisinin aynı sıralamasına sahiptir. $x^{(1)}$ ve $x^{(2)}$ parite bitleri üst (birinci) ve alt (ikinci) RSC kodlayıcılarının çıktılarıdır. 1/2

oranlı turbo kodlayıcının 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.

2.3.2 Turbo Kodçözme

Gürültülü elde edilmiş Markov sürecinin durum sırasını tahmin etmenin iki bilinen yöntemi vardır. Bunlardan birincisi *Viterbi algoritması* (VA) [5], diğeri de (sembol-sembol) *en büyük ileriye dönük* (MAP) [33] algoritmasıdır. Durum sırası tahmin edildikten sonra mesaj bitleri m veya ilgili mesaj kodları x 'in elde edilmesinde herhangi bir zorluk yoktur. VA ve MAP algoritmaları sağlamaya çalıştıkları kriterlere göre birbirlerinden ayrılırlar. Viterbi algoritması alınan y dizisine göre en olası durum dizisini bulur.

$$\hat{s} = \arg \left\{ \max_{\mathbf{s}} P[\mathbf{s} | \mathbf{y}] \right\} \quad (2.23)$$

MAP algoritması ise, y dizisine göre her bir kodlama adımı için en olası durumu bulur.

$$\hat{s}_i = \arg \left\{ \max_{s_i} P[s_i | \mathbf{y}] \right\} \quad (2.24)$$

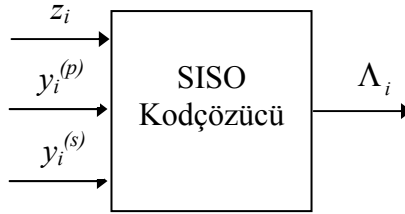
Bu iki algoritma arasındaki fark, VA ile kestirilen durumlar kafes yapı üzerinde birbirine bağlı yollar oluştururken MAP algoritması ile kestirilen durumlar için böyle bir durum söz konusu değildir. Bundan dolayı VA çerçeve hata oranını (Frame Error Ratio, FER) minimize ederken, MAP algoritması bit hata oranını (Bit Error Ratio, BER) minimize eder.

2.3.2.1 Yumuşak-Giriş Yumuşak-Çıkış (Soft-Input Soft-Output) Kodçözme

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 ,

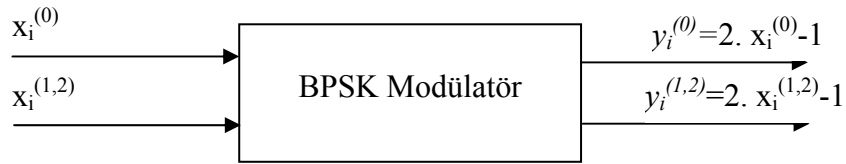
$$\Lambda_i = \ln \frac{P[m_i = 1|y]}{P[m_i = 0|y]} \quad (2.25)$$

ile gösterilir. Ön (*a priori*) bilgiyi giriş olarak alan ve çıkışında ileriye dönük (*a posteriori*) bilgiyi üreten kodçözme algoritmasına yumuşak-girişli yumuşak-çıkışlı (Soft Input Soft Output, SISO) kodçözme algoritması denir. Şekil 2.16’da gösterildiği gibi 1/2 oranlı bir RSC kodlayıcı için SISO kodçözücüsüne üç giriş gelir. Bunlar; sistematik bilgi $y_i^{(s)}$, parite bit $y_i^{(p)}$ ve diğer kodçözücünden gelen ön bilgi z_i ’dir. SISO kodçözücüsünün çıkışı Λ_i , (2.25) eşitliği ile verilmiştir.



Şekil 2.16 1/2 oranlı RSC kodlayıcı için yumuşak-giriş yumuşak-çıkış (SISO) kodçözücü

Modülasyon için Şekil 2.17’de gösterilen İkili Faz Kaydırmalı Anahtarlama (BPSK) uygulanır. BPSK ile değeri “1” olan bilgiler 1 olarak, “0” olan bilgiler ise -1 olarak gönderilir.



Şekil 2.17 BPSK Modulator

Buna bağlı olarak elde edilen gürültülü işaret,

$$y' = \rho \sqrt{E_s} (2x - 1) + n' \quad (2.26)$$

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 kanal AWGN olur. Aksi takdirde kanal sönümlemeli kanal olur ve kanalın tipi ρ 'nun dağılımına bağlıdır. Eşitlik (2.26)'nın genelde kullanılan hali

$$y = \rho(2x - 1) + n \quad (2.27)$$

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

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

$$\Lambda_i = \frac{4\rho_i^{(s)}E_s}{N_0} y_i^{(s)} + z_i + l_i \quad (2.28)$$

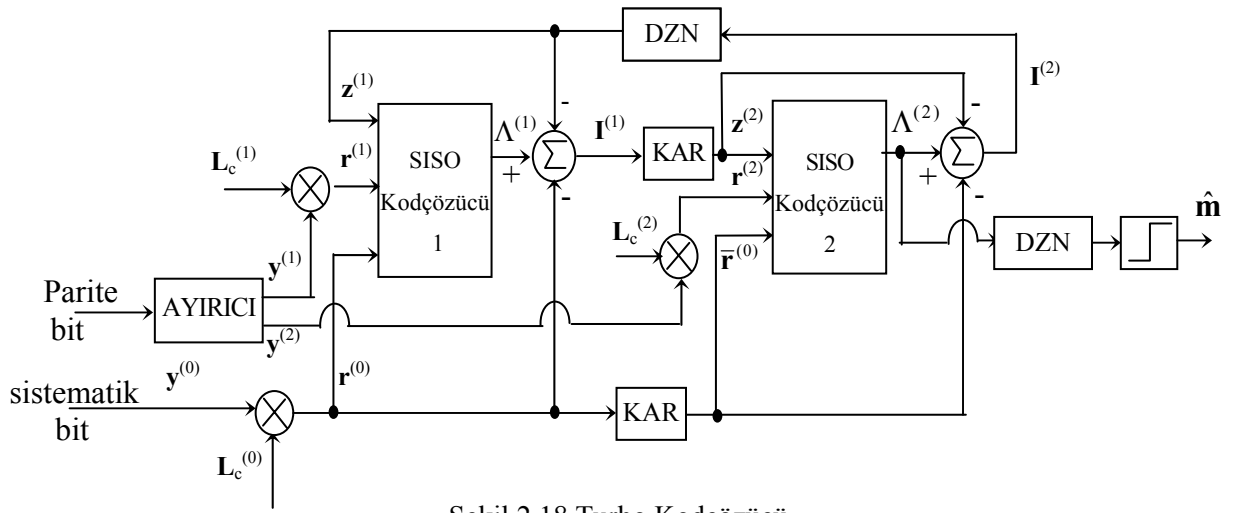
Burada l_i dışsal (*extrinsic*) bilgiyi göstermektedir. Eşitliğin sağ tarafındaki ilk iki terim alınan sistematik veri ve diğer kodlayıcıdan alınan ön bilgi z_i ile ilgilidir, *dışsal* bilgi kodçözme işleminin o anki safhasından elde edilen yeni bilgiyi gösterir. Pozitif geribesleme problemini önlemek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

$$l_i = \Lambda_i - \frac{4\rho_i^{(s)}E_s}{N_0} y_i^{(s)} - z_i \quad (2.29)$$

Turbo kodçözücünün genel yapısı Şekil 2.18'de verilmiştir. İlk kodçözücüye 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 kodçözücüye girer. Bir de ön bilgi $\mathbf{z}^{(1)}$ 'de ilk kodçözücüye girer. İlk kodçözücü, $\Lambda^{(1)}$ çıkışı verir. İlk kodçözücünün *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 kodçözücü için ön bilgi olarak kullanılır. Diğer kodçözücüye karıştırılmış ve $\mathbf{L}_c^{(0)}$ ile çarpılmış sistematik kanal gözlemi ile $\mathbf{L}_c^{(2)}$ ile çarpılmış ikinci kodlayıcının parite biti $\mathbf{r}^{(2)}$ girer. İkinci kodçözücü,

$\Lambda^{(2)}$ 'yi üretir. Ağırlıklı sistematik ve *ön* bilginin çıkarılması ile $\mathbf{I}^{(2)}$ elde edilir. İkinci kodçözücü tarafından elde edilen *dışsal* bilgi düzenlenerek bir sonraki iterasyonda birinci kodçözücüde *ön* bilgi olarak kullanılır. Gerekli iterasyon sayısına ulaşıldığında ikinci kodçözücünün çıkışı düzenlenip katı-karar'dan geçirilerek bilgi bitleri aşağıdaki gibi tahmin edilmiş olur.

$$\hat{m}_i = \begin{cases} 1 & \Lambda_{\alpha(i)}^{(2)} \geq 0 \\ 0 & \Lambda_{\alpha(i)}^{(2)} < 0 \end{cases} \quad (2.30)$$



Şekil 2.18 Turbo Kodçözücü

2.3.2.2 En Büyük İleriye Dönük Algoritma

Sembol-sembol MAP algoritması, 1974 yılında Bahl tarafından Viterbi algoritmasına alternatif olarak sunulmuştur [33]. MAP algoritmasının *ileri-geri* versiyonuna 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 durum geçişindeki mesaj biti için Markov süreci ile yaratılan semboller için gözlemlenen gürültülü 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 $P[m_i = 1 | y]$ ve $P[m_i = 0 | y]$. Turbo kodlamada kullanılan MAP algoritması son iterasyona kadar katı-karar uygulamaz.

Mesaj bitlerin APP'leri bulunmadan önce MAP algoritması gözlemlenen gürültülü dizi (y) ile olası her durum geçişi için $P[s_i \rightarrow s_{i+1} | \mathbf{y}]$ olasılığını hesaplar. Bu olasılık, koşullu olasılık tanımından aşağıdaki gibi yazılabilir.

$$P[s_i \rightarrow s_{i+1} | \mathbf{y}] = \frac{P[s_i \rightarrow s_{i+1}, \mathbf{y}]}{P[\mathbf{y}]} \quad (2.31)$$

Markov sürecinin özellikleri kullanılarak (2.31) eşitliğinin pay kısmı şu şekilde bölünebilir,

$$P[s_i \rightarrow s_{i+1}, \mathbf{y}] = \alpha(s_i) \gamma(s_i \rightarrow s_{i+1}) \beta(s_{i+1}) \quad (2.32)$$

Burada,

$$\alpha(s_i) = P[s_i, (y_0, \dots, y_{i-1})] \quad (2.33)$$

$$\gamma(s_i \rightarrow s_{i+1}) = P[s_{i+1}, y_i | s_i] \quad (2.34)$$

$$\beta(s_{i+1}) = P[(y_{i+1}, \dots, y_{L-1}) | s_{i+1}] \quad (2.35)$$

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

$$\begin{aligned} \gamma(s_i \rightarrow s_{i+1}) &= P[s_{i+1} | s_i] P[y_i | s_i \rightarrow s_{i+1}] \\ &= P[m_i] P[y_i | x_i] \end{aligned} \quad (2.36)$$

şeklinde yazılabilir. Burada m_i ve x_i , s_i durumundan s_{i+1} durumuna geçerken mesaj ve çıkış değerleridir. Eğer s_i den s_{i+1} 'e kafes yapıda bir geçiş yoksa (2.36) eşitliği sıfıra eşittir.

Eşitlik (2.36)'nın ilk terimi, $P[m_i]$, ön bilgidir ve aşağıdaki gibi tanımlanır.

$$P[m_i] = \begin{cases} \frac{e^{z_i}}{1 + e^{z_i}} & m_i = 1 \\ \frac{1}{1 + e^{z_i}} & m_i = 0 \end{cases} \quad (2.37)$$

ve logaritmik hesaplama için aşağıdaki eşitlik kullanılır.

$$\ln P[m_i] = z_i m_i - \ln(1 + e^{z_i}) \quad (2.38)$$

Eşitlik (2.36)'nın ikinci terimi kanal modeli ve modülasyonun bir fonksiyonudur, aşağıdaki gibi tanımlanır.

$$P[y_i | x_i] = \frac{1}{\sqrt{\pi N_0 / E_s}} \exp \left\{ \frac{-E_s}{N_0} \sum_{q=0}^{n-1} [y_i^q - \rho_i^{(q)} (2x_i^{(q)} - 1)]^2 \right\} \quad (2.39)$$

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

$$\alpha(s_i) = \sum_{s_{i-1} \in A} \alpha(s_{i-1}) \gamma(s_{i-1} \rightarrow s_i) \quad (2.40)$$

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

$$\beta(s_i) = \sum_{s_{i+1} \in B} \beta(s_{i+1}) \gamma(s_i \rightarrow s_{i+1}) \quad (2.41)$$

burada B , s_i ' lere bağlı 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

$$P[m_i = 1 | \mathbf{y}] = \sum_{S_1} P[s_i \rightarrow s_{i+1} | \mathbf{y}] \quad (2.42)$$

$$P[m_i = 0 | \mathbf{y}] = \sum_{S_0} P[s_i \rightarrow s_{i+1} | \mathbf{y}] \quad (2.43)$$

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

$$\Lambda_i = \ln \frac{\sum_{S_I} \alpha(s_i) \gamma(s_i \rightarrow s_{i+1}) \beta(s_{i+1})}{\sum_{S_0} \alpha(s_i) \gamma(s_i \rightarrow s_{i+1}) \beta(s_{i+1})} \quad (2.44)$$

Eşitlik (2.31)'deki $P[y]$ terimi, LLR ile oranlama yapıldığında birbirini götüreceğinden ayrıca hesaplanmaz.

2.3.2.3. 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 işlem ve hesaplamalara gereksinmesi, ikincisi ise hesaplama sonuçlarının sonlu hassasiyette gösterilmeye çalışıldığından taşma yapması veya işlem sonucunun sonsuza gitmesidir. Bu iki sebep işlemlerin logaritmik alanda yapılmasını gerektirmektedir. Bu sayede çarpma işlemleri toplamaya dönüşmektedir. Buna rağmen logaritmik alanda toplama işlemine pek başvurulmaz. Logaritmik alanda toplama işleminin nasıl yapıldığını göstermek için öncelikle *Jacobian logaritması* göz önüne alınır.

$$\begin{aligned} \ln(e^x + e^y) &= \max(x, y) + \ln(1 + \exp\{-|y - x|\}) \\ &= \max(x, y) + f_c(|y - x|) \end{aligned} \quad (2.45)$$

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

$$\ln(e^x + e^y) \approx \max(x, y) \quad (2.46)$$

olur.

MAP algoritmasının logaritmik alanda çalışan iki türü vardır. Bunlar *max-log-MAP* ve *log-MAP*'tir. Aralarındaki fark logaritmik alanda toplama işleminin nasıl yapıldığıdır.

Max-log-MAP algoritmasında, toplama işlemi (2.46)'daki maksimum seçme ile yapılır. Log-Map algoritmasında ise (2.45) kullanılarak maksimum seçmenin ardından düzeltme fonksiyonu hesaplanır. Bu düzeltme fonksiyonu iki sayının farkına bağlı olduğu için bir tablodan okunarak kullanılabilir.

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

$$\bar{\gamma}(s_i \rightarrow s_{i+1}) = \ln \gamma(s_i \rightarrow s_{i+1}) \quad (2.47)$$

$$= P[m_i] + P[y_i | x_i] \quad (2.48)$$

ve bu durumda durum geçiş ölçekleyicisinin logaritması

$$\bar{\gamma}(s_i \rightarrow s_{i+1}) = \ln P[m_i] - \frac{1}{2} \ln(\pi N_0 / E_s) - \frac{E_s}{N_0} \sum_{q=0}^{n-1} [y_i^{(q)} - (2x_i^{(q)} - 1)]^2 \quad (2.49)$$

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

$$\bar{\alpha}(s_i) = \ln \alpha(s_i) \quad (2.50)$$

$$\bar{\alpha}(s_i) = \ln \left\{ \sum_{s_{i-1} \in A} \exp[\bar{\alpha}(s_{i-1}) + \bar{\gamma}(s_{i-1} \rightarrow s_i)] \right\} \quad (2.51)$$

$$= \max_{s_{i-1} \in A}^* [\bar{\alpha}(s_{i-1}) + \bar{\gamma}(s_{i-1} \rightarrow s_i)] \quad (2.52)$$

burada A , s_i ' lere bağlı s_{i-1} ' lerin kümesini göstermektedir. 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,

$$\bar{\beta}(s_i) = \ln \beta(s_i) \quad (2.53)$$

$$\bar{\beta}(s_i) = \ln \left\{ \sum_{s_{i+1} \in B} \exp[\bar{\beta}(s_{i+1}) + \bar{\gamma}(s_i \rightarrow s_{i+1})] \right\} \quad (2.54)$$

$$= \max_{s_{i+1} \in B} [\bar{\beta}(s_{i+1}) + \bar{\gamma}(s_i \rightarrow s_{i+1})] \quad (2.55)$$

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.

$$\begin{aligned} \Lambda_i = & \ln \left\{ \sum_{s_i} \exp[\bar{\alpha}(s_i) + \bar{\gamma}(s_i \rightarrow s_{i+1}) + \bar{\beta}(s_{i+1})] \right\} \\ & - \ln \left\{ \sum_{s_0} \exp[\bar{\alpha}(s_i) + \bar{\gamma}(s_i \rightarrow s_{i+1}) + \bar{\beta}(s_{i+1})] \right\} \end{aligned} \quad (2.56)$$

Bu eşitlikte logaritma işlemlerini maksimum seçme işlemi olarak düşünebileceğimizden (2.56) eşitliği aşağıdaki şekile dönüşür.

$$\begin{aligned} \Lambda_i = & \max_{s_i} [\bar{\alpha}(s_i) + \bar{\gamma}(s_i \rightarrow s_{i+1}) + \bar{\beta}(s_{i+1})] \\ & - \max_{s_0} [\bar{\alpha}(s_i) + \bar{\gamma}(s_i \rightarrow s_{i+1}) + \bar{\beta}(s_{i+1})] \end{aligned} \quad (2.57)$$

2.3.3 Turbo Kafes Kodlamalı Modülasyon

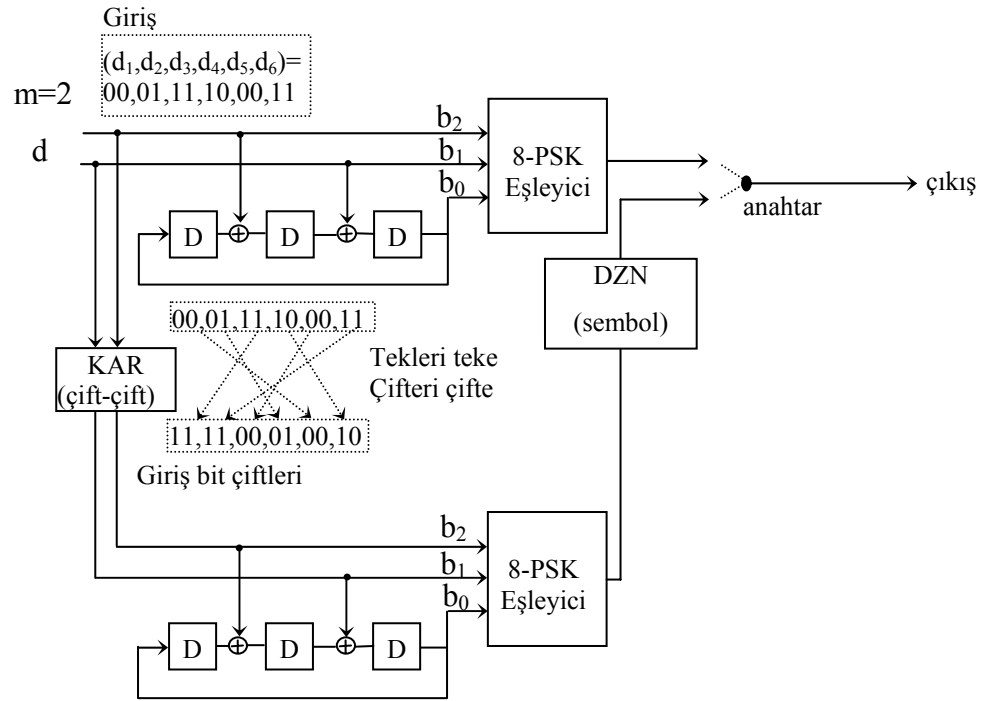
Turbo kafes kodlamalı modülasyon (Turbo Trellis Coded Modulation, TTCM) ilk olarak Blackert ve Wilson tarafından [35]'te sunulmuştur. Daha sonra 1996'da kafes kodlama için kullanılan MAP algoritması Robertson ve Wozz tarafından [36]'da ayrıntılı olarak verilmiştir.

2.3.3.1 Turbo Kafes Kodlayıcı

Turbo Kafes Kodlamalı Modülasyonda kullanılan kodlayıcının (Şekil 2.18) yapısı Turbo kodlayıcıdan biraz farklıdır. Bunlar; TTCM'de girişin en az 2 olması, girişlerin kodlanıp bir işarete eşlenmesi, kullanılan karıştırıcı ve düzenleyicilerin tekleri teke, çiftleri de çifte koyması gerekmesidir. Bu durumun kodçözme işleminde bir takım

farklılıklar oluşturmalarına rağmen yine de kodçözme işlemi Turbo Kodlama ile benzer sayılabilir.

Şekil 2.19’da verilen kodlayıcı yapısında iki giriş ($m=2$) biti ve üç çıkış biti olduğu görülmektedir. Kodlayıcının iki bitlik bir “ d ” girişine karşılık, çıkışındaki “ $b_0b_1b_2$ ” ile gösterilen üç biti Şekil 2.19’da gösterildiği gibi kompleks düzlemde eşlenir. Kodlayıcı çıkışının üç adet olmasından dolayı “ $b_0b_1b_2$ ” olası $8 (2^{m+1})$ sembole eşlenir. “ $b_0b_1b_2$ ”nin onluk düzendeki karşılığı b olarak gösterilirse, eşleyici çıkışı $x=\cos(b*2\pi/8)+j\sin(b*2\pi/8)$ şeklinde elde edilir. Buna göre giriş ve eşleyici çıktıları Şekil 2.20’de verilmiştir.



Şekil 2.19 Turbo Kafaç Kodlayıcısı

Bir diğer fark ise, bir kodçözücünden diğer kodçözücüye herbir işaret için giriş bitlerinin ihtimali kadar bilginin gitmesidir. Yani 2 girişli bir kodlayıcı kullanılırsa bir kodçözücünden diğerine 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 seçilebilir.

İteratif kodçözücü ikili kodçözücüye benzer, fakat bazı farklılıkları vardır. Bunlar bir kodçözücünden diğer kodçözücüye giden bilginin biçiminin farklı olması ve ilk kodçözme adımıdaki davranışın farklılığından kaynaklanır. Bilgi bitleri, diğer bir deyişle sistematik bitler, kodlanan parite biti ile birlikte eşleyici tarafından bir işarete karşı düşürülürler. Her kodçözücü kendi kodlayıcısından gelen bilgiyi çözer, yani tek indisli olanlar birinci kodçözücü, çift indisli olanlar ise ikinci kodçözücü tarafından çözülürler. Her kodçözücü için diğer kodlayıcıdan gelen bilgi “*” asterisk işareti ile gösterilir ve bu sembollere “delikli” (punctured) denir.

İkili Turbo kodlayıcılarda, her k bilgi biti için, kodçözücülerin çıkışları toplanmış üç parçadan oluşmuştur. Bunlar; sistematik bileşen, *ön* bileşen ve *dışsal* bileşendir. Sadece *dışsal* bileşen diğer kodçözücüye verilmelidir, aksi takdirde, bilgi diğer kodçözücüde birden fazla defa kullanılacak, 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.

Dışsal, Ön ve Sistematik Bileşenler

Bu kısımda iteratif kodçözücü anlatılacaktır. Burada olasılıkların logaritmik ifadeleri ($\log \Pr(\cdot)$) kullanılacak ve bu da $L(\cdot)$ ile gösterilecektir. Sistematik ve *dışsal* bileşenler bir kodçözücünden çıkar ve diğerine gönderilir ve o kodçözücüde *ön* bilgi olarak kullanılırlar. MAP algoritmasının çıkışı sistematik ve *dışsal* bileşen olarak *ön* bilgi ($\Pr\{d_k = i\}$)’den bağımsız tanımlanabilir. *Ön* bilgi,

$$L_p(d_k = i) = \log \Pr\{d_k = i\} \quad (2.58)$$

olarak tanımlanır ve bu MAP algoritmasının sonucundan çıkartılarak,

$$L_{e\&s}(d_k = i) = \log \Pr\{d_k = i | \underline{\mathbf{y}}\} - \log \Pr\{d_k = i\} \quad \forall i \in \{0 \dots 2^m - 1\} \quad (2.59)$$

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

Ön Ölçekleyici Hesaplaması

Yöntemi kullanabilmek için *ön* bilginin çıkartılması gerekmektedir. Fakat ilk iterasyonun birinci adımında *ön* bilgiden bahsetmek mümkün değildir. Bu yüzden birinci kodçözücü işleme başlamadan önce *ön* bilginin elde edilmesi gerekmektedir. Bu şekilde *dışsal* bilginin birinci kodçözücünden ikinci kodçözücüye gönderilirken * geçişlerinde sistematik bilgi içerebilmesi sağlanır. Burada gönderilen sembol bilgi grubu d_k ile kısmen, bunun yanısıra diğer kodlayıcı tarafından oluşturulan bilinmeyen parite bit $b_k^{0,*} \in \{0,1\}$ tarafından saptanır. Bu yüzden *ön* bilgi birleşik Bayes kuralı ile saptanır,

$$\begin{aligned} \Pr\{d_k = i | y_k\} &= \text{const} \cdot p(\mathbf{y}_k | d_k = i) \\ &= \text{const} \cdot \sum_{j \in \{0,1\}} p(\mathbf{y}_k, b_k^{0,*} = j | d_k = i) \\ &= \frac{\text{const}}{2} \cdot \sum_{j \in \{0,1\}} p(\mathbf{y}_k | d_k = i, b_k^{0,*} = j) \end{aligned} \quad (2.60)$$

Burada $\Pr\{b_k^{0,*} = j | d_k\} = \Pr\{b_k^{0,*} = j\} = 1/2$ olduğu varsayılır. Başka bir deyişle, x_k içindeki parite biti bilgi bit grubundan istatistiksel olarak bağımsızdır ve bu durumda parite bitinin “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 kodçözücü * geçişinde değilse $\Pr\{d_k = i\} = 1/2^m$ değerine eşitlenir.

TTCM için MAP Algoritması

Bu kısımda, ikili olmayan kafes yapılı kodlayıcılar için sembol-sembol MAP algoritması açıklanacaktır. Burada klasik Kafes Kodlamalı Modülasyon yapıları

düşünülmüştür. Kodlayıcıda v tane bellek olduğu düşünülürse, 2^v tane durum sözkonusudur. Kodlayıcının k anındaki durumu, $S_k \in \{0,1,\dots,2^v-1\}$ şeklinde gösterilir. Kodlayıcıya giren bilgi bitlerinin oluşturduğu m tane bilgi biti, $(0\dots 2^m-1)$ arasındaki tam sayılarla gösterilir ve giriş bitlerine göre $k-1$ 'inci adımdan k 'ıncı adıma geçişi belirler. Karıştırıcı boyutunun N olduğu düşünülürse, blok başına modüle edilen sembol sayısı $N \times n$ dir. Burada n , gönderilen işaretin boyutunun yarısıdır. Bu durumda blok başına gönderilen bilgi biti sayısı $N \times m$ dir. Alıcı N set n gürültülü 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 gürültüsü eklenmiş halidir. Her $\mathbf{x}_k = (x_k^0, \dots, x_k^{(n-1)})$ k anında eşleyiciden çıkan n sembollü bir gruptur.

Kodçözücünün 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.

$$\alpha_{k-1}(M') = \frac{p(S_{k-1} = M', \underline{\mathbf{y}}_1^{k-1})}{p(\underline{\mathbf{y}}_1^{k-1})} \quad (2.61)$$

$$\beta_k(M) = \frac{p(\underline{\mathbf{y}}_{k+1}^N | S_k = M)}{p(\underline{\mathbf{y}}_{k+1}^N | \underline{\mathbf{y}}_1^k)} \quad (2.62)$$

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

$$\begin{aligned} \gamma_i(y_k, M', M) &= p(y_k | d_k = i, S_k = M, S_{k-1} = M') \\ &\cdot q(d_k = i | S_k = M, S_{k-1} = M') \\ &\cdot \Pr\{S_k = M | S_{k-1} = M'\} \end{aligned} \quad (2.63)$$

$q(d_k = i | S_k = M, S_{k-1} = M')$ “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. (2.63) eşitliğinin üçüncü çarpanını hesaplamak için ön bilgi kullanılır.

$$\begin{aligned}
& \Pr\{S_k = M \mid S_{k-1} = M'\} \\
&= \begin{cases} \Pr\{d_k = 0\}, & q(d_k = 0 \mid S_k = M, S_{k-1} = M') = 1 \\ \Pr\{d_k = 1\}, & q(d_k = 1 \mid S_k = M, S_{k-1} = M') = 1 \\ \Pr\{d_k = 2^m - 1\}, & q(d_k = 2^m - 1 \mid S_k = M, S_{k-1} = M') = 1 \end{cases} \\
&= \Pr\{d_k = j\}
\end{aligned} \tag{2.64}$$

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

$(d_k=i, y_k, S_{k-1}=M')$ olayı için, S_k 'nin bilinmesi durumunda, $\underline{y}_{k+1}^N$ 'in hiç bir etkisi yoktur. Bundan dolayı k anı ve önceki durumların bilindiği düşünülerek bir sonraki girişe ilişkin geçiş olasılığı,

$$p(\underline{y}_{k+1}^N \mid S_k = M) = p(\underline{y}_{k+1}^N \mid d_k = i, \mathbf{y}_k, S_k = M, S_{k-1} = M') \tag{2.65}$$

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

$$p(\underline{y}_1^{k-1}) = \frac{p(\underline{y}_1^k)}{p(\mathbf{y}_k \mid \underline{y}_1^{k-1})} \tag{2.66}$$

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

$$\begin{aligned}
\alpha_{k-1}(M') \cdot \beta_k(M) \cdot \gamma(\mathbf{y}_k, M', M) &= p(S_{k-1} = M', \underline{y}_1^{k-1}) \\
&\cdot p(\underline{y}_{k+1}^N, d_k = i, \mathbf{y}_k, S_k = M \mid S_{k-1} = M') \tag{2.67} \\
&\cdot \frac{p(\mathbf{y}_k \mid \underline{y}_1^{k-1})}{p(\underline{y}_1^N)}
\end{aligned}$$

Açık olarak yazılırsa, (2.67)'nin ikinci çarpanı $p(\underline{y}_{k+1}^N, d_k, \mathbf{y}_k, S_k = M \mid S_{k-1} = M') = p(\underline{y}_{k+1}^N, d_k, y_k, S_k = M \mid S_{k-1} = M', \underline{y}_1^{k-1})$ 'dir ve bu durumdan dolayı ileri, geri ve dal geçiş olasılıklarının çarpımı aşağıdaki şekilde yazılabilir.

$$\begin{aligned}\alpha_{k-1}(M') \cdot \beta_k(M) \cdot \gamma(\mathbf{y}_k, M', M) \cdot \frac{1}{p(\mathbf{y}_k | \underline{\mathbf{y}}_1^{k-1})} &= p(S_{k-1} = M', S_k = M, \underline{\mathbf{y}}_1^N) \cdot \frac{1}{p(\underline{\mathbf{y}}_1^N)} \\ &= p(S_{k-1} = M', S_k = M | \underline{\mathbf{y}}_1^N)\end{aligned}\quad (2.68)$$

Buna göre MAP algoritmasının çıkışı aşağıdaki gibi yazılabilir.

$$P_r\{d_k = i | \underline{\mathbf{y}}\} = \text{sabit} \cdot \sum_M \sum_{M'} [\gamma_i(y_k, M', M) \cdot \alpha_{k-1}(M') \cdot \beta_k(M)] \quad \forall i \in \{0 \dots 2^m - 1\} \quad (2.69)$$

Buradaki sabit, eşitliği tüm i üzerinden toplamını alıp normalize ederek ortadan kaldırılabilir. $\Pr\{d_k | \underline{\mathbf{y}}\}$ 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 kalmıştır. İlk olarak,

$$\Pr\{S_k = M | \underline{\mathbf{y}}_1^{k-1}, \mathbf{y}_k\} \cdot p(\mathbf{y}_k | \underline{\mathbf{y}}_1^{k-1}) = p(\mathbf{y}_k, S_k = M | \underline{\mathbf{y}}_1^{k-1}) \quad (2.70)$$

ele alınsın, bu eşitliğin her iki tarafı $p(\mathbf{y}_k | \underline{\mathbf{y}}_1^{k-1})$ 'e bölünürse aşağıdaki şekilde açılabilir.

$$\Pr\{S_k = M | \underline{\mathbf{y}}_1^k\} = \alpha_k(M) = \frac{\sum_{M'} p(y_k, S_k = M, S_{k-1} = M' | \underline{\mathbf{y}}_1^{k-1})}{\sum_M \sum_{M'} p(y_k, S_k = M, S_{k-1} = M' | \underline{\mathbf{y}}_1^{k-1})} \quad (2.71)$$

ve eşitlik (2.69)'dan dolayı

$$\alpha_k(M) = \frac{\sum_{M'} \Pr\{\mathbf{y}_k, S_k = M | S_{k-1} = M'\} \cdot p(S_{k-1} = M' | \underline{\mathbf{y}}_1^{k-1})}{\sum_M \sum_{M'} p(\mathbf{y}_k, S_k = M | S_{k-1} = M') \cdot \Pr\{S_{k-1} = M' | \underline{\mathbf{y}}_1^{k-1}\}} \quad (2.72)$$

şeklinde yazılabilir. Tüm olası giriş bitleri için dal geçiş olasılığı

$$\gamma_T(\mathbf{y}_k, M', M) = \sum_{i=0}^{2^m-1} \gamma_i(\mathbf{y}_k, M', M) \quad (2.73)$$

şeklinde tanımlanır. Bu durumda ileri ve geri olasılıklar,

$$\alpha_k(M) = \frac{\sum_{M'} \gamma_T(\mathbf{y}_k, M', M) \cdot \alpha_{k-1}(M')}{\sum_M \sum_{M'} \gamma_T(\mathbf{y}_k, M', M) \cdot \alpha_{k-1}(M')} \quad (2.74)$$

$$\beta_k(M) = \frac{\sum_{M''} p(S_{k+1} = M'', \underline{\mathbf{y}}_{k+1}^N | S_k = M)}{p(\underline{\mathbf{y}}_{k+1}^N | \underline{\mathbf{y}}_1^k)} \quad (2.75)$$

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

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 eşitlik aşağıdaki gibi gösterilir.

$$\beta_k(M) = \frac{\sum_{M''} p(S_{k+1} = M'', \mathbf{y}_{k+1} | S_k = M) \cdot \frac{p(\underline{\mathbf{y}}_{k+2}^N | S_{k+1} = M'')}{p(\underline{\mathbf{y}}_{k+2}^N | \underline{\mathbf{y}}_1^{k+1})}}{\sum_{M''} \sum_M p(S_{k+1} = M'', S_k = M, \mathbf{y}_{k+1} | \underline{\mathbf{y}}_1^k)} \quad (2.76)$$

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

İleri olasılıklar ($\alpha_k(M)$), kodlayıcının başlangıç durumunun “0” olduğu bilindiğinden dolayı $\alpha_0(0)=1$ alınarak ve dal geçiş olasılıkları γ ’da önceden hesaplanmış olduğundan dolayı (2.74) eşitliğine 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 değerleri kullanılarak (2.76) eşitliğine göre hesaplanabilir. Bu işlemlerden sonra MAP algoritmasının çıkışı (2.69) eşitliğine göre elde edilir.

2.4 SÜREKLİ FAZ MODÜLASYONU

Sayısal veri iletimi, analog veri iletimine göre hata koruma sıkıştırma vb. gibi konularda bir çok üstünlük göstermektedir. Bir çok fiziksel ortam sayısal verilerin doğrudan iletilmesine uygun değildir. Kablosuz kanallar veya bakır kablo gibi bu tür ortamlarda sayısal verinin iletilmesi için sayısal modülatörlere ihtiyaç duyulmaktadır. Sayısal modülatör ayrık zamanlı sayısal veriyi analog bir dalga şekline dönüştürür. Sayısal bir veri dizisinin analog bir dalgaya dönüştürülmesi işlemi modülasyon, tersi ise demodülasyon olarak bilinir. Verinin iletileceği analog dalgaya taşıyıcı denir. Taşıyıcı dalganın çeşitli karakteristikleri değiştirilerek bilgi işareti bu taşıyıcı dalganın üzerine bindirilir.

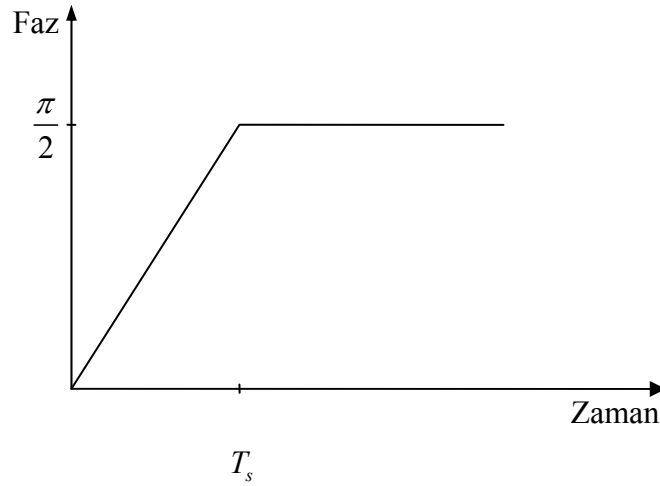
Sayısal faz modülasyonlu yapıda bilgiyi taşımak için taşıyıcı dalganın fazı kullanır. Faz Kaydırmalı Anahtarlama (Phase Shift Keying, PSK) ve Frekans Kaydırmalı Anahtarlama (Frequency Shift Keying, FSK) sayısal faz modülasyonun en bilinen iki tipidir. PSK’da giriş dizisi işaretin fazını her bir işaretleşme aralığında sonlu uzunluktaki bir değer kümesinden seçer. FSK’da ise giriş dizisi işaretin frekansını taşıyıcı frekans çevresinde değiştirir.

Son yıllarda kodlama kuramındaki en büyük gelişme, band genişliği verimliliğini arttıran kodlama yöntemlerinin bulunmasıdır. Bu işaretleşme yöntemleri aynı anda hem enerjiyi hem de band genişliğini koruyarak kodlama kazancı sağlar. Kodlama sırasında enerjinin ve band genişliğinin korunabileceği fikri, 1960’tan önce Shannon [1] tarafından açıklanmış ancak bu alandaki çalışmalar 1975’ten sonra hız kazanabilmiştir. Daha önceki yıllarda enerji verimliliği için band genişliğinin artırılması gerektiği düşünülmektedir.

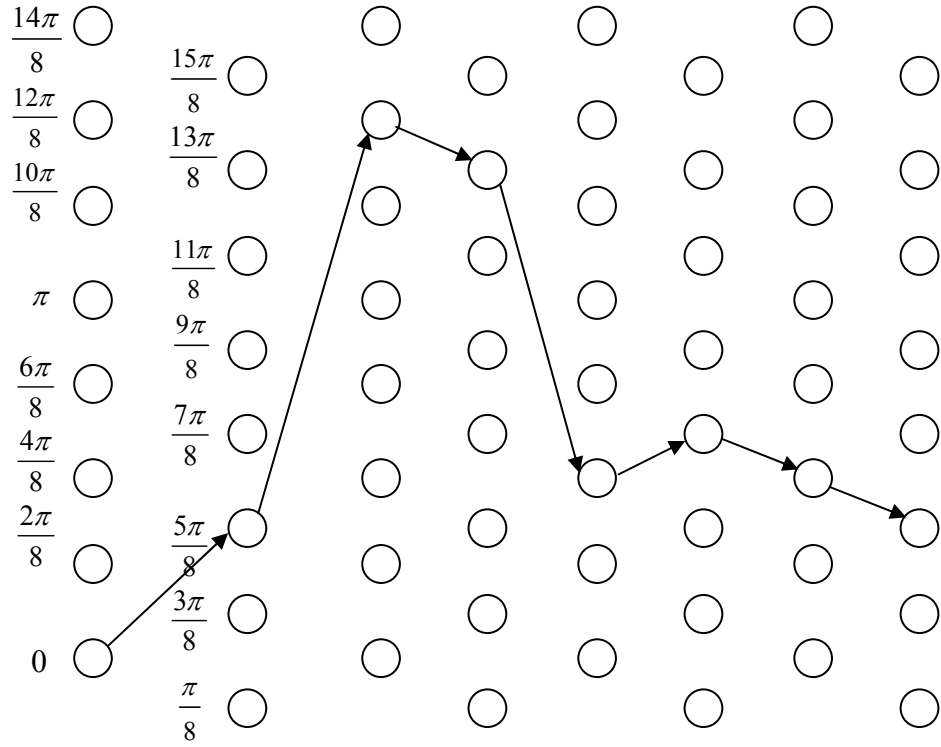
Band genişliği ölçütü olarak, işaretin gücünün %99’unun iletildiği band seçilir. Sabit genlikli Dik Faz Kaydırmalı Anahtarlama (Quadrature Phase Shift Keying, QPSK) modülasyonu, hata düzeltme kodları ile birleştirildiğinde, düşük enerjili ve yüksek band genişlikli bölgede yer alır. Hem enerji hem de band genişliği verimlilikli modülasyon türleri ele alındığında, bunların en büyük alt sınıfını sabit zarflı Sürekli faz modülasyon (Continuous Phase Modulation, CPM) tekniği oluşturur. Bu modülasyon tekniği,

QPSK' ya göre, daha dar ana spektral kulakçığa ve daha düşük yan spektral kulakçıklara sahip olup, aynı E_b / N_0 oranı için daha iyi hata başarımı sağlayabilmektedir.

Faz modülasyon fonksiyonunun tüm işaretleşme aralığında zamanın doğrusal bir fonksiyonu olması durumunda CPM, Sürekli Faz Frekans Kaydırmalı Anahtarlama adını alır. CPM modülatörü, durumları işaretleşme aralığının başında birikmiş fazla tanımlanan ve çıkışları da o anki durumla birlikte giriş sembolüne bağlı olan sonlu durumlu bir makine olarak görülebilir. CPM'in sonlu durumlu makine tanımı bize modülasyon sürecini Şekil 2.23'deki gibi faz kafesinde ilerleyen bir yol gibi izleme kolaylığı sağlar. CPM'in faz sürekliliği yüksek spektral etkinlik sağlar. Spektral etkinlik daha düzgün faz modülasyon fonksiyonu seçilmesi ile arttırılabilir. Örnek bir faz modülasyon fonksiyonu Şekil 2.22'de verilmiştir.



Şekil 2.22 Faz modülasyon fonksiyonu



Şekil 2.23 Faz kafesi

2.4.1 CPM Tekniğinin Tanıtılması

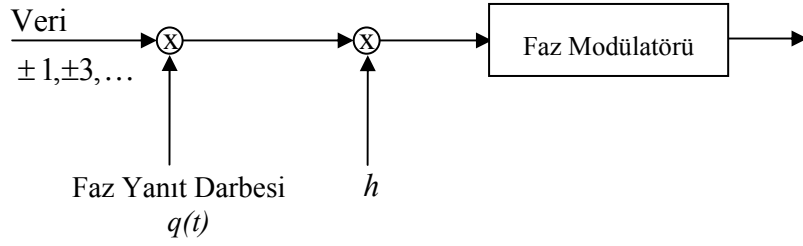
Sabit zarflı sürekli faz modülasyonlu bir işaret,

$$s(t, d) = (2E_s / T)^{1/2} \cos(2\pi f_c t + \Phi(t, d) + \Phi_0) \quad (2.77)$$

biçimindedir. Burada E_s simge başına enerjiyi, T simge süresini, f_c taşıyıcı frekansını ve d ise veri simgelerini gösterir. CPM işarete, bilgiyi taşıyan $\Phi(t, d)$ fazı, n . simge aralığı $nT \leq t \leq (n+1)T$ boyunca bir katlama işlemiyle bulunur. Bu faz, $q(t)$ faz yanıt darbesinin biçimine bağlıdır. Her bir veri simgesine yanıt, toplam işaret fazına etki eder. Bu faz işlevi,

$$\Phi(t, d) = 2\pi h \sum_{i < n} d_i q(t - iT) \quad (2.78)$$

biçiminde tanımlanır. Şekil 2.24'te CPM işaretlerinin üretilmesi şematik olarak gösterilmektedir. CPM'in genel tanımı Sundberg [37] tarafından verilmiştir. Çoğu zaman, bir CPM sistemi $q(t)$ 'nin türevi olan ani frekans darbesi $g(t)$ ile tanımlanır.



Şekil 2.24 CPM işaretlerinin üretilmesi

$g(t)$ frekans darbesi, $0 \leq t \leq LT$ zaman aralığı boyunca sıfırdan farklı olup $q(t) = \int_{-\infty}^{\infty} g(t)dt = 1/2$ 'dir. Burada L , simge aralığının uzunluğudur. D_i , M düzeyli veri simgeleri olup, $\pm 1, \pm 3, \dots, \pm(M-1)$ değerlerini alır. h sabiti, modülasyon indisi olup $\Phi(t, d)$ fazına etki eder. Analog frekans modülasyonundaki β , faz sapma indisine benzer bir rol oynar. Genel olarak yüksek h değerleri için modülasyon tekniği, daha iyi hata başarımı sağlar ancak işaret spektrumu daha geniş bir alanı kaplar. Maksimum faz değişimi $(M-1)\pi h$ 'dır. Enerji ve band genişlikleri, bit başına enerji $E_b = E / \log_2 M$ ve bit başına iletim süresi $T_b = T / \log_2 M$ cinsinden hesaplanmaktadır. Değişik frekans darbeleri $g(t)$, h ve M değerleri ile çok sayıda CPM sistemi tanımlanabilir.

CPM işaretlerinin fazının sürekli olması, bu darbelere bellek katar, bilgiyi taşıyan faz işlevi $\Phi(t, d)$, tüm işaretleşme aralıklarında süreklidir. Veri sistemleri ilişkisizken bile, $L > 1$ seçilerek CPM işaretlere bellek katılabilir. Bu modülasyon tekniğine, *kısmi yanıtli modülasyon* denir. *Tam yanıtli modülasyon* tekniğinde, $L \leq 1$ 'dir. CPM işaretler, çeşitli h değerlerine göre farklı kafes yapılarına sahiptirler. $h = l/p$ için (burada l ve p tam sayılardır) $nT \leq t \leq (n+1)T$ aralığı boyunca $\Phi(t, d)$ fazı,

$$\Phi(t, d) = 2\pi h \sum_{i=n-L+1}^n d_i \cdot q(t - iT) + \theta_n \quad (2.79)$$

olup,

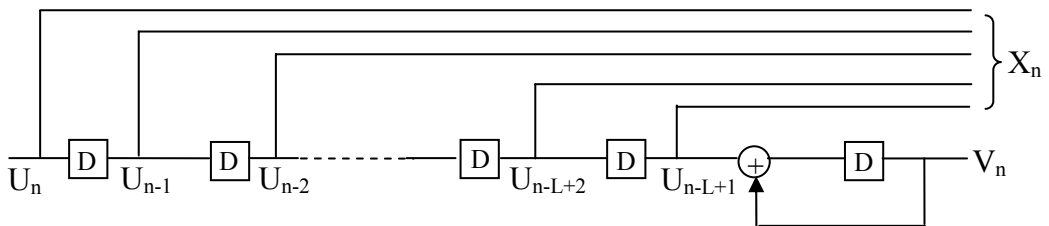
$$\theta_n = \pi h \sum_{i=-\infty}^{n-L} d_i \pmod{2\pi} \quad (2.80)$$

biçiminde tanımlanır. θ_n , n işaretleşme aralığına kadar biriken fazı tanımlar ve $2p$ farklı değere sahiptir. Eşitlik (2.78)'in kalan bölümü, $d_{n-1}, \dots, d_{n-L+1}$ verisine bağlıdır. Böylece, bir CPM kafesindeki durumların sayısı $\theta_n, d_{n-1}, \dots, d_{n-L+1}$ durum değişkenlerinin farklı değerlerinin sayısıdır. Toplam, $2p.M^{L-1}$ durum vardır. $L=1$ için sürekli fazlı frekans kaydırmalı anahtarlama (CPFSK, Continuous Phase Shift Keying) ve diğer tam yanıtı CPM yöntemlerinde $2p$ durum vardır.

2.4.2 Sürekli Faz Modülasyonu İçin Ayrıştırma Tekniği

CPM sistemlerde bilgiyi taşıyan faz işlevinin sürekli olması ve tüm faz durumlarını içeren bir faz durum kafesi yardımıyla olası işaretlerin gösterilebilmesi, katlamalı kodlamaya benzer bir bellek yapısının varlığını işaret eder. Rimoldi [47], bir CPM sisteminin, sürekli fazlı bir kodlayıcı (Continuous Phase Encoder, CPE) ve onu izleyen belleksiz bir modülatöre (Memoryless Modulator, MM) ayrıştırılabileceğini göstermiştir. Böyle bir ayrıştırmanın iki yararı vardır: Birincisi, kodlama işleminin modülasyondan bağımsız ele alınmasını sağlamasıdır. CPE, farklı kodlayıcı tasarımında, farklı CPM yapılarına ve farklı uygun değer kodçözücülerin oluşumuna izin verir. CPE zamanla değişmiyorsa ve doğrusalsa, bir kodlayıcıdır. CPE ile bir dış kodlayıcının art arda sıralanmış tek bir katlamalı kodlayıcıya indirgenebilir. Dış kodlayıcı, modülasyonlu farklı çıkış işaretleri arasındaki Öklid uzaklığını minimum yapacak biçimde tasarlanabilir. CPM'in ayrıştırılmasının ikinci yararı ise MM'in kodlamadan yalıtılması sonucunda, modülatörün, kanalın ve demodülatörün, tek bir ayrık belleksiz kanalla modellenmesini sağlamasıdır.

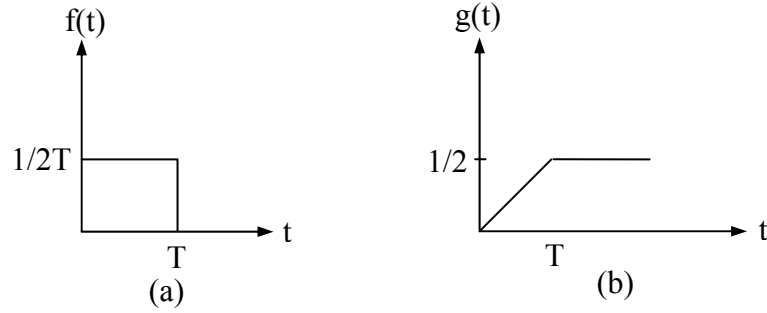
Herhangi bir CPM sistem, CPE ve bir MM olarak ayrıştırılabilir. CPE, modulo- p toplayıcıları ve L gecikme elemanı içeren zamanla değişmeyen doğrusal ardışıl bir devredir. Burada p , modülasyon indisinin paydası ve L , faz yanıtı $q(t)$ 'nin değişen kısmı altında kalan işaretleşme araklılarının sayısıdır.



Şekil 2.25 CPE'nin blok diyagramı

2.4.3 Sürekli Fazlı Frekans Kaydırmalı Anahtarlama (CPFSK)

Sürekli faz modülasyonuna ilişkin ani frekans darbe yanıtı işlevi $f(t)$ 'nin, Şekil 2.26 (a)'da gösterildiği gibi 1REC, yani T süreli bir dikdörtgen darbe ve dolayısıyla faz yanıtının Şekil 2.26(b)'deki gibi olduğu durum, sürekli fazlı frekans kaydırmalı anahtarlama (CPFSK) olarak adlandırılır. CPFSK, bilinen frekans kaydırmalı anahtarlama (FSK) modülasyonunun sürekli fazlı biçimidir. Ancak, faz sürekliliği özelliğinden dolayı daha dar bir frekans spektrumu sağlamasının 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 gerçekleşmemesi faz atlamalarını artıracak 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 2.26 (a) CPFSK'ya ait $f(t)$ (b) $f(t)$ 'nin integrali $g(t)$

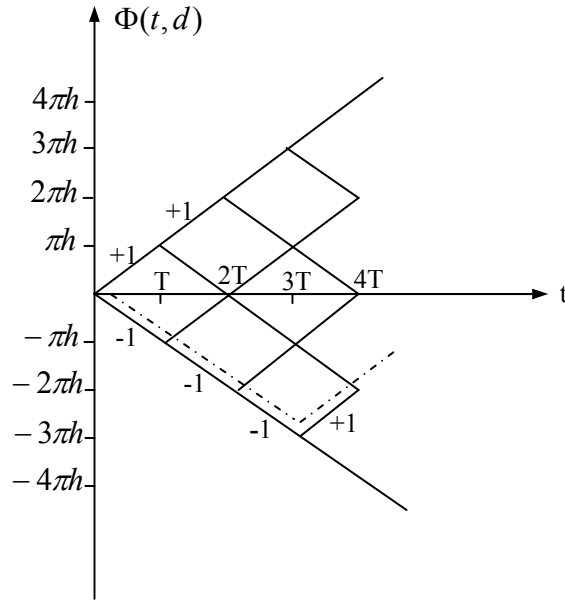
$nt \leq t \leq (n+1)T$ zaman aralığında CPFSK işaretleri

$$\begin{aligned}
 s(t, d) &= \sqrt{\frac{2E}{T}} \cos \left[2\pi f_c t + 2\pi h \sum_{k=-\infty}^{\infty} d_k \frac{(t-kT)}{2T} \right] \\
 &= \sqrt{\frac{2E}{T}} \cos \left[2\pi f_c t + \pi h d_n \frac{(t-nT)}{T} + \varphi_n \right]
 \end{aligned} \tag{2.81}$$

olarak yazılabilir. Burada φ_n ,

$$\varphi_n = \left[\pi h \sum_{k=-\infty}^{n-1} d_k \right]_{\text{mod } 2\pi} \quad (2.82)$$

eşitliği ile ifade edilen ve $(n-1)T$ anına kadar toplamsal olarak oluşan faz değerlerini gösteren bellek terimidir. J çift ise $\varphi_n \in \{0, \pi J / P, 2\pi J / P, \dots, (P-1)\pi J / P\}$ 'dir. CPFSK işaretlerinin AM, PSK ve QAM modülasyonlarında olduğu gibi işaret uzayında gösterilimi mümkün olmamaktadır. Çünkü CPFSK'ya ilişkin işaret uzayı 3'ten fazla boyut içerebilmektedir. Bu nedenle CPFSK işaretlerinin simgelenmesinde bir alternatif olarak (2.80) eşitliğindeki faz terimi $\Phi(t, d)$, zamanın bir işlevi olarak "faz ağacı" yardımıyla gösterilmektedir. Örneğin, başlangıç fazı ve zamanı sıfır alınarak ikili CPFSK'ya ilişkin faz ağacı Şekil 2.27'deki gibi verilebilir. Görüldüğü gibi, her T modülasyon aralığında $d_n = \pm 1$ değerine göre düzgün bir faz değişimi vardır. Ayrıca şekil üzerinde örnek olarak verilen $a = \{-1, -1, -1, +1\}$ bilgi dizisine ilişkin faz değişim yolu kesikli çizgilerle gösterilmiştir.

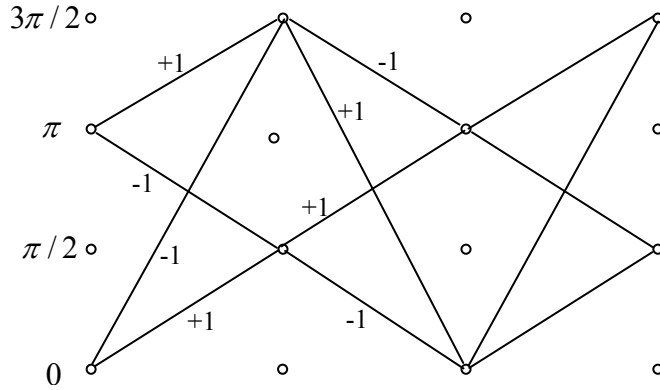


Şekil 2.27 İkili CPFSK faz ağacı

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

$$\Phi(t, d) = [\Phi(t, d)]_{\text{mod } 2\pi} \quad (2.83)$$

ifadesi ile “fiziksel faz” tanımı yapılmaktadır. Bu durumda CPFSK modülasyonunun fiziksel faz kafesi çizilebilir. Örneğin, CPFSK modülasyonunda $M=2$ ve $h=1/2$ seçilmiş durum için elde edilen FFSK modülasyonunun fiziksel faz kafesi Şekil 2.28’deki gibidir.



Şekil 2.28 FFSK' nın fiziksel faz kafesi diyagramı

Şekilden görüldüğü gibi, FFSK modülasyonunun kafes yapısı zamanla değişen türdendir. Kafes yapısının zamanla değişen türden olması, sürekli faz modülasyonunun bir kodlayıcı yardımıyla kodlanması durumundaki analizleri (örneğin hata başaım 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 [25] tarafından önerilmiştir ve "eğik fazlı CPFSK modülasyonu" olarak adlandırılmaktadır. Bu amaçla (2.77) ifadesi,

$$s(t, d) = \sqrt{\frac{2E}{T}} \cos[2\pi f_1 t + \Psi(t, d)] \quad t \geq 0 \quad (2.84)$$

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

$$\Psi(t, d) = \Phi(t, d) + \frac{\pi h(M-1)t}{T} \quad (2.85)$$

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

$$\Psi(t, d) = [\Psi(t, d)]_{\text{mod } 2\pi} \quad (2.86)$$

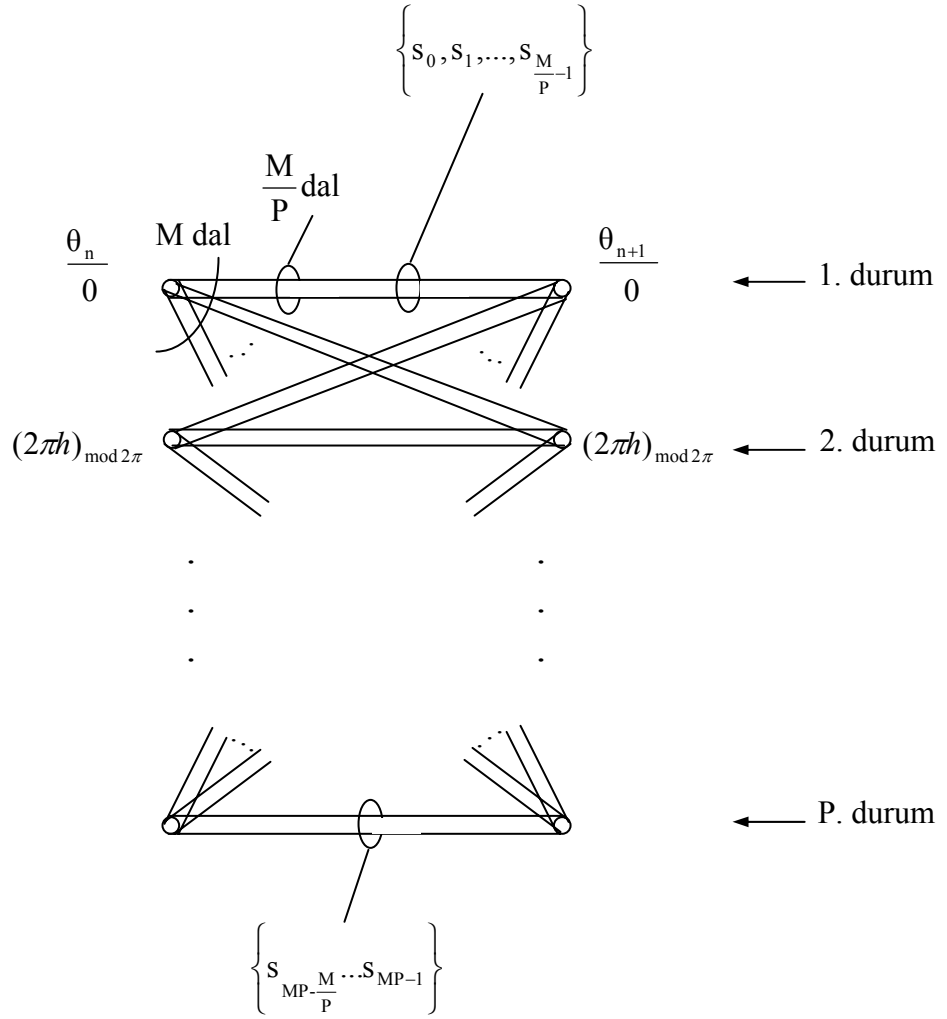
şeklinde yazılabilir. (2.84) ifadesi yeniden düzenlenirse,

$$s(t, d) = \sqrt{\frac{2E}{T}} \cos \left[2\pi f_1 t + 2\pi h \beta_n \frac{(t - nT)}{T} + \theta_n \right] \quad (2.87)$$

yazılabilir. (2.87) ifadesi, eğik fazlı CPFSK modülasyonuna ilişkin genel işaret biçimini göstermektedir, $\theta_n, t = nT$ anında fazın başlangıç değeri, yani modülasyonun faz sürekliliğini sağlayan bellek terimidir. Ayrıca,

$$\theta_{n+1} = (\theta_n + 2\pi h \beta_n)_{\text{mod } 2\pi} \quad (2.88)$$

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. Buradan yararlanarak (2.87) ifadesi dikkatle incelenecek olursa, bu ifadenin θ_n faz değerlerini durum olarak alan ve zamanla değişmeyen sonlu (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 2.29 M 'li CPFSK, $h=J/P$ 'nin fiziksel eğik faz kafesi (işaret kümesi)

Elde edilen kafes diyagramı Şekil 2.29'daki gibidir. Kafes diyagramda her durumdan M adet dal çıkmaktadır. s_i, β_n ve θ_n değerlerine göre elde edilebilecek MP 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 2.29'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

$$d^2(\beta, \beta') = \left\{ \begin{array}{ll} 2E \left[1 - \frac{\sin \Delta_{\Psi_{n+1}} - \sin \Delta_{\Psi_n}}{\Delta_{\Psi_{n+1}} - \Delta_{\Psi_n}} \right], & \Delta_{\Psi_{n+1}} \neq \Delta_{\Psi_n} \\ 2E [1 - \cos \Delta_{\Psi_n}], & \Delta_{\Psi_{n+1}} = \Delta_{\Psi_n} \end{array} \right\} \quad (2.89)$$

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

$$\Delta_{\Psi_n} = \Psi(nT, \beta) - \Psi(nT, \beta') \quad (2.90)$$

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

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

2.4.3.1 4'lü CPFSK, $h=1/2$

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

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

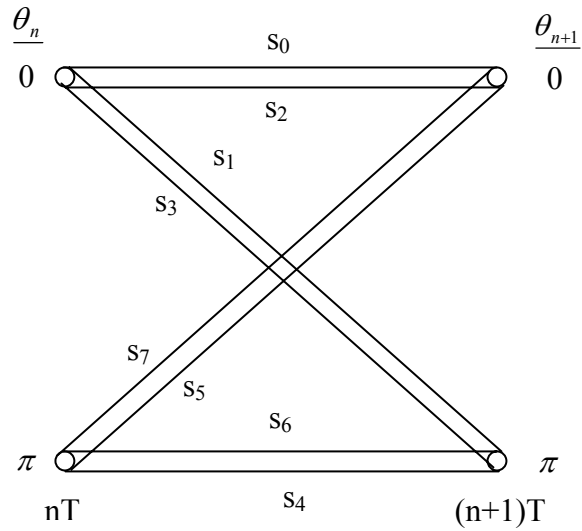
$$\theta_n \in \{0, \pi\}$$

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

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

Tablo 2.2 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

Şekil 2.30 4'lü CPFSK, $h=1/2$ 'nin işaret kümesi

2.4.3.2 8'li CPFSK, $h=1/2$

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

$$\alpha_n \in \{-7, -5, -3, -1, +1, +3, +5, +7\}$$

$$\Rightarrow \beta_n \in \{0, 1, 2, 3, 4, 5, 6, 7\}$$

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

$R_s=3$ bit/simge

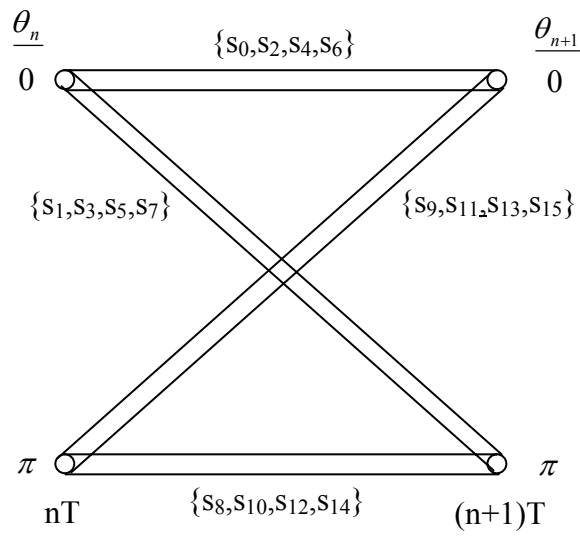
$$d_{sE}^2 = 2E$$

Tablo 2.3 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 2.3 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



Şekil 2.31 8’li CPFSK, h=1/2’nin işaret kümesi

2.4.3.3 16'li CPFSK, $h=1/2$

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

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

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

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

$$R_s = 4 \text{ bit/simge}$$

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

Tablo 2.4 16'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 2.4 16'lı CPFSK, h=1/2 değerleri (Devam)

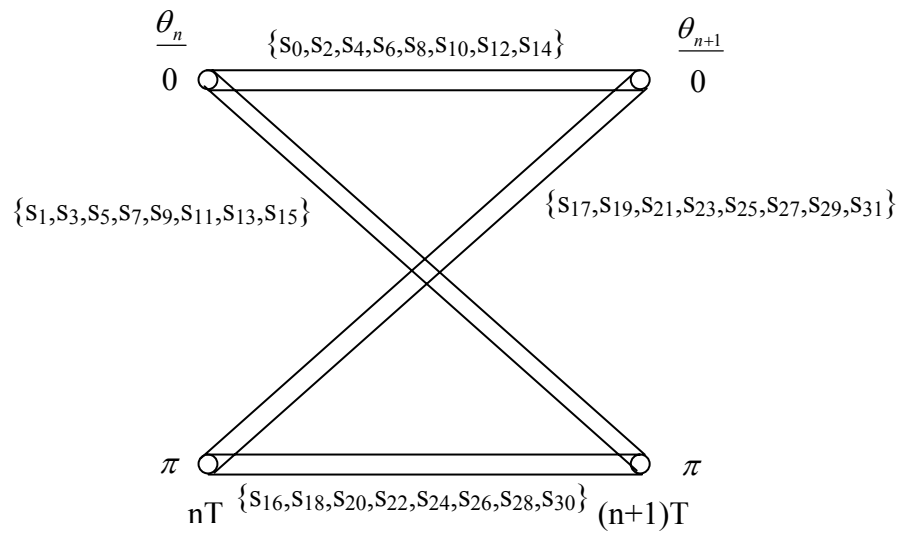
0	8	$8\pi \frac{(t-nT)}{T}$	0	8π	s ₈	0
	9	$9\pi \frac{(t-nT)}{T}$	0	9π	s ₉	π
	10	$10\pi \frac{(t-nT)}{T}$	0	10π	s ₁₀	0
	11	$11\pi \frac{(t-nT)}{T}$	0	11π	s ₁₁	π
	12	$12\pi \frac{(t-nT)}{T}$	0	12π	s ₁₂	0
	13	$13\pi \frac{(t-nT)}{T}$	0	13π	s ₁₃	π
	14	$14\pi \frac{(t-nT)}{T}$	0	14π	s ₁₄	0
	15	$15\pi \frac{(t-nT)}{T}$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

Tablo 2.4 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 2.4 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



Şekil 2.32 16'lı CPFSK h=1/2' nin işaret kümesi

3. MALZEME VE YÖNTEM

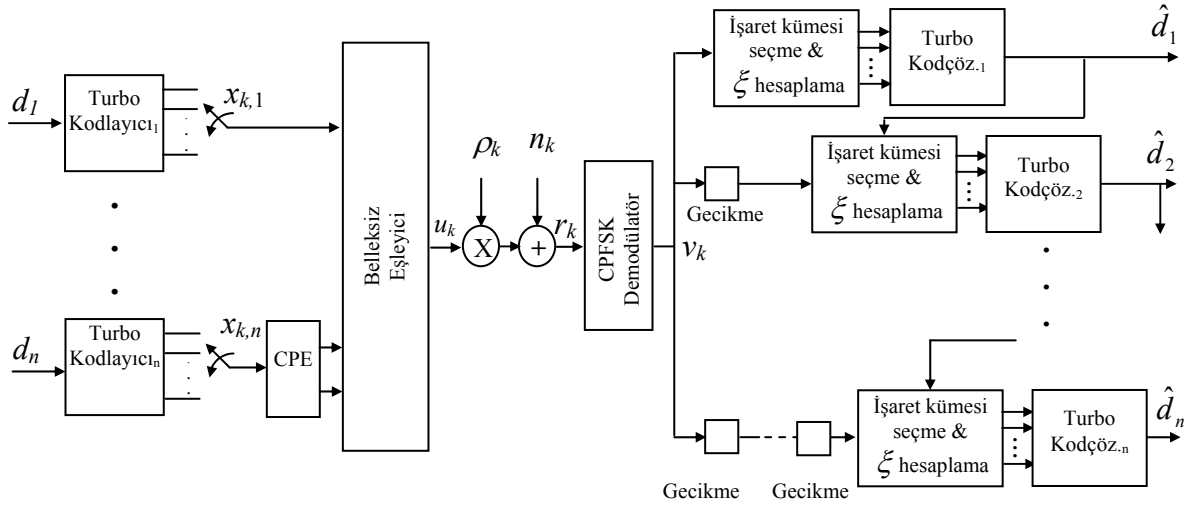
3.1 SÜREKLİ FAZ MODÜLASYONUNUN ÇOK DÜZEYLİ TURBO KODLANMASI

Sürekli faz modülasyonunun çok düzeyli turbo kodlanmasına ilişkin önerilen sistem Şekil 3.1’de görülmektedir. Önerilen sistem, Bölüm 2.2.2’de detayları verilen Imai-Hirakawa tipi çok düzeyli kodlama temeline dayanmaktadır. Herbir düzeyde Turbo kod kullanılabileceği gibi kodçözme karmaşıklığının azaltılması amacıyla bazı düzeylerde daha basit kodlayıcılar da kullanılabilir. Çok düzeyli kodlamanın avantajı değişik kod oranları için çok basit tasarım olanağı sunmasıdır. Her bir düzeyde kullanılan kodlayıcı oranları değiştirilerek değişik kod oranlarına sahip sistem tasarımları kolayca yapılabilir. Tezde sürekli faz modülasyonlu işaretlerin faz sürekliliğinin sağlanması amacıyla sürekli faz modülasyonuna ilişkin “ayırıştırma tekniği” kullanılmıştır. Ayırıştırma tekniğine göre sürekli faz modülatörü, Sürekli Faz Kodlayıcısı (CPE) ve Belleksiz Eşleyici (MM) olarak iki parçaya ayrılabilir. Sürekli faz kodlayıcısı (2.78)’de belirtilen faz teriminin karşılığıdır. Sürekli faz kodlayıcısı, en basit şekilde klasik katlamalı kodlayıcılarla gerçekleştirilebileceği Bölüm 2.4’de açıklanmıştır. Modülasyon indisinin $1/2$ olması ($h=1/2$) sürekli faz modülasyonlu işaretlerin ya “0” yada “ π ” fazından başlayabileceğini ifade eder. Sürekli faz kodlayıcısının belleğinin sıfır durumu “0” fazını bir olması ise “ π ” durumunu gösterir. Bir sonraki işaret belleğin durumuna göre bir önceki işaretin bitiş fazından başlar böylece CPM’in faz sürekliliği sağlanmış olur. Sürekli faz kodlayıcısı son düzeydeki kodlayıcıdan sonra seri olarak yerleştirilmiştir.

3.1.1 Kodlayıcı Yapısı

İletilecek bilgi bit dizisi önce n düzey sayısını göstermek üzere seri-paralel çeviriciyle n tane paralel bit dizisine çevrilir. Herbir düzeydeki kodlayıcı bir paralel diziyi kodlar. Kodlayıcı çıkışlarına delikleme (puncturing) işlemi uygulandıktan sonra son düzeydeki dizi sürekli faz kodlayıcısından geçirilerek iletilecek sürekli faz frekans kaydırmalı anahtarlama işaretleri küme bölmeleme tekniğiyle elde edilir. Küme bölmeleme tekniğine göre birinci düzey çıkışı olan $x_{k,1}$ biti işaret kümesini ikiye böler. $x_{k,1}=0$ ise birinci işaret alt kümesi, $x_{k,1}=1$ ise ikinci işaret alt kümesi seçilir. $x_{k,2}$ ikinci düzey kodlayıcı çıkışıdır ve bir önceki düzeyde seçilen alt kümeyi ilk düzeydekine benzer

şekilde tekrar ikiye böler. Küme bölmeleme işlemi bu şekilde son düzeye kadar devam eder. Son düzeyde sadece kalan iki adet işaretten biri CPE'nin çıkışına göre seçilerek kanala gönderilir. Küme bölmeleme işlemi belleksiz eşleyici tarafından gerçekleştirilir.



Şekil 3.1 Sürekli faz modülasyonunun çok düzeyli turbo kodlanmasına ilişkin blok diyagram

3.1.2 Kodçözücü Yapısı

Turbo kodçözücülerde, her kodçözme ünitesi için iki adet Markov sürecin birleşik kestirimi kullanılmaktadır. Teoride olduğu gibi, bir adet Markov süreci ile de modellenebilirdi. Fakat böyle bir modelleme genellikle karmaşık ve hesaplamaların kontrol edilemeyeceği kodçözücü algoritmalarından meydana gelmektedir.

Turbo kodçözücüde aynı veri setine sahip iki tane Markov sürecinin olduğundan bahsetmiştik. Böylece kestirim iteratif olarak iki kodçözücü arasında bilgilerin paylaşımı ile iyileştirilir. Daha açık ifade ile bir kodçözücünün çıkışı, diğer kodçözücüde giriş bilgisi olarak kullanılmaktadır. Herbir kodçözücünün çıkışı sert-bit (hard-bit) formunda olduğu için bilgilerin paylaşımında çok az bir avantaj sağlamaktadır. Bununla birlikte, kodçözücülerde yumuşak-bit (soft-bit) kararları üretilirse, çok iterasyonlu kodçözme neticesinde önemli bir performans kazancı elde edilmektedir.

Kanaldan alınan gürültülü işaretler (3.1)'deki gibi gösterilirse;

$$r_k = \rho_k u_k + n_k \quad (3.1)$$

r_k alıcıya gelen bozulmuş işaret, u_k vericiden gönderilen M 'li CPFSK modülasyonlu işaret, ρ_k sönümlleme katsayısı ve n_k ise k anındaki Gauss gürültüsüdür. MAP algoritması, her bitin sonsal olasılık değerini mükemmel bir performans ile bulmaktadır. s_k , kodlayıcının k . kodlama adımındaki durumunu ve st ise kodlayıcı düzeyini ifade etmek üzere, $\bar{\gamma}^{st}(s_k \rightarrow s_{k+1})$ dal metriğinin $\gamma^{st}(s_k \rightarrow s_{k+1})$ doğal logaritmasını gösterebilir. Böylece,

$$\bar{\gamma}^{st}(s_k \rightarrow s_{k+1}) = \ln \gamma^{st}(s_k \rightarrow s_{k+1}) \quad (3.2)$$

ve

$$\ln P[d_k] = z_k d_k - \ln(1 + e^{z_k}) \quad (3.3)$$

z_k ise her bir kodçözme aşamasında bulunan turbo kodçözücünün içindeki birinci kodçözücünün çıkışı olup diğer kodçözücüye giriş olarak uygulanan veridir. Önerilen yapıdaki turbo kodlayıcılar ikili turbo kodlayıcı olduğundan kanaldan gelen bozulmuş işaretlerin turbo kodçözücüye girmeden önce ikili faz kaydırmalı anahtarlama (BPSK) işaretlerine indirgenmesi gerekmektedir. Bu indirgeme işlemi için (3.4) ve (3.5)'teki eşitlikler geliştirilmiştir. Burada yapılan işlem kanaldan gelen işaretlerin "0" yada "1" olma olasılıklarının bulunması ve bu olasılıkların $\{-1,1\}$ aralığına normalize edilmesidir.

$$P_{k,0}^{st} = \sum_{j=0}^{M/(2 \cdot st)-1} \frac{1}{(r_k - u_{0,j}^{st})^2} \quad (3.4a)$$

$$P_{k,1}^{st} = \sum_{j=0}^{M/(2 \cdot st)-1} \frac{1}{(r_k - u_{1,j}^{st})^2} \quad (3.4b)$$

$P_{k,0}^{st}$ ve $P_{k,1}^{st}$, k . kodlama adımındaki alınan işaretin sırasıyla "0" ve "1" olma olasılıklarıdır. st ise kodçözme aşaması olup, değişim aralığı $st \in \{1, 2, \dots, \log_2 M\}$ arasında olmaktadır. Alınan işaretin "0" olma olasılığı (3.4a)'daki gibi alınan bozulmuş işaretin o aşamaya ilişkin "0" alt kümesinde bulunan bütün işaretlerin $(u_{0,j}^{st}) j \in \{1, 2, \dots, (M/2 \cdot st)-1\}$ metriklerinin bulunarak toplanması şeklinde yapılmıştır. "1" olma olasılığı da benzer şekilde alınan işaretin "1" alt kümesinde bulunan bütün işaretlerin

$(u_{1,j}^{st})$ $j \in \{1, 2, \dots, (M/2 \cdot st)-1\}$ kullanılarak hesaplanan bütün metriklerinin toplamı şeklinde bulunur.

$$\xi_k^{st,q} = 1 - \frac{2 \cdot P_{k,0}^{st}}{P_{k,0}^{st} + P_{k,1}^{st}} \quad (3.5)$$

Turbo kodun girişi ξ_k bulunduktan sonra eşitlik (3.3)'ün son durumu;

$$\bar{\gamma}^{st}(s_k \rightarrow s_{k+1}) = \ln P[d_k] - \frac{1}{2} \ln(\pi N_0 / E_s) - \frac{E_s}{N_0} \sum_{q=0}^{n-1} [\xi_k^{st,q} - (2x^q - 1)]^2 \quad (3.6)$$

şeklinde olur. Şimdi $\bar{\alpha}(s_k)$, $\alpha(s_k)$ nin doğal logaritması olsun

$$\bar{\alpha}^{st}(s_k) = \ln \alpha^{st}(s_k) \quad (3.7)$$

$$= \ln \left\{ \sum_{s_{k-1} \in A} \exp[\bar{\alpha}^{st}(s_{k-1}) + \bar{\gamma}^{st}(s_{k-1} \rightarrow s_k)] \right\}$$

burada A , s_k 'lere bağlı s_{k-1} 'lerin kümesini göstermektedir. Şimdi $\bar{\beta}^{st}(s_k)$, $\beta^{st}(s_k)$ 'nin doğal logaritması olsun,

$$\bar{\beta}^{st}(s_k) = \ln \beta^{st}(s_k) \quad (3.8)$$

$$= \ln \left\{ \sum_{s_{k+1} \in B} \exp[\bar{\beta}^{st}(s_{k+1}) + \bar{\gamma}^{st}(s_k \rightarrow s_{k+1})] \right\}$$

burada B , s_k 'lara bağlı s_{k+1} 'lerin kümesini göstermektedir.

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

$$\Lambda_k^{st} = \ln \frac{\sum_{S_1} \exp[\bar{\alpha}^{st}(s_k) + \bar{\gamma}^{st}(s_k \rightarrow s_{k+1}) + \bar{\beta}^{st}(s_{k+1})]}{\sum_{S_0} \exp[\bar{\alpha}^{st}(s_k) + \bar{\gamma}^{st}(s_k \rightarrow s_{k+1}) + \bar{\beta}^{st}(s_{k+1})]} \quad (3.9)$$

burada $S_1 = \{s_k \rightarrow s_{k+1} : d_k = 1\}$ mesaj bitinin “1” olduğu durumdaki durum geçişlerinin kümesi, $S_0 = \{s_k \rightarrow s_{k+1} : d_k = 0\}$ ise mesaj bitinin “0” olduğu durumdaki durum geçişlerinin kümesidir. Son iterasyonda ikinci kodçözücünün çıkışı $\Lambda(2)$ göz önünde bulundurularak sert karar verilir.

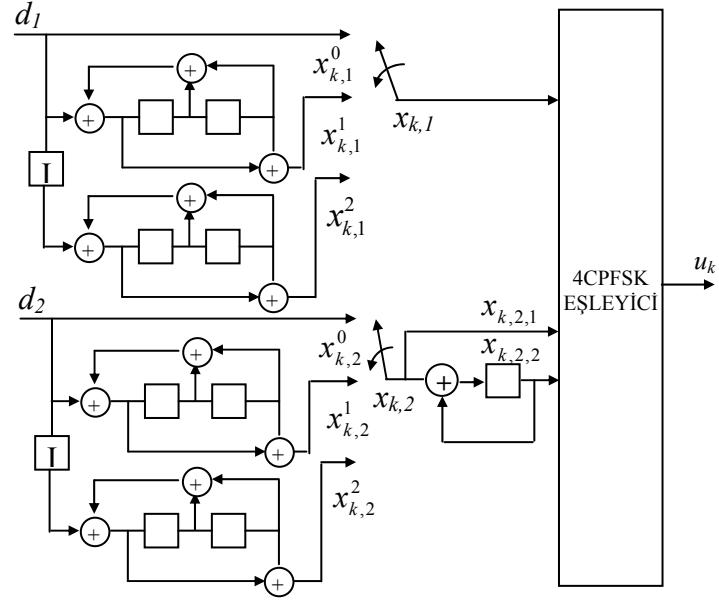
$$\hat{d}_k = \begin{cases} 1, & \Lambda(2) \geq 0 \\ 0, & \Lambda(2) < 0 \end{cases} \quad (3.10)$$

3.1.3 İki Düzeyli Turbo Kodlanmış 4'lü CPFSK Sistemi

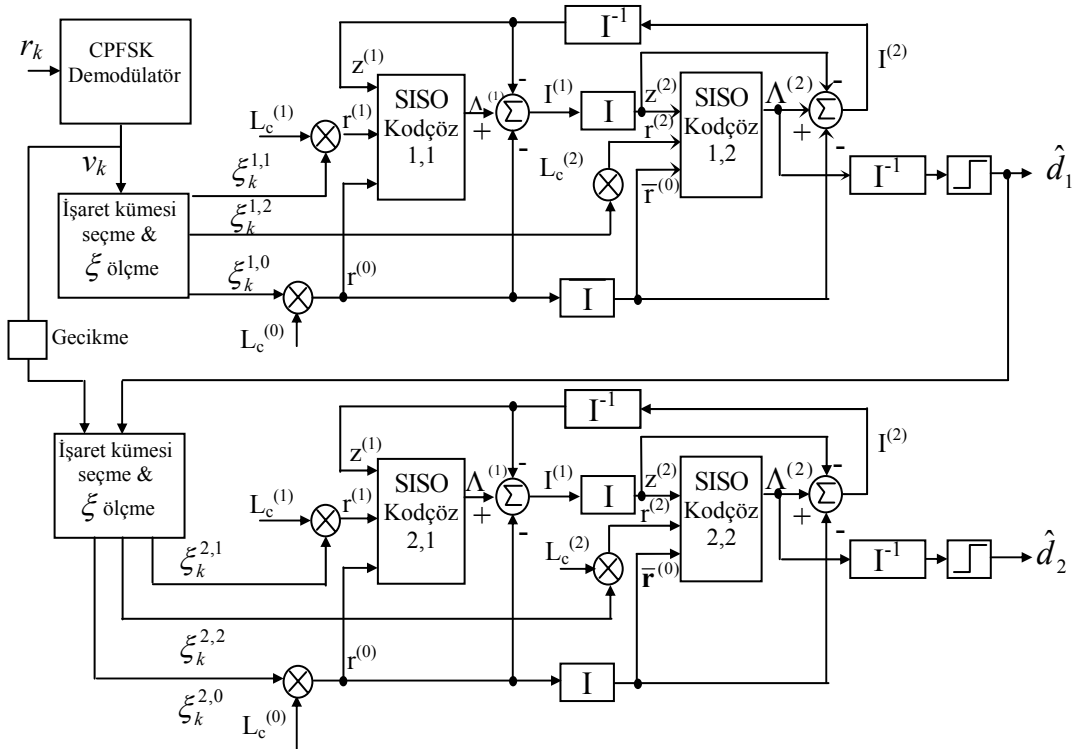
İki düzeyli turbo kodlanmış 4'lü CPFSK sistemi Şekil 3.1'deki blok yapıda $n=2$ seçilmesi durumudur. Herbir düzeydeki turbo kodlarda $M=2$ bellekli 1/3 oranlı geri beslemeli sistematik katlamalı kodlayıcı kullanılmıştır. Bu kodlayıcılara ait geribesleme ve ileri besleme polinomları sırasıyla $g^{(0)}=7$ ve $g^{(1)}=5$ olarak seçilmiştir. D bellek elemanı olmak üzere kodlayıcıların üreteç matrisi (3.11)'de verilmiştir.

$$G(D) = \begin{bmatrix} 1 & \frac{1+D+D^2}{1+D^2} \end{bmatrix} \quad (3.11)$$

Sürekli faz kodlayıcısı ikinci düzey turbo kodlayıcıdan sonra yerleştirilmiştir. Kullanılan sürekli faz kodlayıcısı bir bellekli ve geribeslemeli bir katlamalı kodlayıcıdır. Sistemde kullanılan modülasyon indisi 1/2 seçildiği için bir bellek faz bilgisini tutmakta yeterli olmaktadır. Belleğin sıfır olması durumu işaretin fazının “0”dan başlayacağını bir olması durumu ise “ π ”den başlayacağını göstermektedir. Kodlayıcı yapısı Şekil 3.2'de kodçözücü yapısı ise Şekil 3.3'te gösterilmiştir.

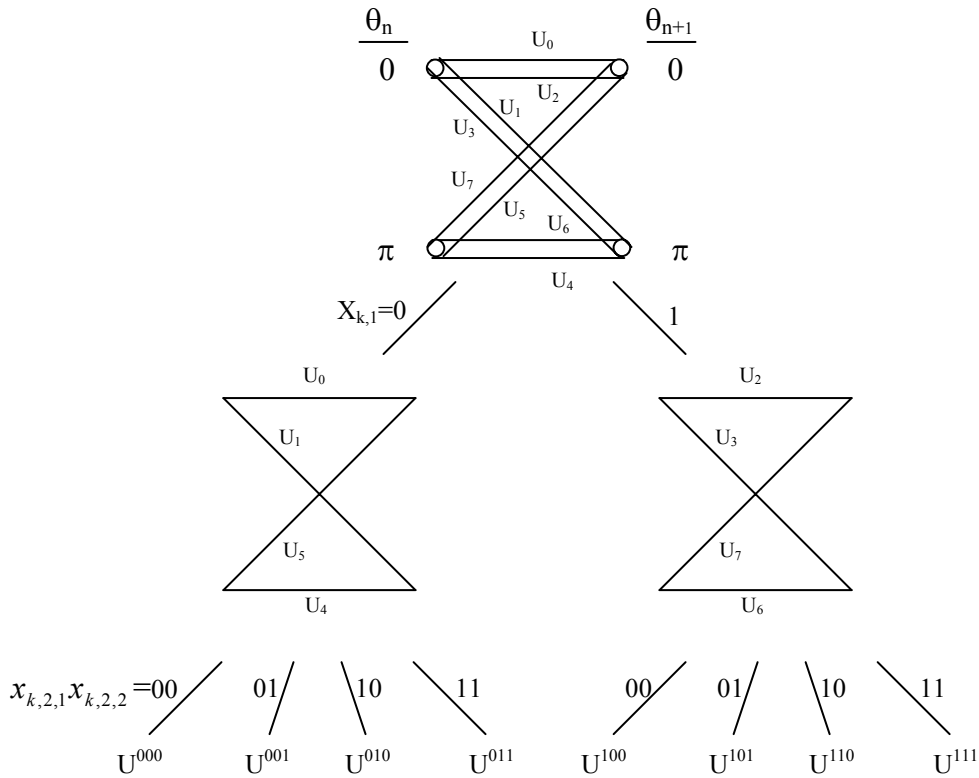


Şekil 3.2 İki düzeyli turbo kodlanmış 4'lü CPFSK sistemi için kodlayıcı yapısı



Şekil 3.3 İki düzeyli turbo kodlanmış 4'lü CPFSK sistemi için kodçözücü yapısı

4'lü CPFSK modülasyonuna ait küme bölmeleme işlemi Şekil 3.4'te görülmektedir. Birinci düzeyde bulunan turbo kodlayıcının çıkışından alınan $x_{k,1}$ biti sekiz elemanlı 4'lü CPFSK işaret kümesini ikiye böler. $x_{k,1}$ biti "0" ise birinci alt küme $\{u_0, u_1, u_4, u_5\}$, "1" ise ikinci alt küme $\{u_2, u_3, u_6, u_7\}$ seçilir. İkinci düzeyde bulunan turbo kodlayıcıdan sonra faz sürekliliğinin sağlanması amacıyla 1/2 oranlı sürekli faz kodlayıcı yerleştirilmiştir. Sürekli faz kodlayıcısının çıkışı olan $x_{k,2,1}$ ve $x_{k,2,2}$ bitleri ise seçilen alt kümedeki dört işaretten hangisinin seçilerek kanala gönderileceğini belirler. $x_{k,2,1}$ ve $x_{k,2,2}$ bitlerinin durumuna göre seçilen alt kümeden hangi işaretin kanala gönderileceği Şekil 3.4'ten kolayca görülmektedir.



Şekil 3.4 4'lü CPFSK modülasyonuna ilişkin küme bölmeleme işlemi

İki düzeyli turbo kodlanmış 4'lü CPFSK sistemine ilişkin kodlayıcı giriş-çıkışları ve işaret eşlemelerinin daha kolay anlaşılabilmesi amacıyla bu bilgiler Tablo 3.1'de verilmiştir.

Tablo 3.1 İki düzeyli turbo kodlanmış 4'lü CPFSK sistemi için kodlayıcı çıkışı-4'lü CPFSK işaret eşlemeleri

Kodlayıcı çıkışları $x_{k,1} \ x_{k,2,1} \ x_{k,2,2}$			4'lü CPFSK işaretleri	4'lü CPFSK $h=1/2$ için işaret uzayı taban katsayıları			
0	0	0	U_0	1	0	0	0
0	0	1	U_1	0	1	0	0
0	1	0	U_4	0	0	1	0
0	1	1	U_5	0	0	0	1
1	0	0	U_2	-1	0	0	0
1	0	1	U_3	0	-1	0	0
1	1	0	U_6	0	0	-1	0
1	1	1	U_7	0	0	0	-1

Kanalda bozulmaya uğrayan işaretler alındıktan sonra Şekil 3.2’de görülen kodçözücü yapısındaki CPFSK demodülatörü ile demodüle edilirler. Demodülatörün çıkışı Tablo 3.1’deki CPFSK işaretleri için işaret uzayı taban katsayılarının bozulmaya uğramış halleridir. Bozulmaya uğramış olan bu katsayılar yine kodçözücü yapısında görülen “İşaret kümesi seçme ve olasılık hesaplama” bloğuna girerler. Bu blokta yapılan işlemi kısaca şu şekilde ifade edebiliriz; öncelikle alınan gürültülü işaretin olası tüm işaretlerle olan metrikleri hesaplanır. Bu işlem (3.4a) ve (3.4b) eşitliklerinin paydalarında bulunan ifadelerle gösterilmiştir. $(r_k - u_{0,j}^{st})^2$ ve $(r_k - u_{1,j}^{st})^2$. Burada r_k kanaldan alınan bozulmuş işarete ilişkin taban katsayılarını, $u_{0,j}^{st}$ çok aşamalı kodçözücünün st . kodçözme aşamasına ilişkin j ’ye bağlı olarak olası tüm birinci alt küme işaretlerini, $u_{1,j}^{st}$ ise olası tüm ikinci alt küme işaretlerini göstermektedir. İki düzeyli turbo kodlanmış 4’lü CPFSK sisteminde birinci kodçözme aşamasında her iki alt kümede 4’er adet işaret bulunmaktadır. Bu iki alt kümeye ait metrikler yukarıda ifade edilen şekilde hesaplanır. Metriklerin tersi alınırsa alınan işaretin o işaret olma olasılıkları bulunur. (3.4a) eşitliğine göre bu olasılıkların birinci alt kümeye ait olanları toplanırsa birinci

düzeydeki kodlayıcı çıkışı olan $x_{k,l}$ bitinin “0” olma olasılığı, (3.4b) eşitliği uyarınca da ikinci alt kümeye ait olasılıklar toplanırsa $x_{k,l}$ bitinin “1” olma olasılığı bulunmuş olur. Kodçözücünün birinci aşaması için bu işlemi daha açık şekilde aşağıdaki gibi gösterebiliriz. Yani (3.4a) ve (3.4b) eşitliklerinde $st=1$, $M=4$, alınırsa “0” ve “1” olma olasılıkları (3.12a) ve (3.12b)’deki gibi hesaplanır.

$$P_{k,0}^1 = \sum_{j=0}^3 \frac{1}{(r_k - u_{0,j}^1)^2} = \left\{ \frac{1}{(r_k - u_{0,0}^1)^2} + \frac{1}{(r_k - u_{0,1}^1)^2} + \frac{1}{(r_k - u_{0,2}^1)^2} + \frac{1}{(r_k - u_{0,3}^1)^2} \right\} \quad (3.12a)$$

$$P_{k,1}^1 = \sum_{j=0}^3 \frac{1}{(r_k - u_{1,j}^1)^2} = \left\{ \frac{1}{(r_k - u_{1,0}^1)^2} + \frac{1}{(r_k - u_{1,1}^1)^2} + \frac{1}{(r_k - u_{1,2}^1)^2} + \frac{1}{(r_k - u_{1,3}^1)^2} \right\} \quad (3.12b)$$

Buradaki sekiz adet CPFSK işaretinin Tablo 3.1’de gösterilen işaretlerle eşleşmeleri aşağıdaki şekildedir.

$$u_{0,0}^1 = u_0, u_{0,1}^1 = u_1, u_{0,2}^1 = u_4, u_{0,3}^1 = u_5 \text{ ve } u_{1,0}^1 = u_2, u_{1,1}^1 = u_3, u_{1,2}^1 = u_6, u_{1,3}^1 = u_7 \quad (3.13)$$

Kodlayıcı düzeylerinde kullanılan turbo kodlayıcılar ikili turbo kod olduğu için bu kodlayıcılara ait kodçözücülerin girişleri BPSK formatında $\{-1,1\}$ olmalıdır. Yukarıda gösterildiği gibi (3.4a) ve (3.4b) eşitlikleriyle hesaplanan “0” ve “1” olma olasılıkları $P_{k,0}^1$ ve $P_{k,1}^1$ kullanılarak (3.5) eşitliği yardımıyla turbo kodçözücüye girecek olan BPSK formatındaki turbo kodçözücü girişi ξ_k^1 hesaplanır. Kodçözücüde MAP algoritması kullanılarak iteratif şekilde kodçözme işlemi yapılır ve bu işlemin sonucunda sert karar olarak birinci düzeydeki turbo kodlayıcı girişi olan d_1 bitinin kestirilmiş olan $\hat{d}_1$ bulunmuş olur.

İkinci kodçözme aşamasında yine birinci aşamadakine paralel işlemler yapılır. Buradaki tek fark bir önceki kodçözme aşamasının çıkışı olan $\hat{x}_{k,1}$ ’in ikinci kodçözme aşaması için bir yan bilgi olarak kullanılmasıdır. $\hat{x}_{k,1}$, birinci aşamada bulunan kodlayıcı girişi

$\hat{d}_1$ 'nin aynı kodlayıcıyla tekrar kodlanmasıyla elde edilebilir. $\hat{x}_{k,1}$, “0” olarak bulunmuşsa şekil 3.4’te görülen birinci alt küme $\{u_0, u_1, u_4, u_5\}$, “1” olarak bulunmuşsa da ikinci alt küme $\{u_2, u_3, u_6, u_7\}$ seçilir. Birinci alt kümenin seçilmiş olduğunu düşünürsek, ikinci aşama için “0” ve “1” olma olasılıkları yine (3.4a) ve (3.4b) eşitliklerinden aşağıdaki gibi kolayca hesaplanabilir. $st=2, M=4$ alınırsa “0” ve “1” olma olasılıkları (3.14a) ve (3.14b)’deki gibi hesaplanır.

$$P_{k,0}^2 = \sum_{j=0}^1 \frac{1}{(r_k - u_{0,j}^2)^2} = \left\{ \frac{1}{(r_k - u_{0,0}^2)^2} + \frac{1}{(r_k - u_{0,1}^2)^2} \right\} \quad (3.14a)$$

$$P_{k,1}^2 = \sum_{j=0}^1 \frac{1}{(r_k - u_{1,j}^2)^2} = \left\{ \frac{1}{(r_k - u_{1,0}^2)^2} + \frac{1}{(r_k - u_{1,1}^2)^2} \right\} \quad (3.14b)$$

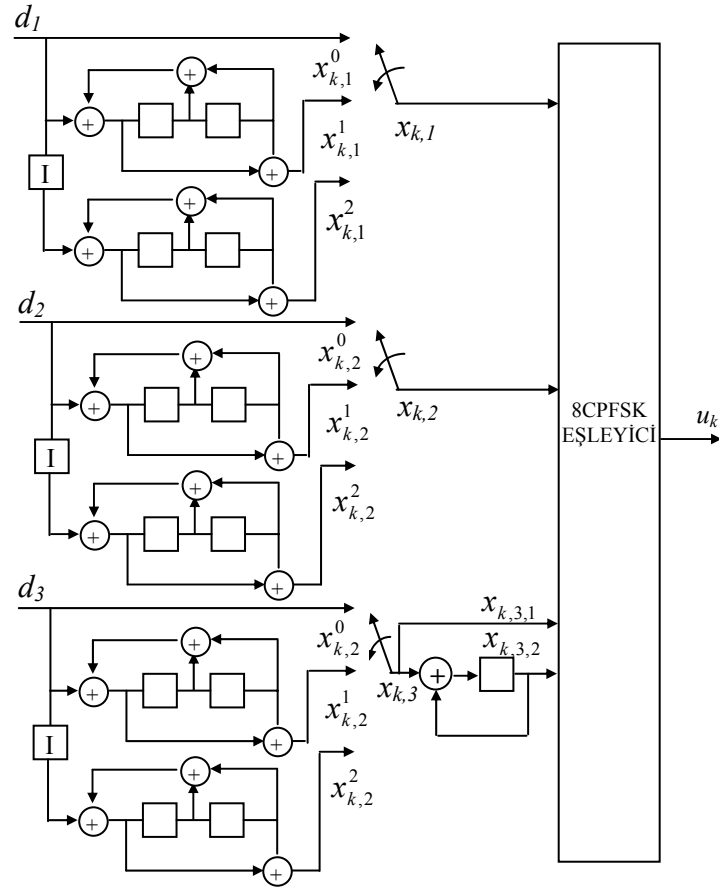
Birinci aşama kodçözücü çıkışı $\hat{x}_{k,1}=0$ olduğu için yukarıdaki olasılık hesaplamalarında yandaki eşitlikler geçerlidir. $u_{0,0}^2 = u_0, u_{0,1}^2 = u_1, u_{1,0}^2 = u_4, u_{1,1}^2 = u_5$. Eğer $\hat{x}_{k,1}=1$ olarak bulunmuş olsaydı ikinci alt küme seçilmiş olacak ve bu eşitlikler $u_{0,0}^2 = u_2, u_{0,1}^2 = u_3, u_{1,0}^2 = u_6, u_{1,1}^2 = u_7$ şeklinde olacaktı.

$P_{k,0}^2$ ve $P_{k,1}^2$ kullanılarak ikinci aşama turbo kodçözücü girişi olan ξ_k^2 hesaplanır. İkinci aşama turbo kodçözücü gerekli işlemleri yaparak ikinci düzey turbo kodlayıcı girişi d_2 'nin kestirilmiş olan $\hat{d}_2$ 'yi bulur. Böylece bir modülasyon aralığı için gönderilen iki bilgi biti ($d_1 d_2$) bulunmuş olur.

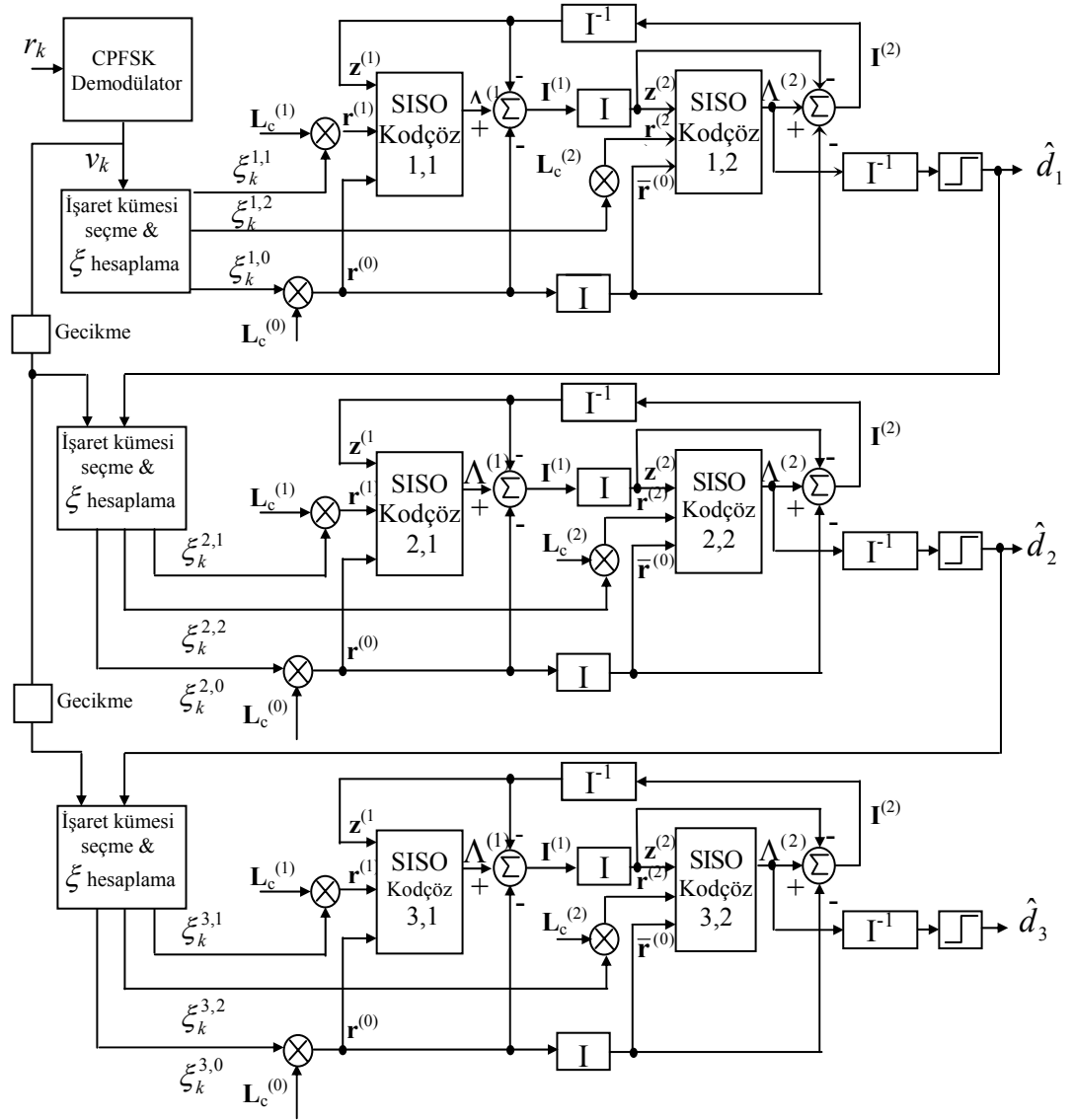
3.1.4 Üç Düzeyli Turbo Kodlanmış 8’li CPFSK Sistemi

Üç düzeyli turbo kodlanmış 8’li CPFSK sistemi iki düzeyli sisteme benzer şekilde genel blok yapının $n=3$ seçilmiş durumudur. Herbir düzeyde kullanılan kodlayıcılar ve en alt düzey olan üçüncü düzeydeki turbo kodlayıcıdan sonra yerleştirilen sürekli faz kodlayıcısı iki düzeyli sistemdekilerle aynı seçilmiştir. İki düzeyli sistemden farklı şekilde kullanılan CPFSK, 4’lü değil 8’lidir. Çünkü üç düzeyli sistemin olası çıkış sembol sayısı 16’dır. 4’lü CPFSK işaret kümesi ise 8 işaretten oluşur. 8’li CPFSK işaret kümesinde ise 16 adet işaret bulunduğu için üç düzeyli sistemde kullanılmaya

uygundur. Üç düzeyli sisteme ilişkin kodlayıcı ve kodçözücü yapıları Şekil 3.5 ve Şekil 3.6’da görülmektedir.

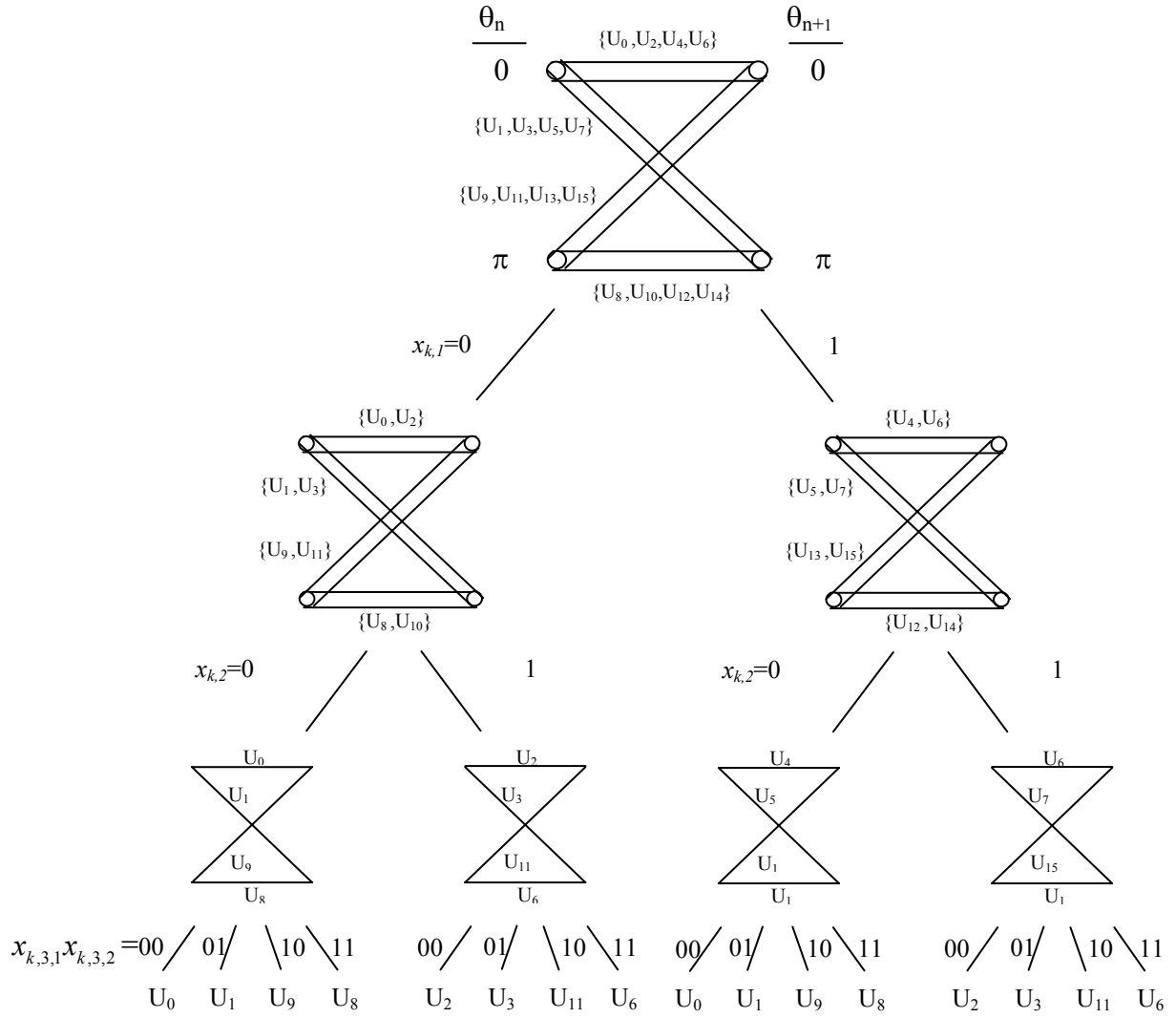


Şekil 3.5 Üç düzeyli turbo kodlanmış 8’li CPFSK sistemi için kodlayıcı yapısı



Şekil 3.6 Üç düzeyli turbo kodlanmış 8’li CPFSK sistemi için kodçözücü yapısı

Üç düzeyli sistemde de modülasyon indisi $h=1/2$ olarak seçildiği için 8’li CPFSK işaretlerinin başlangıç faz değerleri yalnızca iki farklı değer alabilmektedir. Bunlar “0” ve “ π ” fazlarıdır. 8’li CPFSK modülasyonu işaret kümesine ilişkin küme bölme işlemi Şekil 3.7’de görülmektedir.



Şekil 3.7 8’li CPFSK modülasyonuna ilişkin küme bölmeleme işlemi

Küme bölmeleme işleminde üç düzeyli sistemin ilk düzey kodlayıcı çıkışından alınan $x_{k,1}$ biti 16 elemanlı 8’li CPFSK işaret kümesini iki alt kümeye böler. Herbir alt kümede sekiz eleman vardır. Bu iki alt kümeyi üç düzeyli sistemin ikinci düzey kodlayıcı çıkışı olan $x_{k,2}$ biti tekrar ikiye böler. Bu aşamada 4’er elemanlı dört alt küme oluşmuş olur. Üçüncü düzeyde bulunan sürekli faz kodlayıcısının çıkışları olan $x_{k,3,1}$ ve $x_{k,3,2}$ bitleriyle de seçilen ikinci düzey dört elemanlı alt küme içinden hangi işaretin seçilerek kanala gönderileceği belirlenir.

Üç düzeyli turbo kodlanmış 8’li CPFSK sistemine ilişkin kodlayıcı giriş-çıkışları ve işaret eşlemelerine ait bilgiler Tablo 3.2’de detaylı olarak verilmiştir.

Tablo 3.2 Üç düzeyli turbo kodlanmış 8’li CPFSK sistemi için kodlayıcı çıkışı-8’li CPFSK işaret eşlemeleri

Kodlayıcı çıkışları $x_{k,1} \ x_{k,2} \ x_{k,3,1} \ x_{k,3,2}$				8’li CPFSK işaretleri	8’li CPFSK $h=1/2$ için işaret uzayı taban katsayıları							
0	0	0	0	U_0	1	0	0	0	0	0	0	0
0	0	0	1	U_1	0	1	0	0	0	0	0	0
0	0	1	0	U_4	0	0	1	0	0	0	0	0
0	0	1	1	U_5	0	0	0	1	0	0	0	0
0	1	0	0	U_2	0	0	0	0	1	0	0	0
0	1	0	1	U_3	0	0	0	0	0	1	0	0
0	1	1	0	U_6	0	0	0	0	0	0	1	0
0	1	1	1	U_7	0	0	0	0	0	0	0	1
1	0	0	0	U_8	-1	0	0	0	0	0	0	0
1	0	0	1	U_9	0	-1	0	0	0	0	0	0
1	0	1	0	U_{10}	0	0	-1	0	0	0	0	0
1	0	1	1	U_{11}	0	0	0	-1	0	0	0	0
1	1	0	0	U_{12}	0	0	0	0	-1	0	0	0
1	1	0	1	U_{13}	0	0	0	0	0	-1	0	0
1	1	1	0	U_{14}	0	0	0	0	0	0	-1	0
1	1	1	1	U_{15}	0	0	0	0	0	0	0	-1

8’li CPFSK işaretleri Tablo 3.2’den de görülebileceği gibi sekiz boyutla ifade edilebilmektedirler. Tablonun “8’li CPFSK $h=1/2$ için işaret uzayı taban katsayıları” sütunu dikkatle incelenecek olursa buradaki ikinci sekiz işaretin ilk sekiz işaretin negatifleri olduğu kolayca görülebilir. İlk sekiz işaret “0” fazından başlayan işaretler, ikinci sekiz işaretse “ π ” fazından başlayan işaretlerdir.

4. BULGULAR

Bu bölümde tezde önerilen çok düzeyli turbo kodlanmış sürekli faz modülasyonlu sistemlerin hata başarımları incelenmiştir. Hata başarımları eğrileri bilgisayar benzetimi yardımıyla elde edilmiştir. Kanal, toplamsal beyaz Gauss gürültülü (AWGN), ve sönümlemeli (Rician ($K=10$) ve Rayleigh ($K=0$)) olarak modellenmiştir. İzleyen alt bölümde sönümlemeli kanalların detayları verilmektedir.

Birbirinden bağımsız olarak üretilen sönümleme değişkenleri (ρ_n), gönderilen simgeler (u_n) ile çarpılmış ve toplamsal beyaz Gauss gürültüsü (η_n) eklenerek, alıcıya gelen işaret elde edilmiştir. Gürültünün gönderilen işarete karışması, vektör uzayında her bir işaret boyutu için toplamsal olarak gerçekleştirilmiştir. Bu amaçla, CPFSK işaretlerin vektörel uzayda gösterilimi için ayrıca bir program geliştirilmiştir (M 'li CPFSK işaretler vektör uzayında M boyutta ifade edilebilmektedir). Önerilen sistemlerin hata başarımlarının karşılaştırılması amacıyla “Çok Düzeyli Turbo Kodlanmış PSK” sistemlerinin bilgisayar benzetimleri de yapılmıştır. Ayrıca önerilen sistemler hata başarımları performansı yönünden bir başka sistem olan kafes kodlanmış sürekli faz modülasyonlu sistemlerle de karşılaştırılmıştır.

4.1 SÖNÜMLEMELİ KANALLAR

Sürekli faz modülasyonu, gezgin radyo ve uydu kanalları için oldukça uygun bir modülasyon türüdür. Bu kanallarda Gauss gürültüsü olmasına karşın, karşılaşılan daha önemli bir bozucu etki sönümleme etkisidir. Gönderilen işaret, alıcıya farklı yollardan ve değişik zaman gecikmeleri ile ulaştığından sönümleme etkisi oluşur. Bu nedenle, sönümleme etkisi ile özellikle kentsel yerleşim bölgelerindeki yüksek binalar veya ağaç, tepe gibi etkenlerin yoğun olduğu bölgelerde karşılaşılr. Ayrıca birbirini doğrudan görmeyen ve yer üzerinde dalga yayılımı yapamayacak kadar uzakta bulunan iki nokta arasında iletişim kurmak amacıyla iyonosfer katmanlarının yansıtıcı olarak kullanıldığı durumlarda da sönümleme etkisi ile karşılaşılr. İyonosfer katmanlarının yükseklikleri ve elektron yoğunlukları güneş ışınlarıyla yakından ilgilidir ve bu etkenler, gece-gündüz değişimleri ve güneşin parlama oranlarıyla değişken bir özellik gösterir. Bu dinamik

değişimler, iletim yolunun uzunluğunu değiştirmekte ve sönümlemeye neden olmaktadır.

Vericiden iletilen simge dizisi $\underline{z} = (z_0, z_1, \dots)$ ve kanaldan geçerek alıcıya ulaşan simge dizisi $\underline{r} = (r_0, r_1, \dots)$ olmak üzere, sönümlemeli kanaldan alınan n . simge,

$$r_n = \rho_n + \eta_n \quad n=0,1,\dots \quad (4.1)$$

şeklinde yazılabilir $z_n \in \{s_0, s_1, \dots, s_{MP-1}\}$. Burada η_n , ortalaması sıfır ve varyansı $N_0/2$ olan beyaz Gauss gürültüsüdür. $\rho_n \geq 0$ ise, olasılık yoğunluk işlevi Rician dağılımlı yani,

$$p(\rho_n) = 2\rho_n(1+K)\exp(-\rho_n^2(1+K)-K)I_0(2\rho_n\sqrt{K(1+K)}) \quad (4.2)$$

biçiminde olan normalize edilmiş ($E[\rho_n^2]=1$) sönümleme değişkenidir. I_0 , sıfırıncı dereceden birinci tür değiştirilmiş Bessel işlevidir. K , doğrudan alınan işaret gücü ile yerden yansıyan bileşenin gücü toplamının, çevredeki yansımalar sonucu değişik yollardan alınan (saçılmış) işaret gücüne oranını göstermektedir. $K=0$ durumu Rayleigh sönümlemesi olarak adlandırılır ve bu durumda antene gelen tüm işaret çevredeki yansılardan elde edilmektedir (aşırı sönümleme). Bu durumda olasılık yoğunluk işlevi (4.3)'teki gibi olur.

$$p(\rho) = 2\rho\exp(-\rho^2) \quad (4.3)$$

$K=10$ durumu, gezgin uydu kanalları için tipik bir değerdir. $K = \infty$ ise, sönümlemesiz duruma karşı düşer, yani sadece Gauss gürültüsü vardır. Bu tezde alıcının ideal kanal bilgisine sahip olduğu varsayılmaktadır.

Sönümlemeli kanalda tüm işaret/gürültü oranlarında hata başarımını incelemek için kodların aktarım işlevinden yararlanılmaktadır. Eşit olasılıklı ve ζ karesel Öklid uzaklıklı iki simgenin AWGN kanallarda hatalı çözülme olasılığı üst sınırı $e^{-\zeta/4N_0}$ ise, aynı simgelerin sönümlemeli kanallarda hatalı çözülme olasılığı üst sınırı $e^{-\zeta\rho^2/4N_0}$ 'dır [65]. Burada ρ sönümleme değişkenidir.

$$e^{-\zeta \rho^2 / 4 N_0} = \int_0^\infty e^{-\rho^2 / 4 N_0} p(\rho) d\rho \quad (4.4)$$

Kodlanmış M 'li CPFSK, $h=J/P$ kafes yapısının belirlediği geliştirilmiş aktarım işlevinden yararlanır. Bu amaçla, kafes kodun hata diyagramında $[W]^{\zeta \rho^2}$ yazılır ve sönümlemeli kanallar için geliştirilmiş aktarım işlevi $T([W]^{\zeta \rho^2}, I)$ elde edilir. $W = e^{-E/4N_0} = e^{-\gamma}$ biçimindedir. Dolayısıyla, simgeler arasındaki karesel öklid uzaklığını gösteren ζ , E 'ye normalize anlamındadır. Ayrıca,

$$[W]^{\zeta \rho^2} = \frac{1+K}{1+K+\zeta \gamma} e^{-K\zeta \gamma(1+K+\zeta \gamma)} \quad (4.5)$$

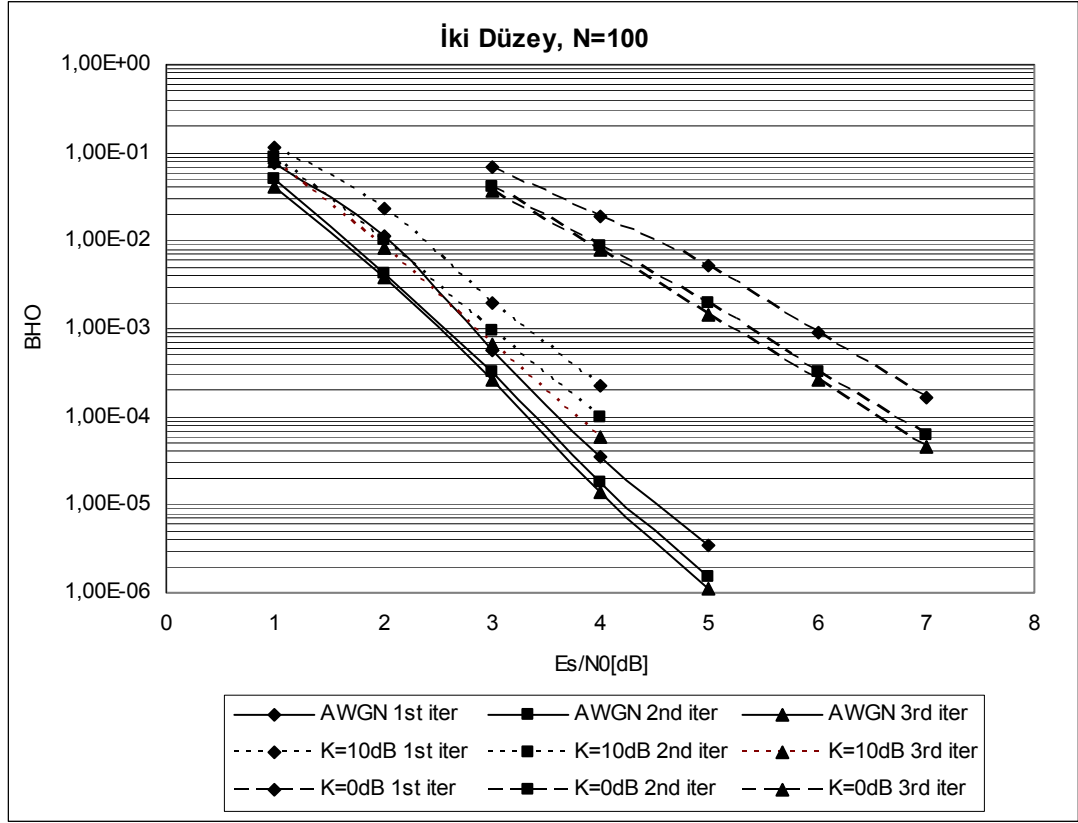
olduğu (4.4)'ten yararlanılarak gösterilebilir. Böylece, sönümlemeli kanallar için bit hata olasılığı üst sınırı,

$$P_b \leq \frac{1}{n} \left. \frac{dT(W, I)}{dI} \right|_{I=1} \quad (4.6)$$

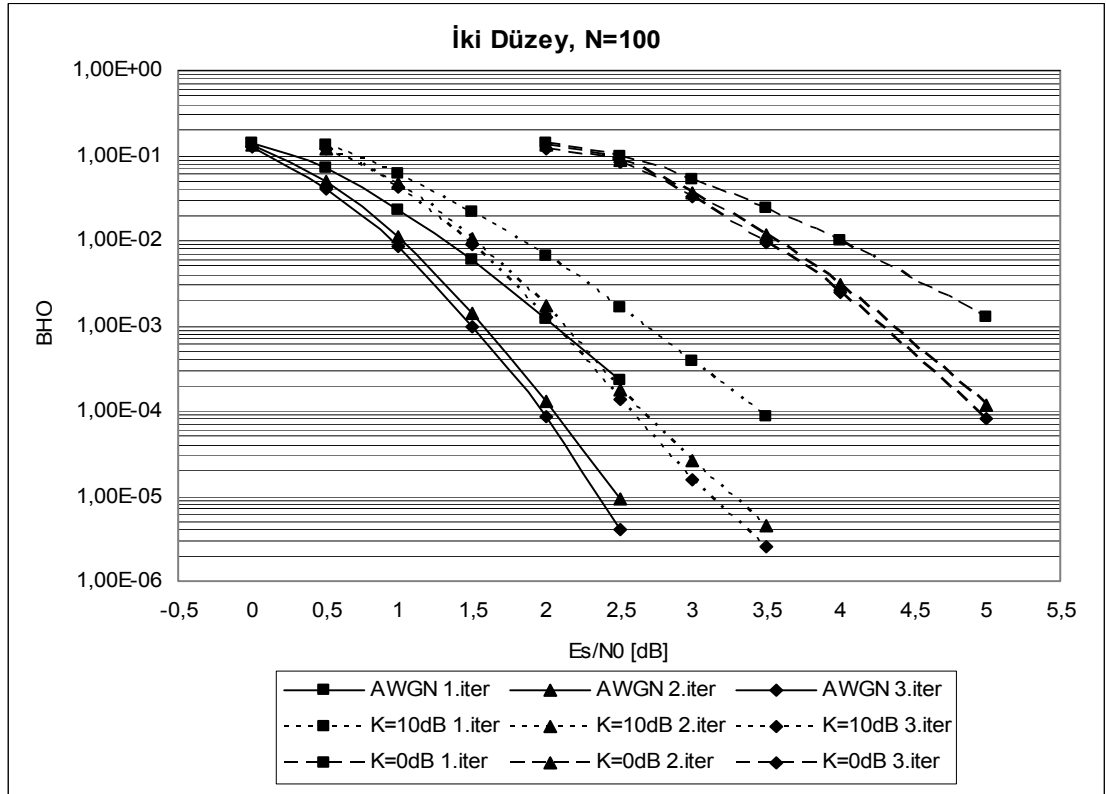
olarak bulunur.

4.2 ÖRNEK SİSTEMLERİN HATA BAŞARIMLARI

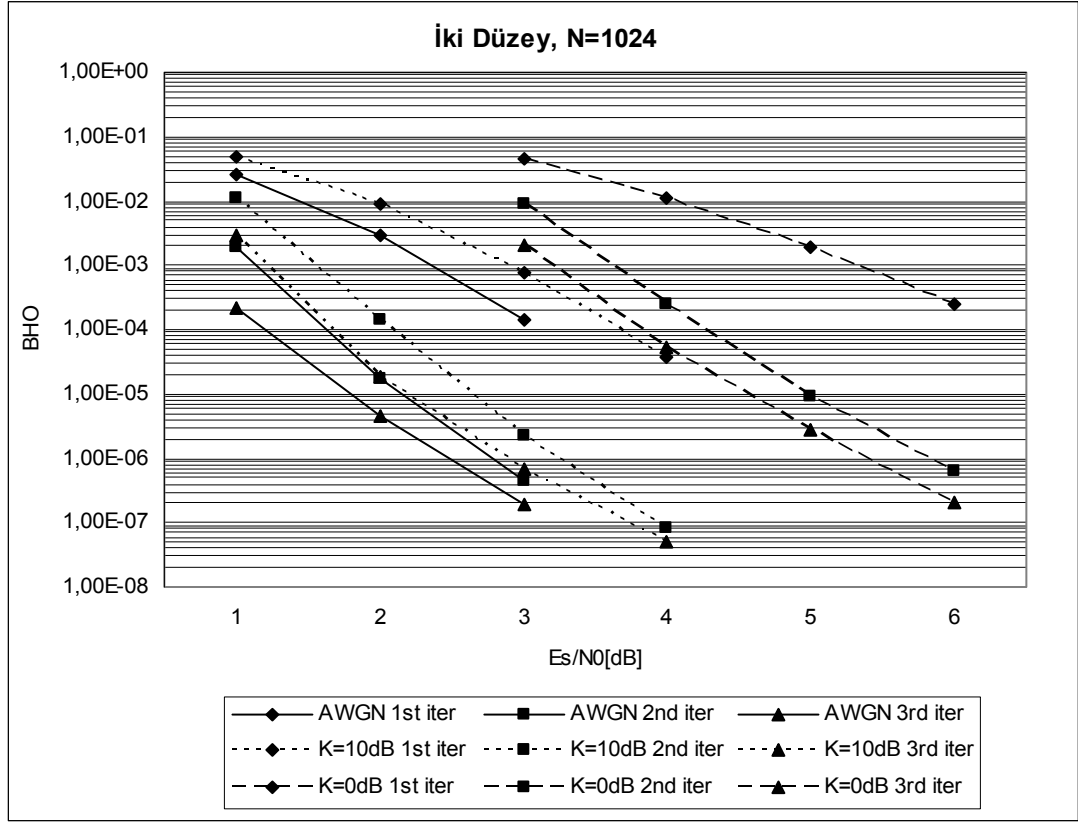
Tezde önerilen sistemlere ilişkin bilgisayar benzetimleri sırasıyla 3.1.3 ve 3.1.4 bölümlerinde detayları verilen iki düzeyli turbo kodlanmış 4'lü CPFSK sistemi ve üç düzeyli turbo kodlanmış 8'li CPFSK sistemi için yapılmıştır. Karşılaştırma amacıyla iki düzeyli turbo kodlanmış 4PSK ve üç düzeyli turbo kodlanmış 8PSK sistemlerinin hata başarımları da elde edilmiştir. Kolaylık olması amacıyla aynı düzey ve aynı çerçeve uzunluğu için bulunan PSK ve CPFSK sistemlerinin hata başarımları eğerlerini gösteren şekiller ard arda konulmuştur. Buna göre ilk olarak çerçeve uzunluğu $N=100$ için Şekil 4.1'de iki düzeyli turbo kodlanmış 4PSK sisteminin Şekil 4.2'de ise iki düzeyli turbo kodlanmış 4'lü CPFSK sisteminin hata başarımları eğerleri görülmektedir. Her iki şekilde de iki düzeyli sistemlerin hata başarımları AWGN, Rician ($K=10$) ve Rayleigh ($K=0$) kanallarda üç iterasyon için çıkarılmıştır. Şekil 4.3 ve Şekil 4.4'te ise sırasıyla iki düzeyli turbo kodlanmış 4PSK ve iki düzeyli turbo kodlanmış 4'lü CPFSK sistemlerinin $N=1024$ için hata başarımları eğerleri verilmiştir.



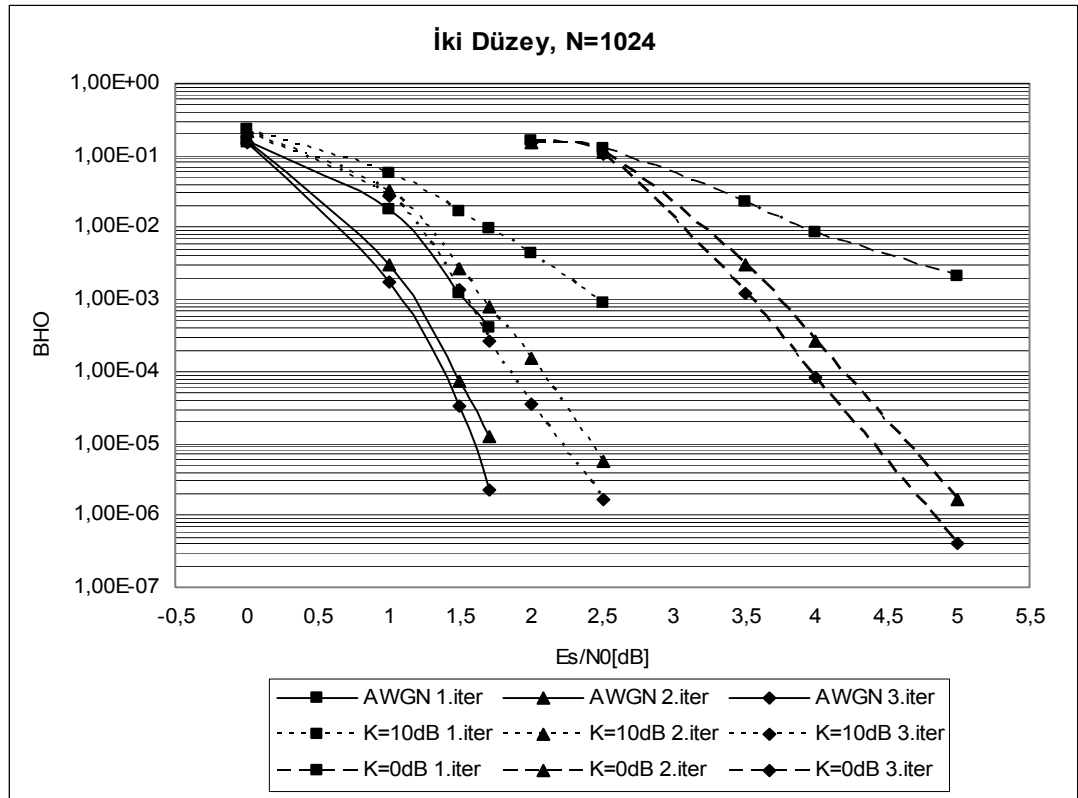
Şekil 4.1 İki düzeyli turbo kodlanmış 4PSK sisteminde $N=100$ için hata başarımlar eğrileri



Şekil 4.2 İki düzeyli turbo kodlanmış 4'lu CPFSK sisteminde $N=100$ için hata başarımlar eğrileri



Şekil 4.3 İki düzeyli turbo kodlanmış 4PSK sisteminde $N=1024$ için hata başarımlar eğrileri



Şekil 4.4 İki düzeyli turbo kodlanmış 4'lü CPFSK sisteminde $N=1024$ için hata başarımlar eğrileri

Hem $N=100$ hem de $N=1024$ için verilen şekillerden de rahatça görülebileceği gibi çok düzeyli turbo kodlanmış sistemlerde PSK modülasyonu yerine CPFSK modülasyonunun kullanılmasıyla önemli oranda bir güç verimi elde edilmektedir. Örnek olarak çerçeve uzunluğu 100 için iki düzeyli 4'lü CPFSK sisteminin iki düzeyli turbo kodlanmış 4PSK sistemine göre 10^{-5} hata olasılığında yaklaşık 2dB'lik kazancı vardır. Rician kanal için 10^{-4} hata olasılığını baz alırsak birinci, ikinci ve üçüncü iterasyonda kazançlar sırasıyla 0.6dB, 2dB ve 2dB olmaktadır. Aynı şekilde Rayleigh kanal için de 10^{-4} hata olasılığı baz alınır üç iterasyon için de kazançlar sırasıyla 1.7dB, 1.3dB ve 1.3dB'dir. Kısaca özetlersek iki düzeyli turbo kodlanmış 4'lü CPFSK sistemlerinin iki düzeyli turbo kodlanmış 4PSK sistemlerine AWGN, Rician ve Rayleigh kanalda sırasıyla yaklaşık 2dB, 0.6-2dB ve 1.3-1.7dB güç kazançları olduğu gösterilmiştir.

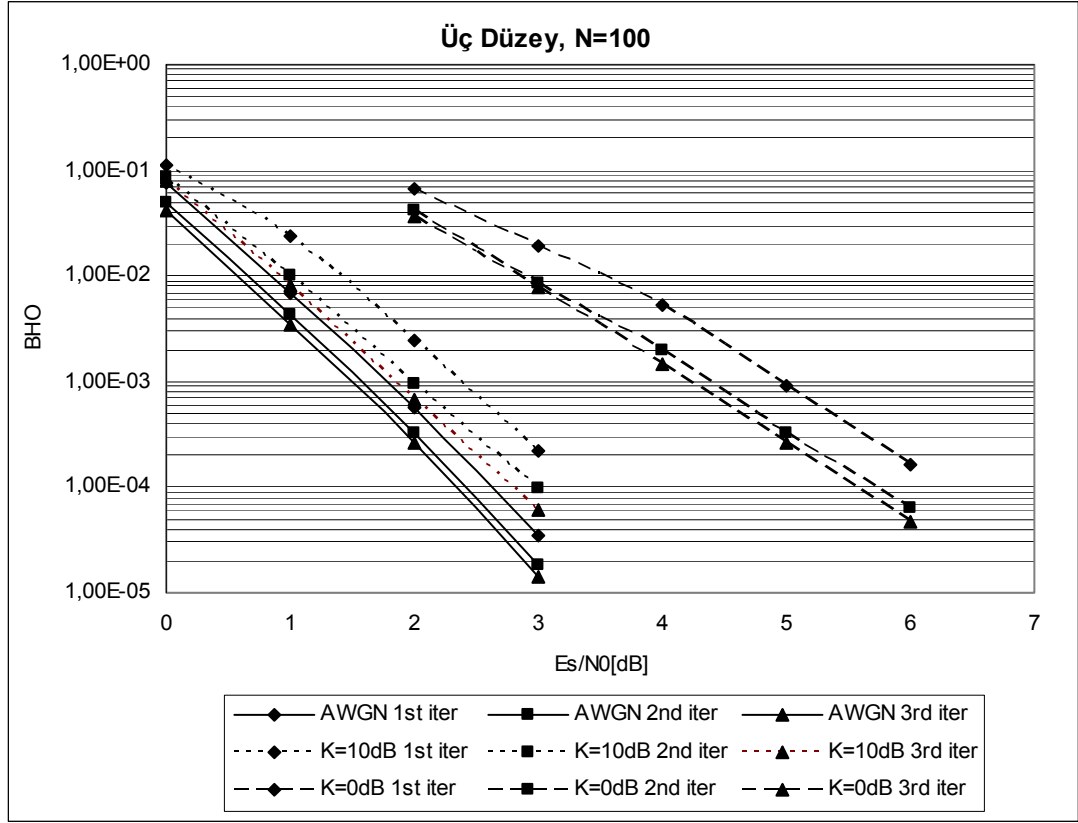
Çerçeve uzunluğu 1024 olduğu durumda ise AWGN kanal için yine 10^{-5} hata olasılığı, Rician ve Rayleigh kanallarda 10^{-4} hata olasılığında iki düzeyli turbo kodlanmış 4'lü CPFSK sisteminin iki düzeyli turbo kodlanmış 4PSK sistemine olan güç kazançları birinci ikinci ve üçüncü iterasyonlar için sırasıyla aşağıdaki şekilde olmaktadır.

AWGN kanalda 0.5dB, 0.7dB ve 0.4dB

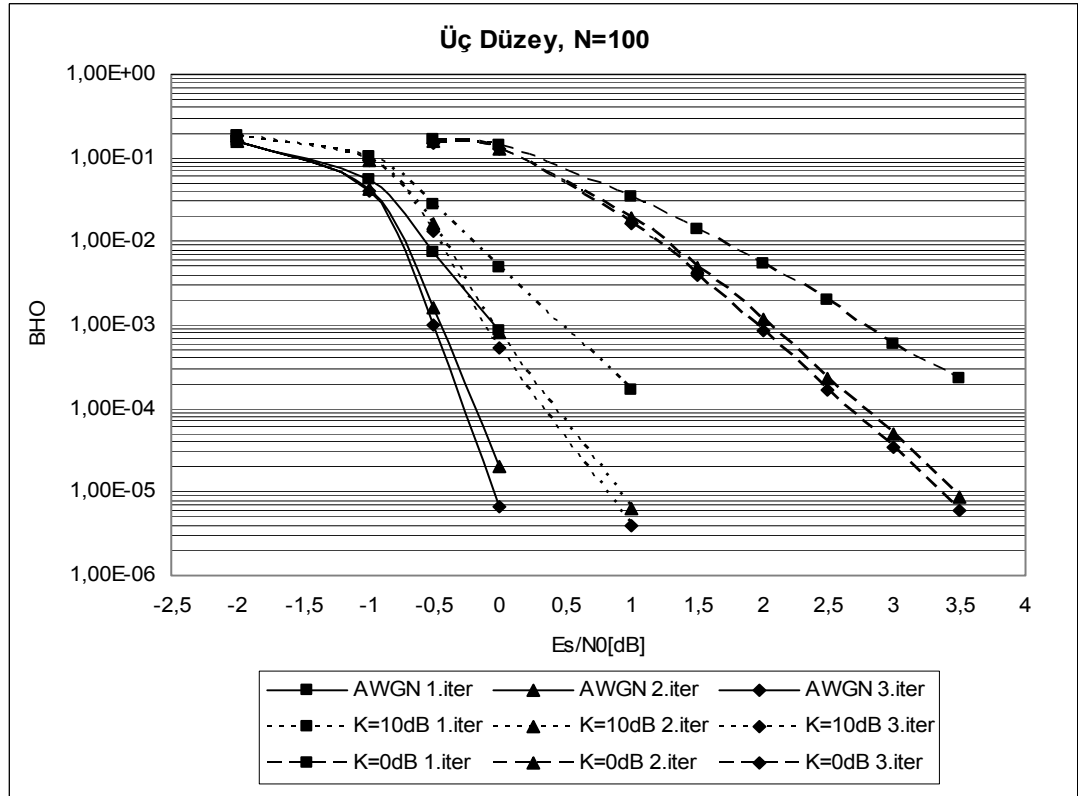
Rician kanalda 0.7dB, 0.2dB ve 0.1dB

Rayleigh kanalda 0.4B, 0.2dB ve 0.1dB

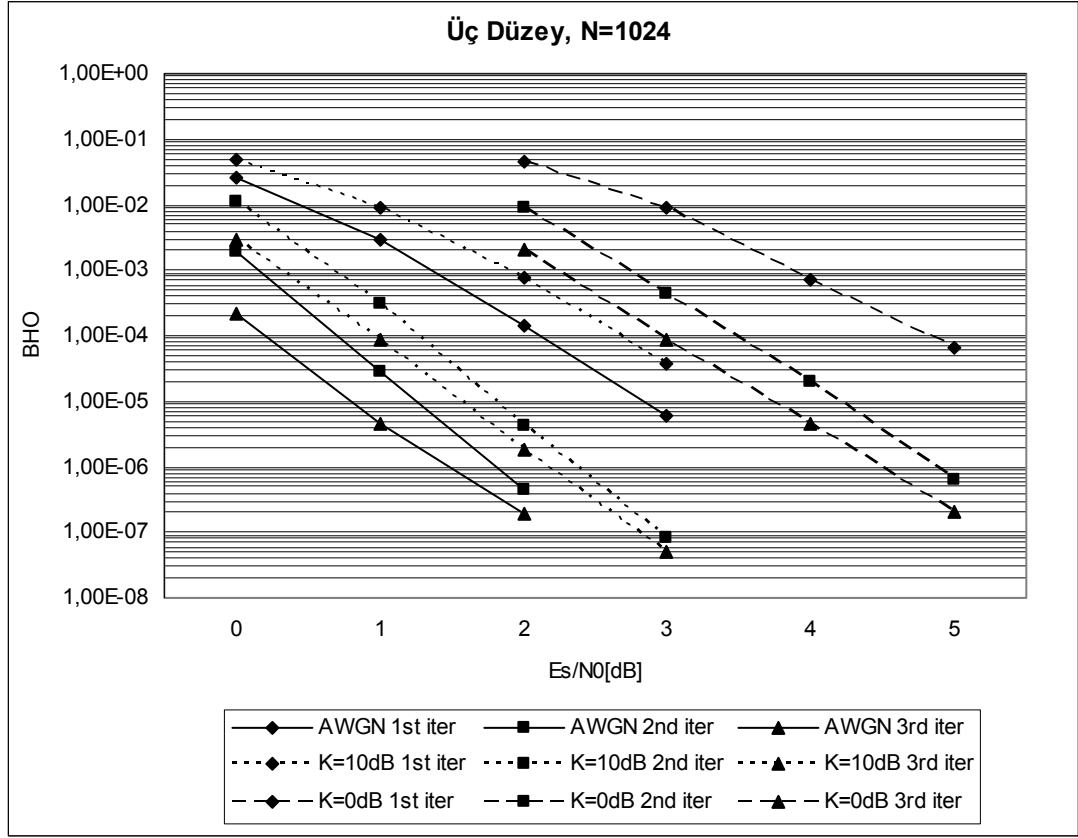
Üç düzeyli turbo kodlanmış 8PSK ve üç düzeyli turbo kodlanmış 8'li CPFSK sistemlerinin $N=100$ durumunda elde edilen hata başarımları eğrileri sırasıyla Şekil 4.5 ve Şekil 4.6'da, $N=1024$ durumunda ise Şekil 4.7 ve Şekil 4.8'de verilmiştir. Şekillerden üç düzeyli CPFSK sisteminin üç düzeyli PSK sistemine göre güç açısından daha verimli olduğu görülmektedir. Çerçeve uzunluğunun 100 olduğu durumu ele alırsak üç düzeyli CPFSK sistemi üç düzeyli PSK sistemine göre AWGN kanalda 10^{-4} hata olasılığı için birinci iterasyonda 2dB, ikinci iterasyonda 2.6dB ve üçüncü iterasyonda 2.6dB güç kazancı sağlar. Bu kazançlar Rician kanalda 10^{-4} hata olasılığı için birinci, ikinci ve üçüncü iterasyonda sırasıyla 2.1dB, 2.5dB ve 2.5dB, Rayleigh kanalda ise yine birinci ikinci ve üçüncü iterasyonlar için sırasıyla 2.4dB, 2.9dB ve 2.9dB olmaktadır.



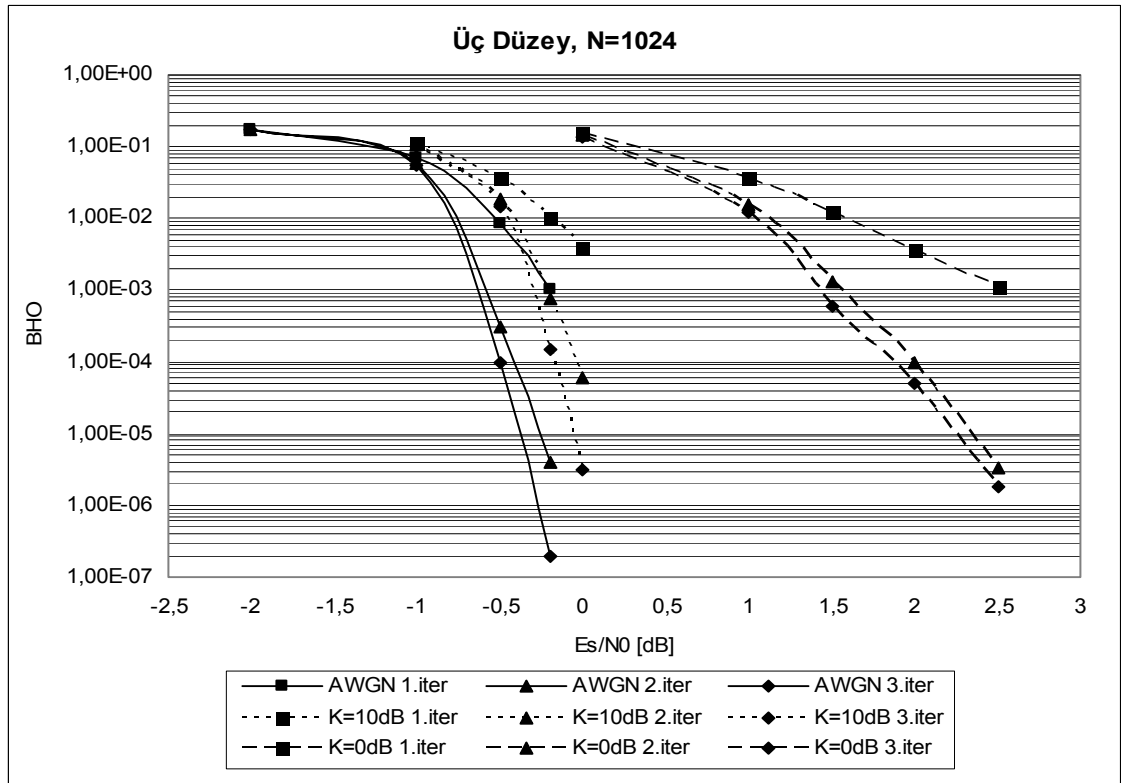
Şekil 4.5 Üç düzeyli turbo kodlanmış 8PSK sisteminde $N=100$ için hata başarımlar eğrileri



Şekil 4.6 Üç düzeyli turbo kodlanmış 8'li CPFSK sisteminde $N=100$ için hata başarımlar eğrileri



Şekil 4.7 Üç düzeyli turbo kodlanmış 8PSK sisteminde $N=1024$ için hata başarımlar eğrileri



Şekil 4.8 Üç düzeyli turbo kodlanmış 8'li CPFSK sisteminde $N=1024$ için hata başarımlar eğrileri

Çerçeve uzunluğu 1024 olduğu durumda ise 10^{-4} hata olasılığında, üç düzeyli turbo kodlanmış 8'li CPFSK sisteminin üç düzeyli turbo kodlanmış 8PSK sistemine olan güç kazançları birinci ikinci ve üçüncü iterasyonlar için sırasıyla aşağıdaki şekilde olmaktadır.

AWGN kanalda 2.1dB, 1.1dB ve 0.8dB

Rician kanalda 2.1dB, 1.2dB ve 1.2dB

Rayleigh kanalda 1.4B, 1.5dB ve 1.1dB

Tezde önerilen çok düzeyli turbo kodlanmış CPFSK sistemlerinin hata başarımları literatürde bulunan bir diğer farklı sistem olan kafes kodlamalı CPFSK sistemleriyle de karşılaştırılabilir. Bu sistemler [66]'da önerilmiş bulunan ve literatürde iyi bilinen sistemlerdir. [66]'da önerilen iki farklı sistem olan kafes kodlanmış 4'lü CPFSK sistemi ve yine kafes kodlanmış 8'li CPFSK sistemi bu tezde önerilen iki düzeyli turbo kodlanmış 4'lü CPFSK sistemi ve üç düzeyli turbo kodlanmış 8'li CPFSK sistemleriyle karşılaştırmak için oldukça ideal sistemlerdir. Kafes kodlanmış 4'lü CPFSK sistemi ve iki düzeyli turbo kodlanmış 4'lü CPFSK sistemlerinin her ikisi de 2/3 kodlama oranına sahip ve 4'lü CPFSK modülasyonu kullanmaktadır. İki düzeyli turbo kodlanmış 4'lü CPFSK sistemi kafes kodlanmış 4'lü CPFSK sistemine göre değişik kanallarda ve değişik iterasyonlar için çerçeve uzunluğu 100 alındığında yaklaşık 3.1-5.1dB aralığında kodlama kazancı sağlamaktadır. Bu kazanç, çerçeve uzunluğu 1024 olduğunda 3.3-5.7dB aralığına çıkmaktadır. Üç düzeyli turbo kodlanmış 8'li CPFSK sistemi ise kafes kodlanmış 8'li CPFSK sistemine göre $N=100$ için değişik kanal ve iterasyonlarda 4.9-7.5dB, $N=1024$ için de 5-8.3dB aralığında kodlama kazancı sağlar. Bu kazançlar Tablo 4.1'de detaylı olarak verilmiştir.

Tabo 4.1'den görülebileceği gibi birinci iterasyonlarla ikinci iterasyonlar arasında yaklaşık 0.5-1dB'lik kazanç vardır. Bu kazanç, ikinci ve üçüncü iterasyonlar arasında sadece 0.1-0.2dB olmaktadır. İterasyon sayısı arttıkça kazanç artım miktarı daha da düşmektedir. Turbo kodlamanın önemli dezavantajlarından bir tanesinin de iterasyona bağımlılık olduğu iyi bilinmektedir. Tezde önerilen yöntemle sistemin iterasyona bağımlılığı azaltılmış ve az iterasyonla yeterli sonuçlar elde edilmiştir. Bilindiği gibi iterasyon sayısının azalması kodçözme süresinin kısılması anlamına gelmektedir ki, bu

da eş zamanlı haberleşme yapabilmek için oldukça önemli bir avantajdır. Tablo 4.1’den görülebilecek bir başka önemli nokta da üç düzeyli sistemler arasındaki kazancın iki düzeyli sistemler arasındakinden daha yüksek olmasıdır. Burada kazancı arttıran temel neden sistem düzey sayısının artması değil, kullanılan modülasyonun 4’lü CPFSK’dan 8’li CPFSK’ya geçmesidir. Örnek olarak alınan sistemler PSK modülasyonu kullandığından modülasyonlu işaretler uzayda iki boyutlu olarak ifade edilmektedir. Oysa 4’lü CPFSK işaretleri 4 boyuta, sekizli CPFSK işaretleri de 8 boyuta sahiptir. Boyut sayısının artması işaretlerin uzayda birbirinden uzaklaşmasını sağlamakta böylece de hata başarımı daha da artmaktadır.

Tablo 4.1 Örnek olarak tasarlanan çok düzeyli turbo kodlanmış CPFSK sistemlerinin kafes kodlanmış CPFSK sistemlerine olan kodlama kazançları

İki düzeyli turbo kodlanmış 4’lü CPFSK sisteminin kafes kodlanmış 4’lü CPFSK sistemine olan kodlama kazançları (dB) (Çerçeve uzunluğu $N=100$)			
İterasyon	AWGN Kanal	Rician Kanal ($K=10\text{dB}$)	Rayleigh Kanal
1	3.1	3.5	4.1
2	3.7	3.9	5
3	3.8	4	5.1
İki düzeyli turbo kodlanmış 4’lü CPFSK sisteminin kafes kodlanmış 4’lü CPFSK sistemine olan kodlama kazançları (dB) (Çerçeve uzunluğu $N=1024$)			
1	3.3	3.6	4.2
2	4.1	4.3	5.5
3	4.2	4.4	5.7
Üç düzeyli turbo kodlanmış 8’li CPFSK sisteminin kafes kodlanmış 8’li CPFSK sistemine olan kodlama kazançları (dB) (Çerçeve uzunluğu $N=100$)			
1	4.9	5.5	6.4
2	5.6	6.3	7.4
3	5.7	6.4	7.5
Üç düzeyli turbo kodlanmış 8’li CPFSK sisteminin kafes kodlanmış 8’li CPFSK sistemine olan kodlama kazançları (dB) (Çerçeve uzunluğu $N=1024$)			
1	5	6.1	7
2	5.8	6.8	8.1
3	5.9	6.9	8.3

Sayısal iletişim sistemlerinin hata başarımlarını incelenirken en önemli ölçüt, sistemin kanal kapasitesi olarak bilinen Shannon sınırına olan uzaklıktır. Önerilen çok düzeyli turbo kodlanmış CPFSK sistemlerinin hata başarımlarının kanal kapasitesine uzaklığının bulunabilmesi için şekil 4.9’da 2,4,8 ve 16’lı CPFSK modülasyonlarına ait sınırlar AWGN kanal için çıkarılmıştır.

AWGN bir kanalın çıkışı,

$$y_n = s_n + w_n \quad (4.7)$$

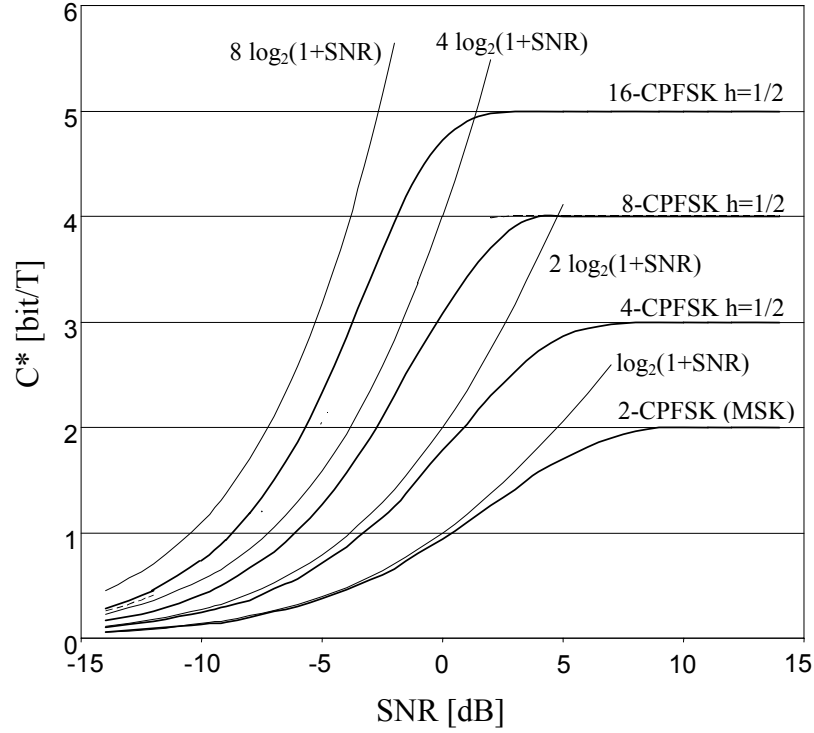
olarak gösterilebilir. Burada s , nT anında kanaldan iletilen işareti, w ise bir boyut için istatistiksel bağımsız, sıfır ortalamalı ve σ^2 varyansa sahip Gauss gürültüsünü göstermektedir. Ortalama İşaret-Gürültü oranı (SNR) aşağıdaki şekilde yazılabilir.

$$SNR = \frac{E\{|s_n|^2\}}{E\{|w_n|^2\}} = \frac{E\{|s_n|^2\}}{M\sigma^2} \quad (4.8)$$

Normalize edilmiş ortalama işaret gücünün 1 olduğunu kabul edersek ($E\{|s_n|^2\}=1$), kanal kapasitesi (4.9)’da gösterildiği gibi hesaplanabilir [2].

$$C^* \approx \log_2(V) - \frac{1}{V} \cdot \sum_{k=0}^{V-1} E \left\{ \log_2 \sum_{i=0}^{V-1} \exp \left[-\frac{|s^k + w - s^i|^2 - |w|^2}{2\sigma^2} \right] \right\} \quad (4.9)$$

Burada V , ilgili CPFSK modülasyonunun işaret kümesinde bulunan toplam işaret sayısını göstermektedir. (4.9)’dan Monte Carlo yöntemiyle kanal kapasitesi (C^*), SNR’ın bir fonksiyonu olarak bulunabilir.



Şekil 4.9 2,4,8 ve 16’lı CPFSK modülasyonları için kanal kapasitesi eğrileri

Şekil 4.9’den görüldüğü gibi CPFSK modülasyonu düşük SNR değerlerinde oldukça yüksek kanal kapasitesine sahiptir. Düşük SNR değerleri için M ’li CPFSK modülasyonuna ait maksimum kanal kapasitesi (4.10) eşitliğinde gösterildiği gibi genelleştirilebilir.

$$C^* \approx \frac{M}{2} \log_2(1 + SNR) \quad (4.10)$$

Tezde tasarlanan iki düzeyli turbo kodlanmış 4’lü CPFSK modülasyonlu sistem $2/3$ bit/T değerine sahiptir. Şekil 4.9’den görüleceği gibi 4’lü CPFSK modülasyonunun $2/3$ bit/T değeri için -5 dB’lik SNR değeri gereklidir. Önerilen sistem için yapılan simülasyon sonuçlarına bakıldığında ise 10^{-5} hata olasılığına $N=100$ için 1.8 dB’de, $N=1024$ olduğunda ise 1.5 dB’de ulaşılmaktadır. Yani iki düzeyli turbo kodlanmış 4’lü CPFSK modülasyonlu sistemin Shannon sınırına uzaklığı $N=100$ için 6.8 dB, $N=1024$ içinse 6.5 dB olmaktadır.

Üç düzeyli turbo kodlanmış 8’li CPFSK sistemi ise 1 bit/T değerine sahiptir. Bu değer için 8’li CPFSK modülasyonunun SNR sınırı yaklaşık -6.2 dB olmaktadır. Simülasyon

sonuçlarına bakıldığında ise, üç düzeyli turbo kodlanmış 8’li CPFSK modülasyonlu sistem 10^{-5} hata olasılığına $N=100$ için 0dB’de $N=1024$ için ise yaklaşık -0.4dB’de ulaşmaktadır. Bu sistemin Shannon sınırına uzaklığı $N=100$ için 6.2dB, $N=1024$ için de 5.8dB olmaktadır.

5000, 10000 bit gibi çok daha büyük çerçeve uzunlukları seçilir ve çok daha fazla kod çözme iterasyonu uygulanırsa, sistemin performansının Shannon sınırına çok daha fazla yaklaştığı görülecektir. Fakat çok büyük çerçeve uzunluğu kullanmak ve çok kod çözme iterasyonu yapabilmek için oldukça yüksek işlem kapasitesine sahip bilgisayarlar kullanılması zorunluluğu açıktır.

TARTIŞMA VE SONUÇ

Bu tezde, sürekli faz modülasyonunun çok düzeyli turbo kodlanmasına ilişkin bir yöntem geliştirilmiş, geliştirilen yöntem için örnek sistemler tasarlanmış ve bu örnek sistemlerin hem toplamsal beyaz gaus gürültülü (AWGN) kanallar hem de sönümlmeli kanallardaki hata başarımları incelenmiştir.

Turbo kodlar çok iyi hata başarımına sahip olmasına rağmen bu kadar iyi bir hata başarımı için çok büyük çerçeve uzunlukları ve kodçözücüde çok sayıda iterasyon gerektirmesi yönünden dezavantajlara sahiptir. Tezde önerilen sistemde turbo kodlar kod çözme karmaşıklığını arttırmayan ve kolay kodlayıcı tasarımına olanak veren çok düzeyli kodlama tekniğine uygulanmış ve çok düzeyli turbo kod tanımı yapılmıştır. Sürekli Faz Modülasyonu sabit zarf, band ve güç verimliliği özelliklerinden dolayı güç ve band sınırlı kanallar için oldukça uygun bir modülasyon yöntemidir. Sürekli faz modülasyonu bilinen faz modülasyonunun sürekli fazlı biçimidir ve faz sürekliliğinden dolayı radyo spektrumunu oldukça verimli kullanır. Sürekli faz modülasyonunun en önemli özelliği olan kanala iletilecek işaretler arasındaki faz sürekliliğinin sağlanması amacıyla sürekli faz kodlayıcısı ile çok düzeyli turbo kodlayıcı birleştirilmiş böylece çok düzeyli turbo kodlar sürekli faz modülasyonuna uygun olarak tasarlanmıştır. Oluşan sistem turbo kodların üstün güç verimi özelliğinin yanında sürekli faz modülasyonu kullandığından band verimli bir sistemdir. Ayrıca çok düzeyli kodlama çok aşamalı kod çözme tekniğine dayalı bir sistem olduğundan istenilen kodlama oranlarında kodlayıcı tasarımı yapılması oldukça basit ve kodçözme karmaşıklığı düşük bir sistemdir. Önerilen yöntemle bilinen klasik ikili turbo kodlardan daha düşük çerçeve uzunlukları ve kod çözmede daha az iterasyon yapılarak aynı hata başarımı elde edilebilmektedir.

Tezde ayrıca iki adet çok düzeyli turbo kodlanmış sürekli faz modülasyonlu sistem tasarımı yapılmış ve bu sistemler detaylarıyla açıklanmıştır. Bu sistemler iki düzeyli turbo kodlanmış 4'lü CPFSK sistemi ve üç düzeyli turbo kodlanmış 8'li CPFSK sistemleridir. İki düzeyli turbo kodlanmış 4'lü CPFSK sistemi iki adet kodlama düzeyinden oluşur ve herbir düzeyde turbo kodlayıcı kullanır. Son düzey olan ikinci düzeydeki turbo kodlayıcıdan sonra yerleştirilen sürekli faz kodlayıcısı sayesinde işaretler 4'lü CPFSK modülasyonuna uygun hale gelir. Alıcı tarafta ise CPFSK demodülatöründen geçen bozulmuş işaretler çok aşamalı turbo kod çözücü

tarafından çözülür. Birinci aşamada birinci düzey turbo kodlayıcı tarafından kodlanan bilgiler çözülür. İkinci aşamada ise, birinci aşamada çözülen bilgi yan bilgi olarak kullanılarak ikinci düzey turbo kodlayıcı tarafından kodlanan bilgiler çözülür. Üç düzeyli turbo kodlanmış 8’li CPFSK sistemi ise üç kodlama düzeyinden ve bu düzeylere paralel olarak üç adet kod çözme aşamasından oluşur. İki düzeyli sisteme benzer şekilde üç düzeyli sistemde de son kodlama düzeyi olan üçüncü düzeyde bulunan turbo kodlayıcıdan sonra, sürekli faz kodlayıcısı yerleştirilmiş ve modülasyonlu işaretler arasında faz sürekliliğinin olması sağlanmıştır.

Tezde tasarlanan sistemlere ait hata başarımlar eğrileri bilgisayar benzetimleriyle AWGN ve sönümlemeli kanallar için elde edilmiştir. Önerilen sistemlerin hata başarımlarının literatürde bulunan benzer sistemlerle karşılaştırılması iki farklı yöntem için yapılmıştır. Bunlardan ilki çok düzeyli turbo kodlanmış PSK sistemleridir. İki düzeyli turbo kodlanmış 4’lü CPFSK sisteminin iki düzeyli turbo kodlanmış 4PSK sistemine göre değişik çerçeve uzunlukları ve değişik kanallarda 0.1-2dB arasında, üç düzeyli turbo kodlanmış 8’li CPFSK sistemi de üç düzeyli turbo kodlanmış 8PSK sistemine göre 0.8-2.9dB aralığında güç kazançları sağladığı bölüm 4’te gösterilmiştir. Tezde tasarlanan sistemlerin hata başarımlar performanslarının karşılaştırılabileceği bir diğer sistem de kafes kodlanmış sürekli faz modülasyonlu sistemlerdir. Tasarlanan sistemleri bu sistemlerle karşılaştırdığımızda ise, iki düzeyli turbo kodlanmış 4’lü CPFSK sisteminin farklı çerçeve uzunlukları ve farklı kanallar için kafes kodlanmış 4’lü CPFSK sistemine göre 3.1-5.7dB kazanç sağladığı gösterilmiştir. Üç düzeyli turbo kodlanmış 8’li CPFSK sisteminin kafes kodlanmış 8’li CPFSK sistemine olan kazancı ise 4.9-8.3dB aralığında değişmektedir.

Bu tezde ele alınan konu hakkında yapılabilecek sonraki çalışmalar şu şekilde özetlenebilir;

- Önerilen yöntemde herbir düzeyde kullanılan turbo kodlayıcılar yerine son yıllarda önemi giderek artan bir başka güç verimli kodlama tekniği olan düşük yoğunluklu parite kontrol kodlarını kullanılarak, sürekli faz modülasyonunun çok düzeyli düşük yoğunluklu parite kontrol kodlarıyla kodlanmasına ilişkin sistemler geliştirilebilir. Veya herbir düzeyde ya turbo kod kullanan yada düşük yoğunluklu parite kontrol kodu kullanan hibrit sistemler üzerinde çalışılabilir.

- Sönümlemeli kanallarda sistemin hata başarımının daha da iyileştirilmesi amacıyla çok düzeyli turbo kodlanmış sürekli faz modülasyonlu sistemler için özel olarak kanal kestirim yöntemleri geliştirilebilir.
- Son yıllarda kablosuz haberleşmenin hızlı bir şekilde yayılmasıyla önemi giderek artan çoklu giriş çoklu çıkış (MIMO) kanalları için yeni sistemler geliştirilebilir.

KAYNAKLAR

1. SHANNON C. E., 1948, A mathematical theory of communication, *Bell Sys. Tech. J.*, 27, 379-423 ve 623-656.
2. UNGERBOECK G., 1982, Channel coding with multi-level/phase signals, *IEEE Trans Inform. Theory*, IT-28, 56-67.
3. UNGERBOECK G., 1987, Trellis coded modulation with redundant signal set part I, *IEEE Commun. Mag.*, 5-11.
4. UNGERBOECK G., 1987, Trellis coded modulation with redundant signal set part II, *IEEE Commun. Mag.*, 15-21.
5. FORNEY, G.D., 1973, The Viterbi Algorithm, *Proc. of IEEE*, 61 (3), 268-278.
6. WILSON, S. G., 1984, Rate 3/4 convolutional coding of 16 PSK: Code design and performance study, *IEEE Trans. Comm.*, Com-32, 1308-1315.
7. WEI, L. F., 1984, Rotationary invariant convolutional channel coding with expanded signal space- Part I: 180°, *IEEE J. Select. Areas Comm.*, SAC-2 (5), 659-671.
8. WEI, L. F., 1984, Rotationary invariant convolutional channel coding with expanded signal space- Part 2: Nonlinear Codes, *IEEE J. Select. Areas Comm.*, SAC-2 (5), 672-685.
9. DIVSALAR, D., SIMON, M. K., 1988, Multiple trellis coded modulation, *IEEE Trans. Comm.*, 36 (4), 410-419.
10. PIETROBON, S. S., 1990, Trellis coded multidimensional phase modulation, *IEEE Trans. Inform Theory.*, IT-36 (1), 63-69.
11. DIVSALAR, D., SIMON, M. K., 1987, Trellis coded modulation for 4800-9600 bits/s transmission over a fading mobile satellite channel, *IEEE J. Select. Areas Comm.*, SAC-5 (2), 162-175.
12. SCHLEGEL, C., COSTELLO, D. J., 1989, Bandwidth efficient coding for fading channels: Code construction and performance analysis, *IEEE J. Select. Areas Comm.*, SAC-7 (9), 1356-1369.
13. Du, J., VUTERIC, B., 1993, Trellis coded 16-QAM for fading channels, *European Transactions on Telecommunications*, 4 (3), 101-107.
14. Du, J., 1995, Construction of new MPSK trellis codes for fading channels, *IEEE Trans. Comm.*, 43 (2/3/4), 776-784.
15. LEONARDO, E. J., 1996, Multidimensional M-PSK trellis codes for fading channels, *IEEE Trans. inform. Theory*, IT-42 (4), 1093-1108.

16. IMAI, H. HIRAKAWA, S., 1977, A new multilevel coding method using error-correcting codes, *IEEE Trans. Inform. Theory*, IT-23 (3), 371-377.
17. YAMAGUCHI, K. IMAI, H., 1987, Highly reliable multilevel channel coding system using binary convolutional codes, *Electronic Letters*, 23 (18), 939-941.
18. YAMAGUCHI, K. IMAI, H., 1989, A study on Imai-Hirakawa trellis-coded modulation schemes", *Lecture Notes in Computer Science*, 357, 443-453.
19. CALDERBANK, A.R., 1989, Multilevel codes and multistage decoding, *IEEE Trans. Comm.*, COM-37 (3), 222-229.
20. KASAMI, T. et al., 1991, On multilevel block modulation codes, *IEEE Trans. Inform. Theory*, IT-37 (4), 965-975.
21. HERZBERG, H. et al., 1993, Concatenated multilevel block coded modulation, *IEEE Trans. Comm.*, COM-41 (1), 41-49.
22. HENKEL, W. KOCH, M., 1993, 90° -Rotationary invariant multilevel convolutionally encoded QAM, *European Transactions on Telecommunications*, 4, 141-147.
23. KOFMAN, Y. et al., 1990, Performance analysis of a multilevel coded modulation system, *EE Pub Technion, Israel*, 751.
24. LEE, W.M., KSCHISCHANG, F.R., 1994, Complexity/performance tradeoffs inn multistage decoders", *European Transactions on Telecommunications*, 5 (6), 665-679.
25. KOFMAN, Y. et al., 1990, A multilevel coded modulation scheme for fading channels, *EE Pub. Technion, Israel*, 778.
26. WU, J. LIN, S., 1993, Multilevel trellis MPSK modulation codes for the Rayleigh fading channel, *IEEE Trans. Comm.*, COM-41 (9), 1311-1318.
27. SESHADRI, N. SUNDBERG, C.E., 1993, Multilevel trellis coded modulations for the Rayleigh fading channel, *IEEE Trans. Comm.*, COM-41 (9), 1300-1310.
28. SUNDBERG, C.E. SESHADRI, N., 1993, Coded modulations for fading channels: An overview, *European Transactions on Telecommunications*, 4 (3), 309-324.
29. ZHANG, L. VUCETIC B., 1995, Multilevel block codes for Rayleigh-fading channels, *IEEE Trans. Comm.*, COM-43 (1), 24-31.
30. RHEE, D.J. et al., 1996, Some block- and trellis coded-modulations for the Rayleigh fading channel, *IEEE Trans. Comm.*, COM-44 (1), 34-42.
31. POTTIE, G.J. TAYLOR, D.P., 1989, Multilevel codes based on partitioning, *IEEE Trans. Inform. Theory*, IT-35 (1), 87-98.

32. BERROU C., GLAVIEUX A., THITIMASJSHIMA P., 1993, Near Shannon-limit error correcting coding and decoding: Turbo codes," *Proc. ICC'93*, 1064-1070.
33. BAHL L. R., COCKE J., JELINEK F., RAVIV J., 1974, Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate, *IEEE Trans. Info. Theory*, IT-20, 284-287.
34. ERFANIAN J. A., PASUPATHY S., 1989, Low-Complexity Parallel-Structure Symbol-by-Symbol Detection for ISI Channels, *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 1-2 Haziran 1989, 350-353.
35. BLACKERT W., WILSON S., 1996, Turbo Trellis Coded Modulation, *Proc. CISS'96*.
36. ROBERTSON P., WORZ T., 1998, Bandwidth-efficient Turbo Trellis-coded modulation using punctured component codes, *IEEE JSAC*, 16 (2), 206-218.
37. SUNDBERG, C. E., 1986, Continuous phase modulation, *IEEE Trans. Comm. Mag*, 24, 25-38.
38. ANDERSON, J. B., 1986, Digital phase modulation, *Plenum*, New York.
39. ANDERSON, J. B., TAYLOR, D. P., 1978, A bandwidth-efficient class of signal space codes, *IEEE Trans. inform. Theory*, IT-24 (6), 703-712.
40. ANDERSON, J. B. SUNDBERG, C. E., 1991, Advances in constant envelope coded modulation, *IEEE Trans. Comm. Mag.*, 36-45.
41. AULIN, T., SUNDBERG, C. E., 1981, Continuous phase modulation- Part 1: Full response signalling, *IEEE Trans. Comm.*, COM-29 (3), 196-209.
42. AULIN, T., SUNDBERG, C. E., 1981, Continuous phase modulation- Part 2: Partial response signalling, *IEEE Trans. Comm.*, COM-29 (3), 210-225.
43. EKANAYAKE, N., 1984, M-ary continuous phase frequency shift keying with modulation index 1/M, *IEE Proceedigs-F*, 131 (2), 173-178.
44. LINDELL, G., 1984, Minimum Euclidean distance for combinations of short range 1/2 convolutional codes and CPFSK modulation, *IEEE Trans. inform. Theory*, IT-30, 509-520.
45. PIZZI, S. V., WILSON, S. G., 1985, Convolutional coding combined with continuous phase modulation, *IEEE Trans. Comm.*, COM-33, 20-29.
46. MORALES-MORENO, F. ve PASUPATHY, S., 1988, Structure, optimization and realization of FFSK trellis codes, *IEEE Trans. Inform. Theory*, 34 (4), 730-751.
47. RIMOLDI, B. E., 1988, A decomposition approach to CPM, *IEEE Trans. inform. Theory*, 34 (2), 260-270.

48. RIMOLDI, B. E., 1989, Design of coded CPFSK modulation systems for bandwidth and energy efficiency, *IEEE Trans. Comm.*, COM-37 (9), 897-905.
49. KARAM, G., 1993, New coded 8-ary CPFSK schemes, *Proc. ICC'93*, Cenevre, isviçre.1059-1063.
50. HUBER, J., LIU, W., 1997, Convolutional codes for CPM using the memory of the modulation process, *GLOBE-COM'97*, Tokyo, 1-5.
51. HUBER, J., LIU, W., 1989, An alternative approach to reduced-complexity CPM-receivers, *IEEE J. Select. Areas Comm.*, SAC-7 (9), 1437-1449.
52. ABRISHAMKAR, F., BIGLIERI, E., 1991, Suboptimum detection of trellis coded CPM for transmission on bandwidth-and power-limited channels, *IEEE Trans. Comm.*, COM-39 (7), 1065-1074.
53. SARI, H. et al., 1996, Trellis coded constant-envelope modulations with linear receivers, *IEEE Trans. Comm.*, COM-44 (10), 1298-1307.
54. ALTUNBAŞ, İ. AYGÖLÜ, Ü., 1995, Multilevel coded M-ary CPFSK with modulation index 1/2 , *Proc. Of 5th Int. Conf. on Advances in Commun. And Control*, COMCON'95, Rethymno, Greece, 504-510.
55. ALTUNBAŞ, İ., 1998, *Sürekli Faz Modülasyonunun Çok Düzeyli Turbo Kodlanması*, Tez (Doktora), İstanbul Teknik Üniversitesi.
56. OSMAN, O., UCAN, O.N., ODABAŞIOĞLU, N., 2005, Performance of Multilevel-Turbo Codes with Group Partitioning Over Satellite Channels, *IEE Proceedings Commun.*, 152 (6), 1055-1059.
57. UCAN, O.N., ODABAŞIOĞLU, N., OSMAN, O., 2004, Çok Düzeyli Turbo Kodlar, *HİTEK 2004*, 9-10 Aralık, İstanbul, 449-453.
58. UCAN, O.N., BUYUKATAK, K., GOSE, E., OSMAN, O., ODABAŞIOĞLU, N., Basımda, Performance of Multilevel-Turbo Codes with Blind/Non-Blind Equalization over WSSUS Multipath Channels, *International Journal of Communication Systems*.
59. BAYAT, O., ODABASIOGLU, N., UCAN, O.N., SHAFI, B., OSMAN, O., 2005, Equalization Of Multilevel Turbo Signals, *Conferance on Information Sciences and Systems CISS'2005*, USA.
60. ODABASIOGLU, N., UCAN, O.N., HEKIM, Y., 2005, Multilevel Turbo Coded-Continuous Phase Frequency Shift Keying (MLTC-CPFSK) Over Satellite Channels in Space Communication, *Proceedings of 2nd International Conference on Recent Advances in Space Technologies*, RAST 2005.
61. OSMAN O., 2004, *Turbo Kodlama Tekniklerinin Başarımı*, Tez (Doktora), İstanbul Üniversitesi.

62. OSMAN O., UCAN O.N., 2005, Blind Equalization of Turbo Trellis-Coded Partial-Response Continuous-Phase Modulation Signaling Over Narrow-Band Rician Fading Channels, *IEEE Trans. On Wireless Comm.*, 4 (2), 397-403.
63. GLOVER I.A., GRANT P.M., 1998, *Digital Communications*, Prentice Hall.
64. RODEN M.S., 1982, *Digital and Data Communication Systems*, Prentice Hall.
65. DIVASLAR, D. ve SIMON, M.K., 1988, "The design of trellis coded MPSK for fading channels: Performance criteria", *IEEE Trans. Comm.*, COM-36, 1004-1012.
66. NARAGHI-POUR, M., 1993, Trellis codes for 4-ary continuous phase frequency shift keying, *IEEE Trans. Comm.*, COM-41 (11), 1582-1587.

ÖZGEÇMİŞ

15.03.1978 tarihinde Konya’da doğan Niyazi Odabaşıoğlu, ilk ve orta okulu Konya ili Akşehir ilçesinde tamamladıktan sonra 1992 yılında Isparta Süleyman Demirel Fen Lisesi’ni kazandı. Buradan 1995 yılında mezun oldu ve aynı yıl İstanbul Üniversitesi Elektrik Elektronik Mühendisliği Bölümünde lisans öğrenimine başladı. 1999 yılında bu bölümden bölüm ikincisi olarak mezun oldu. Yine 1999 yılında İstanbul Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladı. Yüksek lisans eğitimini 2002 yılında bitirerek aynı yıl İstanbul Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği programında doktora eğitimine başladı. 1999 yılının Kasım ayında İstanbul Üniversitesi Elektrik Elektronik Mühendisliği Telekomünikasyon Ana Bilim Dalına Araştırma Görevlisi olarak atandı. Halen aynı bölümde Araştırma Görevlisi olarak çalışmaktadır.