

KIRIKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK ANABİLİM DALI
YÜKSEK LİSANS TEZİ

DÜŞÜK YOĞUNLUKLU EŞLİK KONTROL KODLARI KULLANAN BPSK
SİSTEMLERİNİN AWGN KANAL İÇİN PERFORMANS ANALİZİ

CAHİT YILDIZ

OCAK 2006

Fen Bilimleri Enstitü Müdürünün onayı.

Prof. Dr. M. Yakup ARICA

Müdür

Bu tezin Yüksek Lisans tezi olarak Elektrik-Elektronik Anabilim Dalı standartlarına uygun olduğunu onaylarım.

Prof. Dr. Bilal TOKLU

Anabilim Dalı Başkan V.

Bu tezi okuduğumuzu ve Yüksek Lisans tezi olarak bütün gerekliliklerini yerine getirdiğini onaylarız.

Yrd. Doç. Dr. Eyüp TUNA

Danışman

Jüri Üyeleri

Yrd. Doç. Dr. Eyüp TUNA

Doç. Dr. Şerafettin EREL

Yrd. Doç. Dr. Ediz POLAT

ÖZET

DÜŞÜK YOĞUNLUKLU EŞLİK KONTROL KODLARI KULLANAN BPSK SİSTEMLERİNİN AWGN KANAL İÇİN PERFORMANS ANALİZİ

YILDIZ, Cahit

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Anabilim Dalı, Yüksek Lisans Tezi

Danışman : Yrd. Doç. Dr. Eyüp TUNA

Ocak 2006, 44 sayfa

Bu tezde hata kontrol kodları olarak bilinen LDPC kodları ele alınmıştır ve blok ve konvolüsyon kodları ile karşılaştırılması yapılmıştır. Her üç hata kontrol kodu için BPSK sistemlerinde, AWGN kanal için performans analizleri yapılmış ve sonuçları karşılaştırılmıştır.

LDPC kodları ile blok ve konvolüsyon kodlarının BPSK sistemleri üzerindeki bit hata olasılığı (BER) karşılaştırılması Matlab programıyla simülasyon yapılarak verilmiştir. Simülasyon sonuçlarından LDPC kodlarının diğer hata kontrol kodlarına göre daha üstün bir performans sağladığı gösterilmiştir.

Anahtar Kelimeler: LDPC kodları, Blok kodlar, Konvolüsyon kodları

ABSTRACT

THE PERFORMANCE ANALYSIS OF BPSK SCHEMES USING LOW DENSITY PARITY-CHECK CODE OVER AWGN CHANNEL

YILDIZ, Cahit

Kırıkkale University

Graduate School of Natural and Applied Sciences

Department of Electrical and Electronics Engineering, M.S. Thesis

Adviser: Asst. Prof. Dr. Eyüp TUNA

January 2006, 44 pages

This thesis presents a performance comparison between LDPC and block and convolution error control codes. First, the structure of LDPC codes is explained. Then LDPC codes are compared with block codes and convolution codes. Specifically, performance results for AWGN channel in BPSK systems of these three codes have been compared. Bit error rates (BER) on BPSK systems of LDPC codes, block codes, and convolution codes are simulated using Matlab. Simulation results show that LDPC codes provide better performance than the other two error control codes.

Key Words: LDPC codes, Block codes, Convolution codes

Aileme

TEŐEKKÜR

Tezimin hazırlanması esnasında her türlü yardımını esirgemeyen ve bilimsel konularda daima yardımını gördüğüm tez yöneticisi hocam, Sayın Yrd. Doç. Dr. Eyüp TUNA'ya ve hayatım boyunca bana birçok konuda olduğu gibi, tezimi hazırlamam esnasında da yardımlarını ve desteklerini esirgemeyen aileme teşekkür ederim.

SİMGELER DİZİNİ

H	Eşlik kontrol matrisi
G	Üreteç matrisi
w_c	H eşlik kontrol matrisinin her bir sütununun içerdiği 1 sayısı
w_r	H eşlik kontrol matrisinin her bir satırının içerdiği 1 sayısı
$\lambda(x)$	Değişken düğüm derece dağılım polinomu
$p(x)$	Kontrol düğümü derece dağılım polinomu
P_i^{int}	İçsel olasılık
P_i^{ext}	Dışsal olasılık
LLR	Logaritmik olasılık oranı

KISALTMALAR

ARQ	Otomatik tekrar isteği
AWGN	Toplanabilir beyaz Gauss gürültülü kanal
BPSK	İkili faz kaydırmalı anahtarlama
FEC	İleri hata düzeltme
LDPC	Düşük yoğunluklu eşlik kontrol kodu
SPA	Toplam-çarpım algoritması

ŞEKİLLER DİZİNİ

ŞEKİL

2.1. Konvolüsyon Kodlayıcısı	12
2.2. Konvolüsyon Kodlayıcı İçin Durum Diyagramı	15
2.3. Konvolüsyon Kodlayıcı İçin Kafes Diyagramı	16
2.4. Kafes Boyunca Mümkün Yollarının Gösterilmesi	18
2.5. Eşlik Kontrol Matrisi Yapısı	20
2.6. Eşlik Kontrol Matrisine Bağlı LDPC Kodun Tanner Grafiği	22
2.7. Gallager'in Eşlik Kontrol Matrisi Yapısı	24
2.8. Örnek Bir Bit Çevirim İşlemi	28
2.9. Periyodun 4 Olduğu Bir Bit Çevrim Kod Çözme İşlemi	29
2.10. Örnek Bir Toplam-Çarpım Kod Çözme Algoritması	34
3.1. (7,4) Blok Kodun AWGN Kanal Üzerindeki Karşılaştırılmalı Performansı	36
3.2. 1/2 Kod Oranlı Konvolüsyon Kodlayıcısı	37
3.3. 1/2 Oranlı Konvolüsyon Kodun AWGN Kanal Üzerindeki Karşılaştırılmalı Performansı	38
3.4. H Eşlik Kontrol Matrisi	39
3.5. 1/2 Oranlı LDPC Kodun AWGN Kanal Üzerindeki Karşılaştırılmalı Performansı	40
4.1. Karşılaştırmalı Simülasyon Sonuçları	41

ÇİZELGELER DİZİNİ

TABLO

2.1. Konvolüsyon Kodlayıcının 1/2 Kod Oranı İçin En Uygun Biçimi	14
--	----

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
SİMGELER DİZİNİ	iv
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	vi
İÇİNDEKİLER	vii
1. GİRİŞ	1
1.1. Kaynak Özetleri	3
1.1.1. Çalışmanın Amacı	5
2. MATERYAL VE YÖNTEM	6
2.1. Lineer Blok Kodlar	6
2.1.1. Lineer Blok Kodların Kodlanması	6
2.1.2. Kod Kelimesinin Sendromu ve Hata Sezme	8
2.1.3. Minimum Uzaklık ve Blok Kodların Hata Sezme Ve Hata Düzeltilme Kapasiteleri	9
2.1.4. Kod Çözme	10
2.2. Konvolüsyon Kodları	12
2.2.1. Kod Üretimi	12
2.2.2. Durum ve Kafes Diyagramı	14
2.2.3. Konvolüsyon Kodlarda Kod Çözme ve Viterbi Algoritması	17

2.3. Düşük Yoğunluklu Eşlik Kontrol (LDPC) Kodları	19
2.3.1. LDPC Kodlarının Gösterimleri	20
2.3.1.1. LDPC Kodlarının Matrisel Gösterimi	20
2.3.1.2. LDPC Kodlarının Grafiksel Gösterimi	21
2.3.2. LDPC Kod Tasarım Yaklaşımları	23
2.3.2.1. Gallager Kodları	23
2.3.2.2. Mackay Kodları	24
2.3.2.3. Düzensiz LDPC Kodları	25
2.3.3. LDPC Kod Çözümü	26
2.3.3.1. Yinelemeli Kod Çözümü	26
2.3.3.2. Toplam-Çarpım Kod Çözümü	30
3. ARAŞTIRMA BULGULARI VE TARTIŞMA	35
3.1. Blok Kodu Kullanan BPSK Sisteminin AWGN Kanal Üzerindeki Performans Analizi	35
3.2. Konvolüsyon Kodu Kullanan BPSK Sisteminin AWGN Kanal Üzerindeki Performans Analizi	37
3.3. LDPC Kodu Kullanan BPSK Sisteminin AWGN Kanal Üzerindeki Performans Analizi	39
4. SONUÇ	41
KAYNAKLAR	42

1. GİRİŞ

C.E Shannon'a dek haberleşme teorikleri gürültülü bir kanal üzerinden enformasyon iletimi işleminde yapılan hata olasılığını küçültmek için yegane yolun işaret/gürültü oranının büyütülmesi ve/veya iletim hızının düşürülmesi gerektiğini düşünmekteydiler. Claude Shannon'un 1948'de yayınlanan iletişimin matematiksel teorisi adlı tezinde gürültülü kodlama teorisine bir başlangıç yapılmıştır⁽¹⁾. Shannon'un bu tezinde, her iletişim kanalının kanal kapasitesi adı verilen bir sayıyla ilişkili olduğu saptanmıştır. Gönderilen bilgi kanal tarafından bozulsa bile, bilginin kanal kapasitesinden daha düşük oranda gönderilmesi koşulu altında güvenli bir iletişimin sağlanabileceği kanıtlanmıştır. Güvenli bir iletişim sağlamak için literatürde değişik hata sezme ve/veya hata düzeltme şeklinde hata kontrol kodları kullanılmıştır.

Veri haberleşmesinde ileri hata düzeltme (FEC) ve otomatik tekrar isteği (ARQ) olmak üzere temelde iki çeşit hata kontrol protokolü vardır.

FEC hata kontrol protokolünde iletim hatalarının giderilmesi, alınan kelimedeki hataların düzeltilmesi şeklinde yapılır. Yani hatalar düzeltilir ve elde edilen doğru bilgi kullanıcıya verilir. Alıcı hataların varlığını sezemez veya gerçek yerlerini belirleyemez, belirlediği halde hata miktarı kullanılan hata düzeltici kodun hata düzeltme kapasitesinin üzerinde olur ise kullanıcıya hatalı bilgi verilmesi kaçınılmaz olur⁽²⁾. Bu sebepten dolayı FEC hata kontrol protokolünü kullanan sistemlerin güvenilirliklerini arttırmak için hata düzeltme kapasitesi yüksek olan kodların kullanılması gerekir.

Hatayı en kolay sezebilen hata sezme sistemi, eşlik-kontrol kodu kullanan sistemlerdir. İkili mesaj bitine ilave bit eklemek şeklinde elde edilirler. k adet enformasyon bitine $n-k$ adet eşlik bitleri ekleyerek oluşturulan kodlara blok kodlar denir ve (n,k) şeklinde gösterilirler. Kodlanmasının kolaylığından dolayı özellikle hata sezme amaçlı olarak yaygın bir şekilde kullanılırlar.

Düşük yoğunluklu eşlik kontrol kodları (LDPC), eşlik-kontrol matrisi H 'nin düşük yoğunluklu olarak 1'lere sahip olduğu bir tür blok kodudur. "Düşük-yoğunluk" ile kastedilen kodun özelliğini belirleyen eşlik-kontrol matrisinin blok kodunun matrisine göre daha az sayıda 1, daha fazla sayıda 0 içermesidir. Düzenli LDPC kodu, H eşlik-kontrol matrisinin, her bir sütununun aynı w_c sayıda 1 içermesi ve her bir satırının da $w_r = w_c(n/m)$ sayıda 1 içermesidir. Burada m , $n-k$ tane eşlik-kontrol bitini gösterir ve $w_c \ll m$ 'dir. Kod oranı $R = k/n$ olduğundan, bu parametreler ile kod oranı arasında $R = 1 - w_c/w_r$ şeklinde bir ilişki mevcuttur. H düşük yoğunluklu olduğu halde her bir satır ve sütundaki 1'lerin sayısı sabit değilse, o zaman bu koda düzensiz LDPC kodu denir. Gallager'in 1962'deki ilk tezinde^(3,4), LDPC kodlarının kullanılmasıyla Shannon'un limit değerine oldukça yakın bir performans elde edildiği görülmüştür.

Bu tezde LDPC kodları kullanan BPSK sistemlerinin, AWGN kanal için performansları elde edilmiştir. Sonuçlar diğer hata sezme ve düzeltme kodları olan, blok kodlar ve konvolüsyon kodlarının kullanılmasıyla elde edilmiş olan sonuçlar ile karşılaştırılmıştır.

1.1.Kaynak Özetleri

LDPC kodları üzerine yapılan ilk çalışma 1962 yılında Gallager'in yaptığı doktora tezidir^(3,4). Gallager bu çalışmasında düzenli LDPC kodlarının H eşlik kontrol matrisinin yapısının $w_c \geq 3$ ve $w_r > w_c$ boyutlarında mükemmel bir özelliğe sahip olduğunu göstermiştir. LDPC kodun kod çözümünde yinelemeli kod çözme algoritmasını kullanmıştır ve kod kelimesi uzunluğu büyük olduğunda bu yöntemle kod çözmenin performans kapasitesini sınırladığı saptamıştır.

MacKay⁽⁵⁾ çalışmasında LDPC kodlarının eşlik kontrol matrisini yarı-rasgele özelliğe sahip olacak şekilde üretmiştir. H eşlik kontrol matrisinden Gauss eleme yoluyla G üreteç matrisini üretmiştir. G'nin alt matrisi olan P matrisinin yeterince seyrek olmayışı kodlamadaki karmaşıklığı arttırmıştır.

Tanner⁽⁶⁾ LDPC kodlarının iki parçalı grafik olarak adlandırdığı grafikte etkili olarak nasıl gösterilebileceğini ortaya koymuştur. LDPC kodunun Tanner grafiği, konvolüsyon kodunun kafes diyagramına benzer bir şekilde, kodun eksiksiz olarak gösterilmesini sağlar ve kod çözme algoritmasının tanımlanmasına yardım eder. Tanner grafiğindeki periyodun LDPC kodları üzerinde önemli bir etkisi vardır: Küçük periyotların LDPC kodlarına uygulanan yinelemeli kod çözme algoritmasının performansını azaltıcı yönde bir etkisi olur. Tanner periyottan bağımsız olan LDPC kodlarının toplam-çarpım algoritmasıyla (SPA) iyi performans verdiğini göstermiştir. Literatürde LDPC kodlarına kod çözme algoritması olarak toplam-çarpım algoritmasının kullanıldığı Chung⁽⁷⁾, Richardson ve Urbanke'nin⁽⁸⁾ çalışmaları da mevcuttur.

MacKay ve Neal^(9,5) çalışmalarında LDPC kodlarının, pratik uygulamada, grafiksel gösteriminin kısa periyottan bağımsız olduğunu göstermişlerdir.

Etzion⁽¹⁰⁾, $k/n \geq 0,5$ oranlı periyotsuz Tanner grafiđi ile gösterilebilen lineer kodların minimum uzaklıđının yaklaşık olarak 2 olduđunu bulmuştur. Bu sonuđ ile LDPC kodlarının performansının eřik deđerini sađlamıř, dolayısıyla LDPC kod seçiminde önemli bir kriter ortaya koymuřtur.

Bugüne kadar yapılan alıřmalara bakıldıđında yinelemeli kod özme algoritmasıyla yapılan analizlerin az olduđu görölür. Bu alandaki alıřmaların büyük kısmını C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, ve R. L. Urbanke⁽¹¹⁾ yapmıřtır.

Luby ve arkadaşları⁽¹²⁾ düzensiz LDPC kodlarının düđüm dađılımı ile genişletilebileceđini ve düzenli kodların düzensiz kodlar kadar iyi performans göstermediđini kanıtlamıřlardır.

Richardson⁽¹³⁾ ve Luby⁽¹⁴⁾ düzensiz LDPC kodlarının derece dađılımı polinomlarını $\lambda(x)$ ve $p(x)$ olarak tanımlamıř ve bu tür kodların bu polinomlarla kanalların özelliklerine göre nasıl uygunlařtırılacađını göstermiřtir.

Chung⁽¹⁵⁾ ve arkadaşları yaptıđı alıřmada düzensiz LDPC kodlarının düđüm derece dađılımı ile toplam-arpım algoritması yardımıyla özölüđünde Shannon'un limit deđerine en yakın ve en iyi performans verdiđini göstermiřlerdir.

R. Lucas, M. P. C. Fossorier, Y. Kou, ve S. Lin⁽¹⁶⁾ yaptıkları alıřmada, önerdikleri sonlu geometrik yapıdaki eřlik kontrol matrisine sahip periyodik LDPC kodlarının yinelemeli kod özme ile performanslarının arttıđı görölümüřtür.

Forney⁽¹⁹⁾ alıřmasında konvolüsyon kodlayıcının kafes diyagramını tarafından nasıl gösterileceđini geliřtirmiřtir.

Stephan ten Brink, Gerhard Kramer ve Alexei Ashikhmin⁽²³⁾, düzensiz LDPC kodu ile kodlanmış kod kelimesinin nasıl modüle edileceğini araştırmışlardır. Bunun için öncelikle, uygun bir modülatör ve dedektör seçilmiş ve LDPC kodu ile birleştirilmiştir. Yinelemeli kod çözme algoritmasının bu sisteme nasıl uyarlanacağını araştırmışlardır.

Pascal O. Vontobel ve Ralf Koetter⁽²⁴⁾, LDPC kodlarının kod çözme algoritması olan yinelemeli kod çözme algoritmasının sınırlı uzunluktaki kod kelimesi için çeşitli analizler yapmışlardır.

Jose M. F. Moura, Jin Lu ve Haotian Zhang⁽²⁵⁾, geniş kalınlığa sahip düzenli LDPC kodlarının Tanner grafiği ile nasıl tasarlanacağını araştırmışlardır. Geniş kalınlık, LDPC kodları için bit hata olasılığını (BER) arttırıcı bir özelliktir.

1.1.1. Çalışmanın Amacı

Bu tezde, hata kontrol kodları olarak bilinen blok kodları, konvolüsyon kodları ve LDPC kodlarının yapısı ayrıntılı olarak anlatılacaktır. Bu hata kontrol kodlarının kullanıldığı BPSK sistemlerinin AWGN kanal üzerindeki performansları ayrı ayrı ve karşılaştırılmalı olarak verilecektir ve simülasyonları yapılacaktır. Simülasyon sonuçlarından LDPC kodlarının diğer hata kontrol kodlarına göre daha üstün bir performans sağladığı gösterilecektir.

2. MATERYAL VE YÖNTEM

2.1. Lineer Blok Kodlar

k adet enformasyon bitine n-k adet eşlik bitleri ekleyerek oluşturulan kodlara blok kodlar denir ve (n,k) şeklinde gösterilirler. Kodlanmasının kolaylığından dolayı özellikle hata sezme amaçlı olarak yaygın bir şekilde kullanılırlar. Kod kelimesi içindeki konumuna bağlı olarak blok kodlar sistematik ve sistematik olmayan kodlar olarak iki grupta incelenir. Fazlalık bitleri olarak da anılan eşlik bitleri kod kelimesinde, mesaj bitlerinin hemen öncesine yerleştirilmişse bu kodlara sistematik kodlar denir. Bu blok kod aynı zamanda lineerdir. Kod kelimesindeki fazlalık bitleri mesaj bitleriyle harmanlanmışsa bu kodlara da sistematik olmayan kodlar denir⁽¹⁷⁾.

2.1.1. Lineer Blok Kodların Kodlanması

Bir (n,k) lineer blok kodunda, her bir v kod kelimesi $(u_0 u_1 \wedge u_{k-1})$ şeklindeki k adet mesaj biti ile $(g_0 g_1 \wedge g_{k-1})$ şeklindeki üreteç matrisinin lineer bir çarpımından elde edilir.

$$v = u_0 \cdot g_0 + u_1 \cdot g_1 + \dots + u_{k-1} \cdot g_{k-1} \quad (2.1)$$

Burada, $u_i = 0$ yada 1'dir. $g_i = (g_{i0}, g_{i1}, g_{i2}, \dots, g_{i,n-1})$ olmak üzere üreteç matrisinin elemanları $k \times n$ boyutunda bir matrise yerleştirilirse;

$$G = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{pmatrix} = \begin{pmatrix} g_{00} & g_{01} & g_{02} & \dots & g_{0,n-1} \\ g_{10} & g_{11} & g_{12} & \dots & g_{1,n-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \dots & g_{k-1,n-1} \end{pmatrix} \quad (2.2)$$

ifadesi elde edilir. Kodlanması gereken $u = (u_0, u_1, \dots, u_{k-1})$ şeklindeki mesaj bitleri ile elde edilecek kod kelimesi şu şekilde olacaktır;

$$v = u \cdot G \quad (2.3)$$

Lineer sistematik bir (n,k) blok kodu için, G üreteç matrisinin yapısı

$$G = \begin{pmatrix} p_{00} & p_{01} & \dots & p_{0,n-k-1} \\ p_{10} & p_{11} & \dots & p_{1,n-k-1} \\ \vdots & \vdots & \dots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \dots & p_{k-1,n-k-1} \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad (2.4)$$

p matrisi
kxk birim matrisi

şeklinindedir. Burada $p_{ij} = 1$ yada 0 'dır. Görüldüğü gibi G üreteç matrisi iki alt matrise sahiptir. Bunlardan biri $k \times k$ boyutunda I_k birim matrisi diğeri de $k \times (n-k)$ boyutundaki p matrisidir. Blok kodundaki maharet tamamen bu p matrisinin seçimine bağlıdır. Lineer sistematik blok kod için kod kelimesinin yapısına yeniden dönecek olursak, ilk $n-k$ sayıdaki bit fazlalık bitleri, sonrasındaki k bit de k adet enformasyon bitine karşılık düştüğü için, n bitten oluşan bir v kod kelimesi şu şekilde ifade edilir;

$$v_j = \begin{cases} u_0 \cdot p_{0j} + u_1 \cdot p_{1j} + \dots + u_{k-1} \cdot p_{k-1,j} & , 0 \leq j < n - k \\ u_j & , n - k \leq j < n \end{cases} \quad \text{için} \quad (2.5)$$

2.1.2. Kod Kelimesinin Sendromu ve Hata Sezme

Alıcıya gelen bir kod kelimesinin hatalı olup olmadığını anlamak için o kod kelimesinin sendromu hesaplanır. Eğer sendrom sıfır olursa alınan kod kelimesi hatasız demektir. Sendromun sıfırdan farklı çıkması ise o kod kelimesinin hatalı olduğunu gösterir. Bir kod kelimesinin S sendrom vektörü şu eşitlikle hesaplanır;

$$S = r \cdot H^T \quad (2.6)$$

Burada r alınan kod kelimesi, H'de eşlik matrisidir. Eşlik matrisi lineer blok kodlar için tanımlı matrislerden biridir ve;

$$H = (I_{n-k} \mid P^T) \quad (2.7)$$

olarak ifade edilir. Her bir G üreteç matrisi için bir eşlik matrisi tanımlanır. Eşlik matrisi, bir v kod kelimesinin G üreteç matrisinden elde edilmiş bir kod kelimesi olup olmadığını belirlemede kullanılır. v kod kelimesi, eğer $v \cdot H^T = 0$ eşitliğini sağlıyorsa G üreteç matrisinden elde edilmiş bir kod kelimesidir. Zaten sendrom hesabında da eşlik matrisinin bu özelliği kullanılır. Eğer alıcıya gelen kod kelimesinde bir hata oluşmuşsa bu kod kelimesi bu G üreteç matrisine ait bir kod kelimesi olmayacaktır. Dolayısıyla sendromu sıfır olarak hesaplanmayacaktır. Ancak kanaldaki hata oluşumu beklenenin üzerinde gerçekleşirse iletilen kod kelimesi başka bir kod kelimesine dönüşür. Bu durumda sendrom sıfır olarak hesaplanır, gerçekte hatalı iletilmiş olan kod kelimesi doğru bir kod kelimesi olarak algılanır dolayısıyla yanlış mesaj kullanıcıya verilir. İşte kodun sezme kapasitesi burada önemini hissettirmektedir. Güçlü olarak nitelendirilebilecek bir kod ile böylesi oluşumların sıklığının azaltılması durumunda güvenli bir iletişim gerçekleştirilebilir .

2.1.3. Minimum Uzaklık ve Blok Kodların Hata Sezme ve Hata Düzeltme

Kapasiteleri

Minimum uzaklık herhangi bir kodun hata sezme ve hata düzeltme kapasitelerini belirlemede kullanılan bir parametredir. Minimum uzaklık bir koda ait kod kelimelerinin birbirlerine benzerlikleriyle ilgilidir. Örneğin (1110010) ile (1001011) arasındaki farklı bit konumlarının sayısı 4'tür. (000000) ile (1111111) arasındaki farklı bit konumlarının sayısı ise 7'dir. Tüm kod kelimeleri birbirleriyle karşılaştırıldıklarında elde edilen farklı bit sayıları konumlarının minimum olanı minimum uzaklık olarak isimlendirilirler. Bir blok kodun minimum uzaklığını belirlemek için ona ait kod kelimelerini ayrı ayrı karşılaştırmak yerine sıfırdan farklı tüm kod kelimelerinin Hamming ağırlıklarını bulmak daha kolay bir yöntem olur. Bir kod kelimesinin Hamming ağırlığı kod kelimesini oluşturan 1'lerin sayısıdır. Örneğin (1110010)'ın Hamming ağırlığı 4, (1111111)'in Hamming ağırlığı 7'dir. Hesaplanan Hamming ağırlıklarının minimum değeri o kodun minimum uzaklık değerini verir⁽¹⁷⁾.

Minimum uzaklıkla bir blok kodun hata sezme kapasitesi arasındaki ilişki şudur: İletim sonrasında alınan kod kelimesindeki hatalı bit sayısı t olsun. Eğer t değeri d_{\min} sayısından küçükse yani;

$$t \leq d_{\min} - 1 \quad (2.8)$$

şartı sağlanıyorsa bu kod kelimesi alıcı tarafından hatalı olarak belirlenebilir. Buradan da görülüyor ki d_{\min} değeri ne kadar büyük olursa yani kod kelimeleri arasındaki bit konumları farklılığı ne kadar çoksa, kanalda oluşacak o kadar fazla sayıda bit hatası blok kod tarafından sezilebilir⁽¹⁷⁾.

Minimum uzaklıkla hata düzeltme kapasitesi arasındaki ilişki ise şudur: Alınan kod kelimesindeki hatalı bitlerin sayısı t olmak üzere;

$$2t + 1 \leq d_{\min} \text{ , } t \text{ tek} \quad (2.9)$$

$$2t + 2 \leq d_{\min} \text{ , } t \text{ çift}$$

şartı sağlanabiliyorsa kod kelimesindeki hatalar alıcı tarafından düzeltilebilir.

2.1.4. Kod Çözme

Lineer blok kod eğer hata sezme amaçlı olarak kullanılırsa alıcıda yapılacak işlem çok kolay olmaktadır. Çünkü alıcıda yapılacak iş alınan r kod kelimesinin sendromunu hesaplamaktır. Sendrom sıfırdan farklı olursa kod kelimesi hatalı demektir ve vericiye kod kelimesinin tekrar iletilmesi isteği gönderilir. Sendromun sıfır olması ise kod kelimesinin hatasız olduğunu gösterir, doğru olarak elde edilen mesaj kullanıcıya verilir⁽¹⁷⁾.

Eğer blok kod hata düzeltme amaçlı olarak kullanılırsa o zaman kod çözme işlemi gerekmektedir. Kod çözme işleminde de yapılan ilk iş yine kod kelimesinin sendromunu hesaplamaktır. Sendrom sıfırsa yani alınan kod kelimesinde hata yoksa kod çözme işlemi kolay olacaktır. Çünkü bir (n,k) blok kodunda kod kelimesinin sonundaki k adet bit mesaj bitleridir, dolayısıyla bu bitler kod kelimesinden kolaylıkla çıkarılır ve kullanıcıya mesaj olarak verilir⁽¹⁷⁾.

Sendromun sıfırdan farklı çıkması durumunda hata vektörü (e) hesaplanır. Hata vektörü kod kelimesinde hatalı olan bitin konumunu gösterir. Dolayısıyla alıcıya gelen bir r kod kelimesi şu eşitlikle ifade edilir;

$$r = v + e \quad (2.10)$$

Örneğin (1110010) kod kelimesi iletme verildiğinde 6. bit bozulmaya uğramış olsun. Bu durumda hata matrisi (0100000) olacaktır. Alınan kod kelimesi de

$v = (1010010)$ olacaktır. Yani $(1110010) + (0100000) = (1010010)$ eşitliği bulunmaktadır. Alıcıda e hata vektörü hesaplandığında iletilen v kod kelimesi de şu şekilde bulunacaktır;

$$v = r + e \quad (2.11)$$

Yukarıdaki örneği kullanacak olursak, $r+e$ toplamı, $(1010010) + (0100000) = (1110010)$ şeklinde orijinal kod kelimesini vermektedir. Görüldüğü gibi bütün mesele hata vektörünü bulmakta yatmaktadır. Alınan kod kelimesinin sendromunu hesaplayacak olursak;

$$\begin{aligned} S &= r \cdot H^T \\ &= (v+e) \cdot H^T \\ &= v \cdot H^T + e \cdot H^T \\ &= e \cdot H^T \end{aligned} \quad (2.12)$$

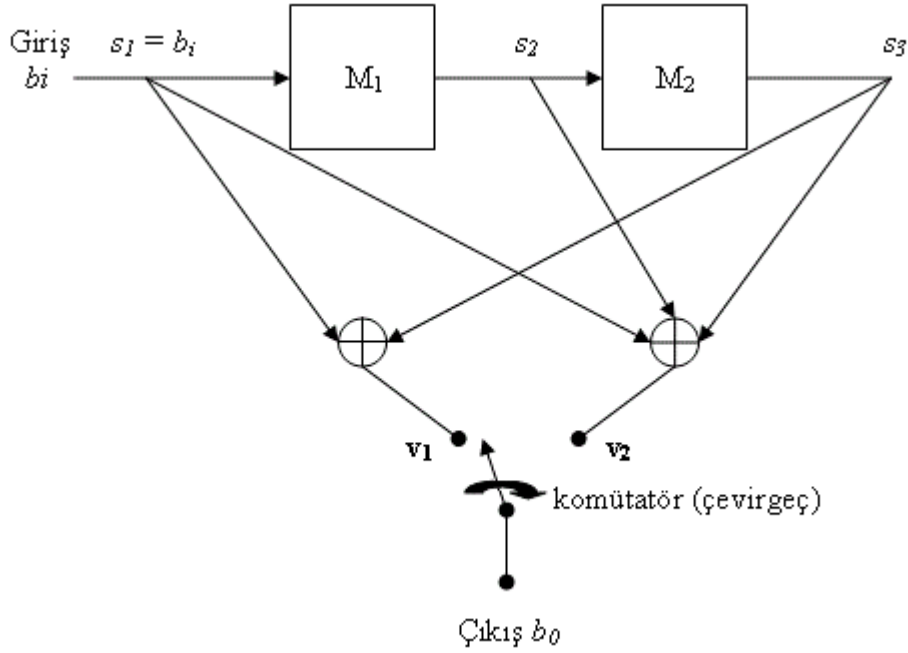
eşitliği elde edilir ki hata vektörü işte bu eşitliğin çözülmesiyle bulunur. Sonuç olarak bir kod kelimesini çözme işlemi üç adımdan oluşmaktadır;

- i- $S = r \cdot H^T$ eşitliği kullanılarak, alınan kod kelimesinin sendromu hesaplanır.
- ii- Hesaplanan sendrom vektörüne ait hata kalıbı, kullanılan kodun düzeltilebilir hata kalıpları tablosundan okunur ki bu değer $S = e \cdot H^T$ eşitliğini sağlayan kalıptır.
- iii- Hata kalıbı, $v = r + e$ eşitliğinde yerine konularak kod kelimesi çözülür.

2.2. Konvolüsyon Kodları

2.2.1. Kod Üretimi

Konvolüsyon kodları, K sayıda kaydırmalı kaydedicinin çıkışlarını ayrıcalıklı VEYA lojik toplayıcılar kullanılarak toplanmasıyla üretilirler. $K = 2$ ve $v = 2$ çıkış için bu kodlayıcı Şekil 2.1’de gösterilmektedir⁽²⁰⁾.



Şekil 2.1. Konvolüsyon Kodlayıcısı

Burada; M_1 - M_2 1 bitlik hafıza elemanlarıdır. Toplayıcıların v_1 , v_2 çıkışları;

$$\begin{aligned} v_1 &= s_1 \oplus s_2 \\ v_2 &= s_1 \oplus s_2 \oplus s_3 \end{aligned} \quad (2.13)$$

şeklindedir.

Kodlayıcının çalışması aşağıdaki gibidir; başlangıçta kaydırmalı kaydedicinin boş olduğunu kabul ediyoruz. Giriş veri bitleri dizisinin birinci biti, M_1 'e girilir. Bu

bit aralığı süresince komütatör toplayıcıların v_1, v_2 çıkışlarını örnekler. Bu durumda mevcut zamanda 1 bitlik giriş kodlanmış 2 bitlik çıkışı verir. Bu nedenle kodlayıcının kod oranı $1/2$ 'dir. Takip eden bit kodlayıcıya girilirken önceki bit M_1 'den M_2 'ye aktarılır ve komütatör tekrar tüm toplayıcı çıkışlarını örnekler. Bu işlem mesajın son bitinin M_1 'e girilmesine kadar devam eder. Bundan sonra, bütün mesaj bitlerinin işleme alınması ve dolayısıyla komple kodlama işlemi gerçekleştirilmesi için hafıza elemanı sayısınca 0 bitleri kodlayıcıya girilir ve bunlar da kodlanır. Sonuçta en son mesaj biti M_2 kaydırmalı kaydediciyi terk etmiş, hafıza elemanları boşaltılmış ve kodlayıcı sonraki mesajlar için hazır hale getirilmiş olur .

Eğer mesaj dizisindeki bitlerinin sayısı L ise, çıkışta elde edilen kodlanmış dizideki bitlerin sayısı $v(L+K)$ olur. Uygulamanın bir gereği olarak K nispeten çok küçük sayı iken, L genellikle oldukça büyük bir sayıdır. Bu yüzden $v(L+K) \approx vL$ 'dir. Sonuç olarak kod bitlerinin sayısı, v kere mesaj biti sayısıdır. Bu yüzden kodun kod oranı $1/v$ 'dir⁽²⁰⁾.

Konvolüsyon kodlayıcının işlemi sürekli bir işlemdir, giriş bitlerinin sayısı ne kadar çok olursa olsun dizi kodlayıcıda sürekli şekilde işlenir. Her bit kaydırmalı kaydedicinin bölümlerinde mesaj bitleri aralığı kadar bekler. Dolayısıyla her bir giriş biti v bitlerinin K grubunu etkiler.

Kodlanmış çıkış, kaydırmalı kaydedicinin bölümlerinin K sayısına, kullanılan ayrıcalıklı VEYA toplayıcıları sayısına ve kaydırmalı kaydedicinin bölümlerinin toplayıcılara bağlantı şekline bağlıdır. En uygun kodlayıcıların araştırılması açısından uzun ve uğraştırıcı bilgisayar araştırmaları yapılmıştır. Kod oranı $1/2$ olan kodlayıcının bazı K değerleri için sonuçları aşağıdaki Tablo 2.1'de gösterilmektedir.

Tablo 2.1. Konvolüsyon Kodlayıcının 1/2 Kod Oranı İçin En Uygun Biçimi

K	v_1	v_2
3	1,1,1	1,0,1
4	1,1,1,1	1,1,0,1
5	1,1,1,0,1	1,0,0,1,1
6	1,1,1,0,1,1	1,1,0,0,0,1
7	1,1,1,1,0,0,1	1,0,1,1,0,1,1
8	1,1,1,1,1,0,0,1	1,0,1,0,0,1,1,1

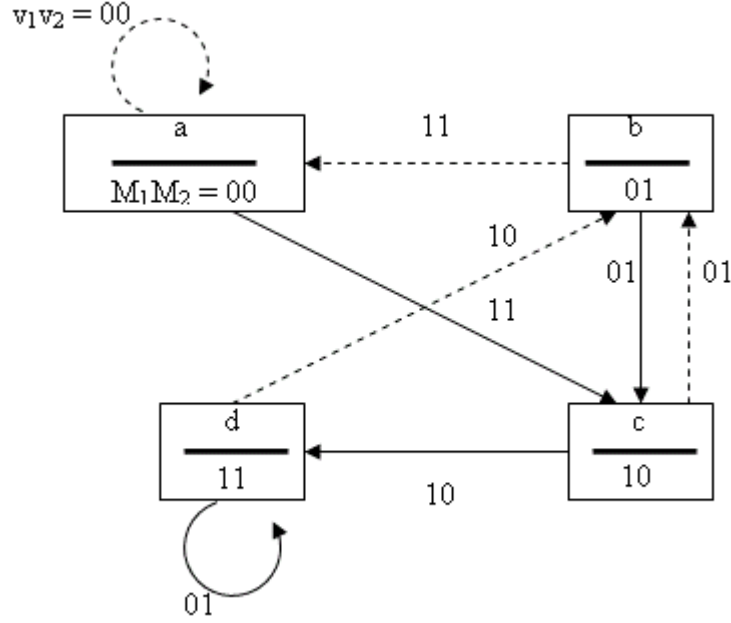
Tabloda, 0'lar bağlantı yapılmadığını gösterirken 1'ler toplayıcılara bağlantıyı temsil eder. Örneğin doğrultusunda $K = 5$ durumu için tabloda 1, 2, 3 ve 5 bölümleri v_1 'i üreten toplayıcıya, 1, 4 ve 5 bölümleri v_2 'yi üreten toplayıcıya bağlanır. Daha geniş tablolar literatürde mevcuttur⁽²⁰⁾.

2.2.2. Durum ve Kafes Diyagramı

Bir konvolüsyon kodlayıcı, durum diyagramı tarafından tanımlanabilir. Şekil 2.1'deki konvolüsyon kodlayıcı göz önüne alınır. Kod oranı 1/2 olan kodlayıcıda ; $v_1 = s_1 + s_3$ ve $v_2 = s_1 + s_2 + s_3$ 'dür. Her bir k aralığındaki v_1 ve v_2 çıkışları, o aralığın başlangıcında kodlayıcıya girilen bite ve kodlayıcının önceki durumuna bağlıdır. Bu yüzden Şekil 2.1'deki kodlayıcının hafıza elemanlarının $M_1M_2 = 00, 01, 10$ ve 11 olan mümkün 4 durumu sırasıyla a, b, c ve d olarak isimlendirilir⁽²⁰⁾.

Herhangi bir giriş dizisine kodlayıcının cevabı Şekil 2.2'de ki durum diyagramı ile gösterilir. Burada dört durum ve aralarındaki geçişler oklar ile

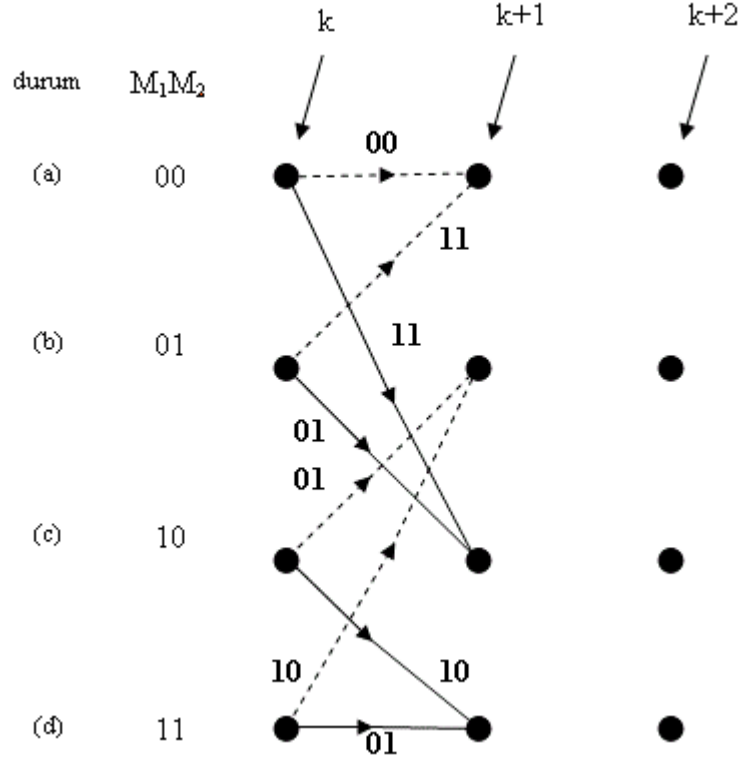
gösterilmiştir. Kesik çizgili oklar kodlayıcıya girilen 0 bitinin sağladığı geçişi, düz oklar kodlayıcıya girilen 1 bitinin sağladığı geçişi temsil eder⁽²⁰⁾.



Şekil 2.2. Konvolüsyon kodlayıcı için durum diyagramı

Her bir ok, durum ve sonraki giriş tarafından belirlenen kodlayıcı çıkışı v_1v_2 şeklinde geçiş üzerinde işaretlenir. Örneğin kodlayıcının k aralığında a ($M_1M_2 = 00$) durumunda olduğunu ve $k+1$ aralığında M_1 'e 1 bitinin girildiğini kabul edelim. Bu durumda, bu giriş biti kodlayıcının c durumuna geçmesine neden olacaktır. $k+1$ aralığı süresince kodlayıcı c ($M_1M_2 = 10$) durumda olur ve $v_1v_2 = 11$ çıkışını üretir. Her bir durumda iken kodlayıcıya 0 yada 1 bitinin girileceği olasılığı dikkate alınarak benzer şekilde durum diyagramının çizimi tamamlanır.

Konvolüsyon kodlayıcı için diğer bir gösterim yolu ise Forney tarafından geliştirilen kafes diyagramıdır⁽¹⁹⁾. Bir kodlayıcı için kafes diyagramı, giriş biti 0 olduğu zaman üst dalı, 1 olduğu zaman alt dalı izler⁽²⁰⁾.



Şekil 2.3. Konvolüsyon kodlayıcı için kafes diyagramı

k aralığında kodlayıcının b durumunda olduğunu varsayalım. Sonraki giriş biti a durumuna yada c durumuna giden iki dalın birisi üzerinden taşınacaktır ve $k+1$ aralığı süresince kodlayıcı yeni durumda kalacaktır. İkinci bit kodlayıcıyı $k+1$ 'den $k+2$ 'ye a durumundan 2 ve c durumundan 2 olmak üzere 4 mümkün dalın herhangi biri üzerinden taşıyacaktır. Önceki durumların her birinden ayrılan dal sayısı 2 'dir. 4 durum olduğuna göre $k+2$ 'den $k+3$ 'e geçmek için mümkün dalların sayısı $4 \times 2 = 8$ olur. Bundan sonra elde edilen dalların sayısı her zaman 8 'dir. O zaman kafes içindeki olası yolların toplam sayısı $2 \times 4 \times 8 \times 8 \times \dots$ 'dir. Kafes içinde k 'dan $k+l$ 'ye olası toplam yol sayısı $8^{l-1} = 2^{3(l-1)}$ olur⁽²⁰⁾.

2.2.3. Konvolüsyon Kodlarda Kod Çözme ve Viterbi Algoritması

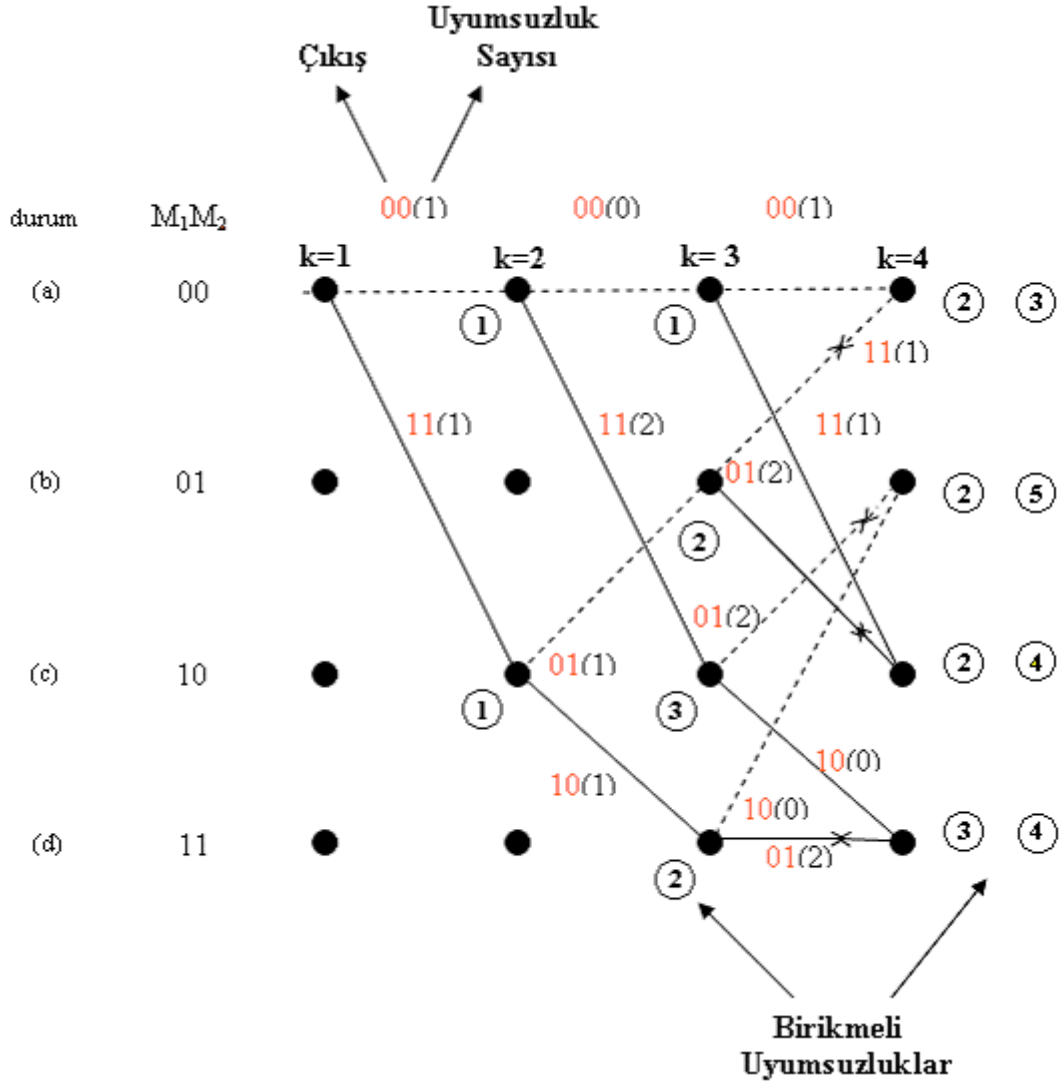
Konvolüsyon kodlayıcısının kodlama işlemi, hafıza elemanları boş olan kodlayıcı ile başlanır, ondan sonra kodlayıcı içinden bilgi bitlerinin akışı sağlanarak kodlama yapılır ve kodlama işlemi bittikten sonra kodlayıcının boşaltılarak bir sonraki işlem için hazır hale getirilmesi için kodlayıcı boşaltılır. Bunun için kodlanacak diziye dizinin en son biti kodlayıcıyı terk edinceye kadar boş bit olarak anılan 0 bitleri eklenir.

Kod çözme işleminde ise, başlangıç noktasından son noktaya kadar kafes diyagramındaki olası yolların tamamı hesaba katılır. Her mümkün giriş bitleri dizisi kendi yolunu üretir. Her bir yol için, kod çözücü bir çıkış bitleri dizisine karar verir. Karar verilen çıkış bitleri dizisi alınması olası bit dizileriyle karşılaştırılır. Karar verilen çıkış bitler dizisi olası yollardan birisi ile özdeş ise bu yolun dikkate değer yol olduğu varsayılır. Eğer birbirlerine tam benzerlik yoksa, o zaman alınan diziyle karşılaştırıldığında en az bit uyumsuzluğu üreten bir yol giriş dizisi olarak kabul edilir. Yolların sayısı dizi uzunluğu ile üssel olarak artar⁽²⁰⁾.

Viterbi algoritmasını açıklamak ve göstermek için Şekil 2.1'deki konvolüsyon kodlayıcısı kullanılır. Başlangıçta kodlayıcının boş olduğunu kabul edilirse, kodlayıcı Şekil 2.3'deki a durumundadır ($M_1M_2 = 00$). 5 bitlik bir enformasyon dizisi kodlayıcıda kodlandıktan sonra iletilsin ve iletim sonunda çıkış bitleri (v_{1R}, v_{2R}) aşağıdaki gibi alınmış olsun⁽²⁰⁾.

$$v_{1R}, v_{2R} = 10 \ 00 \ 10 \ 00 \ 00 \quad (2.14)$$

Başlangıç durumu a olduğundan eğer ilk enformasyon biti 0 olursa alınan ilk bit çifti 00, 1 olursa alınan ilk bit çifti 11 olur. Bu durumda alınmış olan ilk bit çifti 10 olduğundan her iki durumla da uyumsuzdur ve hata oluşmuştur⁽²⁰⁾.



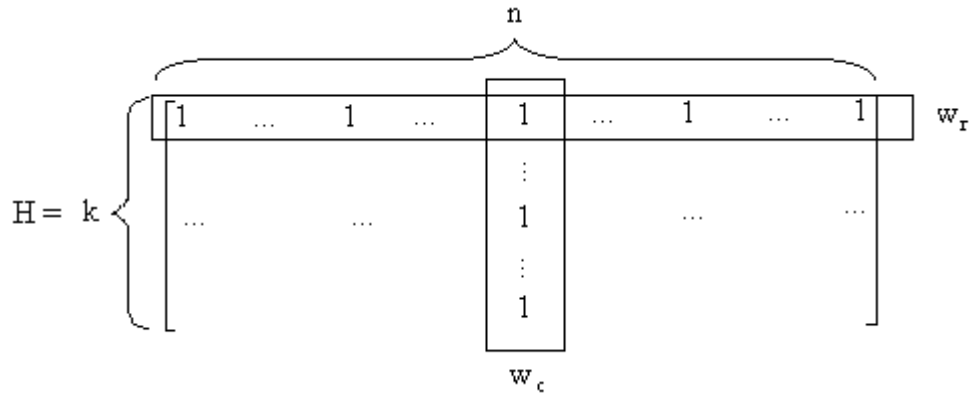
Şekil 2.4. Kafes boyunca mümkün yollarının gösterilmesi

Kodlayıcının kafes diyagramında gösterildiği gibi, mümkün yollardan kodlayıcıya doğru kod çözme algoritmasını izleyelim. $k=1$ aralığında giriş bitinin 0 olması durumunda, 00 çıkışı üretilir ve kodlayıcı tekrar a durumuna döner (kesikli çizgi). Giriş bitinin 1 olması durumunda, 11 çıkışı üretilir ve kodlayıcıyı c durumuna gider. Bu yolla kafes diyagramı Şekil 2.4’de gösterildiği gibi çizilir. Alınan dizinin bitleri kafes diyagramındaki yolları birleştirilen bitler ile karşılaştırılır. Her bir periyotta ki uyumsuzlukların sayısı Şekil 2.4’de parantez içinde gösterilmiştir.

Gerçek çıkış ile 10 arasındaki uyumsuzluk bitlerinin sayısı birdir. Sonraki aralıkta, giriş 0 olunca çıkış tekrardan 00 olmalıdır ve alınan bitlerle karşılaştırıldığında uyumsuzluk yoktur. Sonuç olarak, her bir duruma ulaşıldığında karşılaşılan bit uyumsuzluklarının toplamı çember içinde Şekil 2.4’de gösterilmiştir. Viterbi algoritması bu şekilde çizilen kafes diyagramındaki toplam uyumsuzluk sayısı en az olan yolu seçecektir ve kod çözme işlemi gerçekleşecektir⁽²⁰⁾.

2.3. Düşük Yoğunluklu Eşlik Kontrol (LDPC) Kodları

Düşük yoğunluklu eşlik kontrol kodları (LDPC), eşlik-kontrol matrisi H ’nin düşük yoğunluklu olarak 1’lere sahip olduğu bir tür blok kodudur. “Düşük-yoğunluk” ile kastedilen kodun özelliğini belirleyen eşlik-kontrol matrisinin blok kodunun matrisine göre daha az sayıda 1, daha fazla sayıda 0 içermesidir. Düzenli LDPC kodu, H eşlik-kontrol matrisinin, her bir sütununun aynı w_c sayıda 1 içermesi ve her bir satırının da $w_r = w_c (n/m)$ sayıda 1 içermesidir. Burada m , $n-k$ tane eşlik-kontrol bitini gösterir ve $w_c \ll m$ ’dir. Kod oranı $R = k/n$ olduğundan, bu parametreler ile kod oranı arasında $R = 1 - w_c/w_r$ şeklinde bir ilişki mevcuttur. H düşük yoğunluklu olduğu halde her bir satır ve sütunundaki 1’lerin sayısı sabit değilse, o zaman bu koda düzensiz LDPC kodu denir. H eşlik kontrol matrisinin yapısı Şekil 2.5’de gösterildiği gibidir.



Şekil 2.5. Eşlik kontrol matrisi yapısı

2.3.1. LDPC Kodlarının Gösterimleri

2.3.1.1. LDPC Kodlarının Matrisel Gösterimi

LDPC kod biçimi, lineer blok kodların bir sınıfı olması nedeni ile LDPC kodları F_2 ikili alan üzerindeki ikili n -tapılların F_2^n bir k boyutlu c alt uzayı olarak tanımlanabilir. C 'de yayılan temel $B = \{g_0, g_1, g_2, \dots, g_{k-1}\}$ 'yi bulabiliriz, öyle ki her $c \in C$ bazı $\{u_i\}$ için $c = u_0 \cdot g_0 + u_1 \cdot g_1 + \dots + u_{k-1} \cdot g_{k-1}$ olarak yazılabilir. Daha kapalı olarak, $c = uG$ yazılabilir, burada $u = [u_0, u_1, u_2, \dots, u_{k-1}]$ 'dir ve G , bütün vektörleri satır vektörleri olan $k \times n$ boyutlu üreteç matrisi olarak isimlendirilir. G 'nin $(n-k)$ boyutlu C^\perp boş uzayı $xG^T = 0$ olduğu için bütün $x \in F_2^n$ vektörlerini kapsar ve $B^\perp = \{h_0, h_1, h_2, \dots, h_{n-k-1}\}$ temelinde yayılır. Böylece, her bir $c \in C$ için $(n-k) \times n$ boyutlu eşlik-kontrol matrisi olarak anılan satırları $\{h_i\}$ vektörleri olan H 'nin yer aldığı $cH^T = 0$ olur. H 'ye eşlik-kontrol matrisi denir. Çünkü alınan kod kelimesinde farklı $m = n - k$ tane eşlik-kontrol bitini gerçekleştirir^(3,4).

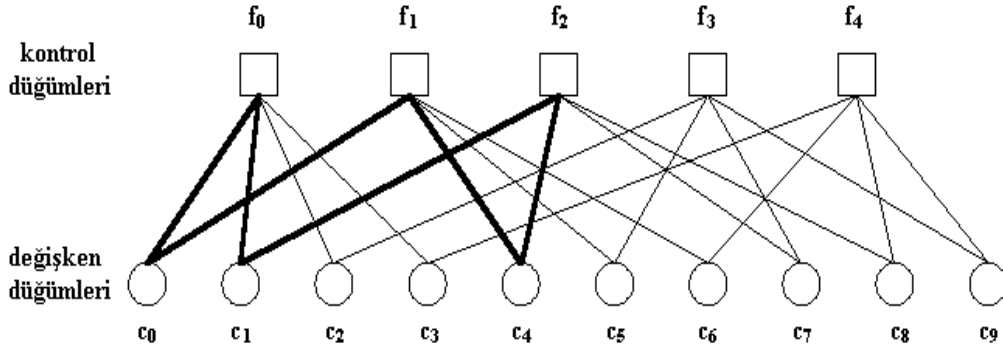
2.3.1.2. LDPC Kodlarının Grafiksel Gösterimi

Tanner, LDPC kodlarını iki parçalı grafik olarak isimlendirdiği grafik tarafından etkili olarak nasıl gösterebileceğini ortaya koydu ve gösterdi. Konvolüsyon kodun kafesine benzer olarak LDPC kodun Tanner grafiği, kodun eksiksiz olarak gösterimini sağlar ve kod çözme algoritmasının tanımlanmasına yardım eder. İki parçalı grafik, düğümlerin iki tipe ayrılabilirdiği bir grafikdir ve dallar sadece farklı iki tipteki iki düğüme bağlanabilirler. Tanner grafiğinde bu iki tip düğümler; değişken düğümleri ve kontrol düğümleridir. Kodun Tanner grafiği şu kurala uygun olarak çizilir; kontrol düğümü j , H 'nin h_{ji} elemanı 1 olduğunda her ne durumda ise değişken düğümü i 'ye bağlanır. Her biri eşlik-kontrol denklemi için olmak üzere $m = n - k$ tane kontrol düğümü ve her biri c_i kod kelimesi için olmak üzere n tane değişken düğümü vardır. Bundan başka, H 'nin m satırı m tane kontrol düğümü bağlantısını ve H 'nin n tane sütunu ise n tane değişken düğümü bağlantısını gösterir⁽⁶⁾.

(10,5) LDPC kodu için $w_c = 2$ ve $w_r = w_c(n/m)$ $w_r = 4$ ve H matrisi aşağıdaki gibi ise⁽²¹⁾;

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

H'ye bağılı Tanner grafiđi;



Şekil 2.6. Eşlik kontrol matrisine bağılı LDPC kodun Tanner grafiđi

şeklindedir.

H matrisinin sıfıncı satırı $h_{00}=h_{01}=h_{02}=h_{03}= 1$ (diđerleri sıfır) olduğundan deđişken düğümleri c_0, c_1, c_2, c_3 , kontrol düğümü f_0 ile bağılanır. Sırasıyla H'nin 1, 2, 3, 4 satırlarına karşılık gelen f_0, f_1, f_2, f_3 kontrol düğümleri içinde benzer durum geçerlidir. Aynı kontrol düğümüne bağılanan bit deđerleri toplamı sıfır olmalıdır ($cH^T = 0$). Satırlar boyunca da Tanner grafiđi yapısı oluşturulabilir. Örneđin H'nin sıfıncı satırı $h_{00}=h_{10}= 1$ olduğundan, deđişken düğümü c_0 , kontrol düğümleri f_0 ve f_1 'e gösterildiđi gibi uygun olarak bağılanır⁽²¹⁾.

Bu örnekteki Tanner grafiđi düzenlidir, her deđişken düğüm 2 tane köşeye ve her kontrol düğümü ise 4 tane köşe bağılantısına sahiptir (her deđişken düğümün derecesi 2 ve kontrol düğümün derecesi 4'dür). $w_c = 2$ ve $w_r = 4$ olduğuna göre;

$$mw_r = nw_c \quad (2.15)$$

olur.

2.3.2. LDPC Kod Tasarım Yaklaşımları

LDPC kodunun yapısı, düşük yoğunluklu eşlik kontrol matrisinin yapısına bağlıdır.

2.3.2.1. Gallager Kodları

Gallager'in LDPC kodu, düzenli LDPC kodudur ve H matris formu aşağıdaki gibidir.

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \cdot \\ \cdot \\ H_{w_c} \end{bmatrix} \quad (2.16)$$

buradaki H_d alt matrisler aşağıdaki yapıya sahiptir. μ ve w_r 1'den büyük sabitlerdir. Her alt matris H_d , sütun boyutu 1 ve satır boyutu $\mu \times \mu w_r$ ile ilişkilendirilir. Alt matris H_i şu forma sahiptir; $i = 1, 2, \dots, \mu$ için, i . satır, $(i-1)w_r+1$ den $i.w_r$ sütunlarında tüm w_r 'ler 1 olmaktadır. Diğer alt matrisler H_1 'in basit sütun işlemleriyle elde edilir. Bu yapıya göre düzenlenmiş olan H matrisi düzenlidir ve satır uzunluğu w_r ve sütun uzunluğu w_c olan $\mu w_c \times \mu w_r$ boyuta sahiptir⁽²²⁾.

	11111111
	11111111
	11111111
H =	1. bloğun basit sütun işlemleriyle elde edilen alt matrisi
	1. bloğun basit sütun işlemleriyle elde edilen alt matrisi

Şekil 2.7. Gallager'in eşlik kontrol matrisi yapısı

Gallager düzenli LDPC kodlarının H eşlik kontrol matrisinin yapısının $w_c \geq 3$ ve $w_r > w_c$ boyutlarında mükemmel bir özelliğe sahip olduğunu göstermiştir. LDPC kodun kod çözümünde yinelemeli kod çözme algoritmasını kullanmıştır ve kod kelimesi uzunluğu büyük olduğunda bu yöntemle kod çözmenin performans kapasitesini sınırladığı saptamıştır^(3,4).

2.3.2.2. MacKay Kodları

MacKay seyrek H matrisleri ile ikili kodların tasarımının faydalarını keşfetmiştir ve bu tür kodlamanın kapasite limit değerine yakın bir performans yeteneği olduğunu göstermiştir⁽⁵⁾. MacKay⁽⁵⁾ algoritmasında seyrek H matrisini yarı-rasgele olarak üretmiştir. Bu algoritma şu şekildedir;

- 1- H matrisi, sabit satır uzunluğu ve w_c sütun uzunluğu olmak üzere rasgele olarak üretilir.

- 2- H matrisi, w_r satır uzunluğu garantiye alınırken iki sütun arasındaki örtüşmenin 1'den büyük olmaması koşuluyla w_c sütun uzunluğu ile rasgele olarak üretilir.
- 3- H matrisi 2'deki gibi üretilir, ayrıca kısa periyotlardan kaçınılır.
- 4- H matrisi 3'deki gibi üretilir, ayrıca $H = [H_1 H_2]$ şeklinde yapılandırılır. H_2 matrisi tersine çevrilebilir matris olmalıdır veya en azından H tam rank olmalıdır.

Bu şekilde üretilmiş olan H matrisi yardımıyla kodlanan MacKay kodlarının eksikliği, düşük karmaşık kodlama sağlama yapısının yeterli olmayışıdır. Kodlama, H matrisinin Gauss-Jordan eleme yöntemiyle $[P^T I]$ şekline getirildikten sonra, sistematik yapıdaki $G = [I P]$ üreteç matrisi ile yapılır. Kodlamada ki bu problem G matrisinin alt matrisi olan P matrisinin seyrekliğinin olmayışıdır. Bu yüzden kod kelimesinin $n = 1000$ veya daha fazla genişlikte olması kodlamanın karmaşıklığını arttıracaktır⁽²²⁾.

2.3.2.3. Düzensiz LDPC Kodları

Richardson⁽¹³⁾, ve Luby⁽¹⁴⁾, düzensiz LDPC kodlarının derece dağılım polinomlarını $\lambda(x)$ ve $\rho(x)$ olarak tanımlamışlardır ve bu tür kodların bu polinomlarla kanalların özelliklerine göre nasıl uygunlaştırılacağını göstermişlerdir.

Düzensiz LDPC kodları için w_c ve w_r parametreleri satır ve sütun sayılarının fonksiyonudur ve gösterimleri farklıdır. Bunun yerine literatürde kullanılan değişken düğüm ve kontrol düğümü derece dağılım polinomları $\lambda(x)$ ve $\rho(x)$ tarafından elde edilir ve gösterilirler⁽¹³⁾.

Bu polinomlar;

$$\lambda(x) = \sum_{d=1}^{d_v} \lambda_d x^{d-1} \quad (2.17)$$

dir. Burada; λ_d , tüm köşelerin d-dereceli değişken düğümlere bağlandığı parçayı belirtir ve d_v , maksimum değişken düğüm derecesidir. Aynı şekilde;

$$\rho(x) = \sum_{d=1}^{d_c} \rho_d x^{d-1} \quad (2.18)$$

dir. Burada; ρ_d , tüm köşelerin d-dereceli kontrol düğümlerine bağlandığı parçayı gösterir ve d_c maksimum kontrol düğüm derecesidir.

Yukarıda verilen düzenli LDPC kodu için $w_c = d_v = 2$ ve $w_r = d_c = 4$ olduğundan; $\lambda(x) = x$ ve $\rho(x) = x^3$ olur.

2.3.3. LDPC Kod Çözümü

2.3.3.1. Yinelemeli Kod Çözümü

Bir yinelemeli kod çözme yolu olan bit-çevirme algoritması, her bir alınan bitin anlık kuvvetli kararı (0 veya 1) değerlendirmesine dayanmaktadır. Yinelemeli kod çözmenin yapılabilmesi için gerekli olan kodun Tanner grafiğindeki düğümler arasındaki mesajların geçişidir. Bit-çevirme algoritması için mesajlar basittir: Eğer alınan bit 1 yada 0 ise ve tanımlanan düğüme bağlı olan bit düğümü kontrol düğümlerinin her birine alınan mesajı gönderir ve her kontrol düğümü eşlik kontrol eşitliğinin sağlanması yada sağlanmaması durumuna bakmaksızın bit düğümünün her birine mesajı gönderir⁽¹¹⁾.

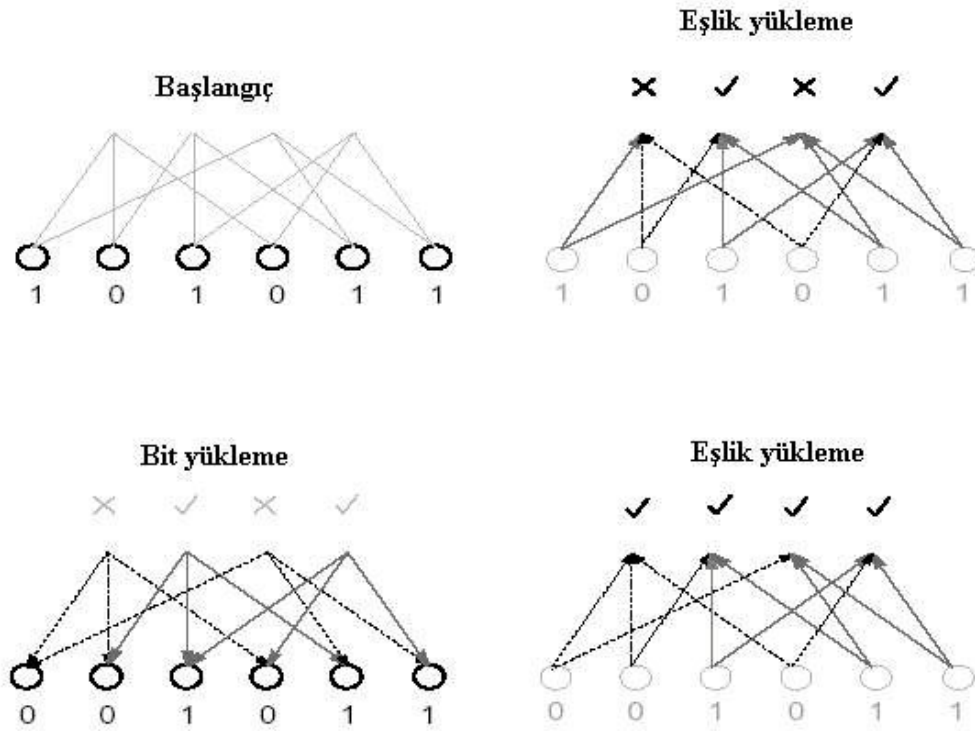
Bit-çevirici kod çözüm algoritması aşağıdaki gibidir⁽²¹⁾.

1. Adım: Başlangıç: Her bir düğüm kanaldan alınan bit değeriyle atanır ve bu değeri gösteren kontrol düğümüne mesajlar yollanır.

2. Adım: Eşlik yükleme: Her kontrol düğümü bit düğümündeki mesajları kullanarak eşlik kontrol denkleminin sağlanıp sağlanmadığını hesaplar. Eğer tüm eşlik kontrol denklemleri sağlanırsa algoritma sonuçlanır yoksa her kontrol düğümü eşlik kontrol denklemini sağlayıp sağlamadığını gösteren bağlı bit düğümlerine mesaj yollar.

3. Adım: Bit yükleme: Eğer her bit düğümünden alınan mesajların çoğu "sağlanmamış" ise, bit düğümü geçerli değerini çevirir, yoksa değer saklanır. Eğer izin verilen maksimum yinelemelerin sayısı algoritmanın bitişine ulaşırsa yakınsamada ki başarısızlık bildirilir, yoksa her bit düğümü yeni mesajları bağlı olduğu kontrol düğümüne yollar ve 2. adım tekrarlanır.

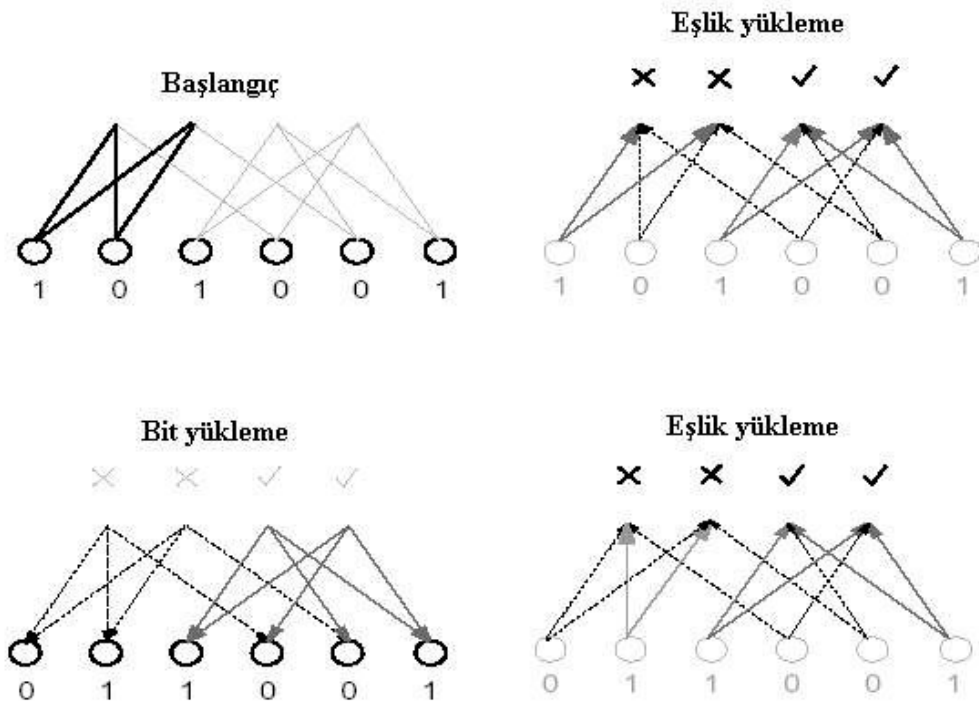
Bit çevirici kod çözme işlemini özetlersek, örneğin kod kelimesi $c = 001011$ ise kanal çıkışından alınan kod kelimesi $r = 101011$ olarak alınsın. Alınan kod kelimesinin kod çözümü için gerekli adımlar Şekil 2.8'de gösterilmiştir⁽²¹⁾.



Şekil 2.8. Örnek bir bit çevirim işlemi

1. adımda bit değerleri sırasıyla 1, 0, 1, 0, 1, ve 1 olarak belirlenir ve mesajlar bu değerleri gösteren kontrol düğümlerine gönderilir. 2. adımda eşlik kontrol denklemleri sadece çift numaralı bitleri içeren, eşlik kontrol denklemleri 1 olduğunda sağlanır. Birinci ve üçüncü kontrol düğümleri için bu durum sağlanmamıştır ve bağlı olduğu bitlere "sağlanmamış" mesajını yollar. 3. adımda birinci bit "sağlanmamıştır" mesajını gösteren mesajlar çoğunluğuna sahiptir, bu yüzden değerini 1'den 0'a çevirir. Adım 2 tekrarlanır ve tüm dört eşlik kontrol denklemini sağlanır, algoritma durur ve kodu çözülmüş kod kelimesi $c = 001011$ 'e döner. Bu yüzden alınan kod kelimesi olası kod kelimeleri açıkça araştırmayı gerektirmeden doğru bir şekilde kodu çözülür.

Kodun Tanner grafiğindeki periyodun varlığı, yinelemeli kod çözüme işleminin etkinliğini azaltır. Örnek verirse, periyodun dört olmasının bozucu etkisi bir önceki örnekteki kanal çıkışından alınan kodu aşağıdaki şekildeki yeni kodu elde etmek için ayarlanır. Bu kod için geçerli kod kelimesinin 001001 olduğu fakat ilk bitin bozulduğu farz edilir. Bu yüzden kanaldan $r = 101001$ alınır. Bu kod kelimesi için bit çevirim algoritmasının adımları Şekil 2.9'da gösterilmiştir⁽²¹⁾.



Şekil 2.9. Periyodun 4 olduğu bir bit çevrim kod çözme işlemi

Anlık bit değeri sırasıyla 1, 0, 1, 0, 0, 1 ve mesajlar bu değeri gösteren kontrol düğümlerine gönderilir. 2. adım gösteriyor ki birinci ve ikinci eşlik kontrol denklemleri sağlanmıyor. 3. adımda da bir ve ikinci eşlik kontrol denklemleri sağlanmıyor. 3. adımda bir ve ikinci bitlerin ikisi de "sağlanmamıştır" mesajlarının çoğunluğuna sahiptir bu yüzden bit değerlerini çevirir 2. adım tekrarlandığında

görüyoruz ki birinci ve ikinci eşlik kontrol denklemleri sağlanmıyor. Bu noktada daha fazla yinleme ilk iki bitin arasından bir tanesi çevirme değerinin hep hatalı olmasına sebep olur ve algoritma başarısız olur. Sonuç olarak periyodun dört olması durumunda, kod kelimesinin ilk iki bit değeri aynı eşlik kontrol denklemleri içinde olduğundan her iki eşlik kontrol denklemlerinde sağlanmaz. Yani hangi bitin hataya sebep olduğunu bulmak imkansızdır⁽²¹⁾.

2.3.3.2. Toplam-Çarpım Kod Çözümü

Toplam-çarpım kod çözüm algoritması ilk olarak Gallager tarafından 1962'de sahte-rasgele yapılı LDPC kodlarını içeren tezinde tanıtmıştır^(3,4). Toplam-çarpım algoritmasıyla çözülmüş 10^7 blok uzunluklu yüksek kullanımlı düzensiz LDPC kodları Shannon'un limit değerine iki-girişli Toplamsal Beyaz Gürültü Kanalda (AWGN) desibel'in yüzde birine yaklaşabildiği bilinmektedir⁽¹⁵⁾.

Toplam-çarpım algoritması bit çevirim algoritmasına benzer düşünülebilir. Fakat her kararı temsil eden mesajlar ile olabilir değerler logaritmik olasılık oranlarıyla sunulur. Bit-çevirimli kod çözümü, kanaldan alınan değer anlık kuvvetli karar sonucunda ikili olarak sonuçlandırıldığı, yani kanaldan alınan değer pozitifse 0 negatifse 1 olduğu bir algoritmadır. Kuvvetli karar kullanmanın kod çözerken tek kusuru sinyalin güvenilirliğine bağlı olan bilgi yani yumuşak bilgi atılır. Alınan değer büyüklüğü kararın güvenilirliğinin ölçüsüdür. Toplam-çarpım kod çözümleri gibi yumuşak karar kod çözümleri iletilen sinyal için olası ifadeleri elde etmek için kanal özelliklerinin bilgileriyle birlikte alınan yumuşak bilgiyi kullanır⁽²¹⁾.

İkili sinyal için 1 olma olasılığı p , 0 olma olasılığı $1-p$ ve logaritmik olasılık oranı (LLR) olarak gösterilir⁽²¹⁾.

$$\text{LLR}(p) = \log\left(\frac{1-p}{p}\right) \quad (2.19)$$

LLR'nin işareti kuvvetli karardır ve $|\text{LLR}(p)|$ 'nin büyüklüğü bu kararın güvenilirliğidir. Olasılıkların logaritmik gösteriminin yararı; uyarlama karmaşıklığını azaltmak için sonucun olasılıkların çarpımı olmasına rağmen, logaritmik olasılık oranlarının toplamıdır.

Toplam-çarpım kod çözümünün amacı; kod kelimesindeki her bit için sonsal olasılıkların (APP) hesaplanmasıdır. APP, $P_i = P\{c_i = 1/N\}$ tüm eşlik kontrol kısıtlamalarının sağlandığı i.inci kod kelimesinin bitinin N durumunda 1 şartlı olasılığıdır. İçsel veya önsel olasılık P_i^{int} , kod kısıtlama bilgilerinden bağımsız orjinal olasılıktır ve dışsal olasılık P_i^{ext} N durumundan ne öğrenildiğini temsil eder⁽²¹⁾.

Toplam-çarpım algoritması yinelemeli olarak her bitin yaklaşık APP değerini hesaplar. Eğer kod periyodik değilse yaklaşımlar kesindir. Bir periyottaki kontrol sınırlamalarından elde edilen dışsal bilgi, sonraki yineleme için önsel bilgi olarak kullanılır. Eşlik kontrol sınırlamasından elde edilen dışsal bit bilgisi, yinelemenin başlangıcındaki bit için önsel değerinden bağımsızdır. Sonraki yinelemelerden sağlanan dışsal bilgi, periyot yoluyla bilginin döndürülmesine kadar orjinal önsel olasılıktan bağımsız kalır⁽²¹⁾.

j'nci eşlik kontrol denkleminde kod kelimesinin i'nci dışsal olasılığını hesaplamak için, i'nci bitin 1 olduğunu kabul edilirse bu olasılık hesaplanır ve eşlik kontrol eşitliği sağlanır. Bu olasılık diğer kod kelime bitlerinin tek numaralı bitlerinin 1 olduğu olasılıktır.

$$P_{i,j} = \frac{1}{2} + \frac{1}{2} \prod_{i' \in B_j, i' \neq i} (1 - 2P_{i'}^{\text{int}}) \quad (2.20)$$

B_j notasyonu ilgilenilen kodların j 'nci eşlik kontrol denklemindeki bitlerin kolon yerlerini gösterir. Denklem (2.20) logaritmik olasılık notasyonunun içine konulursa;

$$\tanh = \left(\frac{1}{2} \log_e \left(\frac{1-p}{p} \right) \right) = 1 - 2p \quad (2.21)$$

$$\text{LLR}(P_{i,j}^{\text{ext}}) = \log_e \left(\frac{1 + \prod_{i' \in B_j, i' \neq i} \tanh(\text{LLR}(P_{i'}^{\text{int}})/2)}{1 - \prod_{i' \in B_j, i' \neq i} \tanh(\text{LLR}(P_{i'}^{\text{int}})/2)} \right) \quad (2.22)$$

elde edilir. Her yinelemedeki i . bitinin tahmin edilen APP'nin LLR'si basitleştirilirse;

$$\text{LLR}(P_i) = \text{LLR}(P_i^{\text{int}}) + \sum_{j \in A_i} \text{LLR}(P_{i,j}^{\text{ext}}) \quad (2.23)$$

elde edilir. Burada A_i , kodun i 'nci bitin eşlik kontrol denklemlerinin satır yerlerinin gösterimidir. Toplam-çarpım algoritması aşağıdaki gibidir.

1. Adım: Başlangıç: Bit düğümü i 'den kontrol düğümü j 'ye gönderilen başlangıç mesajı, kanal özellikleriyle alınan işaret y_i 'nin LLR'sidir. İşaret-gürültü oranı E_b/N_0 olan AWGN kanal için;

$$L_{i,j} = R_i = 4y_i \frac{E_b}{N_0} \quad (2.24)$$

2.Adım: Bit kontrolü: Kontrol düğümü j'den bit düğümü i'ye kadarki asıl mesaj, eğer i biti LLR olarak 1 farz edilirse eşlik kontrol j'nin sağlanması olasılığıdır.

$$E_{i,j} = \log_e \left(\frac{1 + \prod_{i' \in B_j, i' \neq i} \tanh(L_{i',j} / 2)}{1 - \prod_{i' \in B_j, i' \neq i} \tanh(L_{i',j} / 2)} \right) \quad (2.25)$$

3. Adım: Kod kelime testi: Birleşmiş LLR, dışsal LLR'lerin ve adım 1'de hesaplanan orijinal LLR'lerin toplamıdır.

$$L_i = \sum_{j \in A_i} E_{i,j} + R_i \quad (2.26)$$

Her bit için kesin karar sağlanırsa;

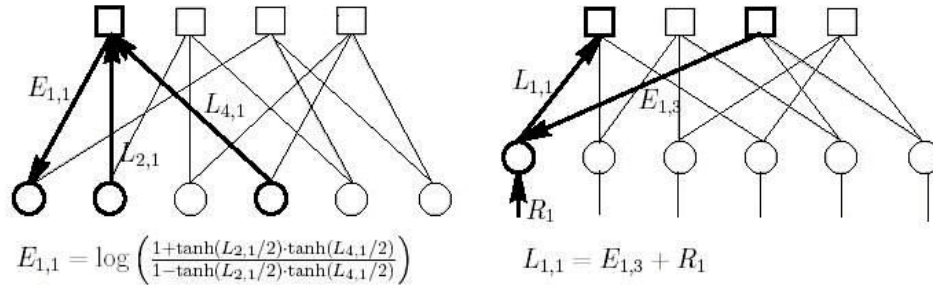
$$z_i = \begin{cases} 1, & L_i \leq 0 \\ 0, & L_i \geq 0 \end{cases} \quad (2.27)$$

olur. Eğer $z = [z_1, \dots, z_n]$ geçerli kod kelimesi $H z^T = 0$ eşitliğini sağlarsa veya izin verilen maksimum yineleme sayısında tamamlanırsa, algoritma biter⁽²¹⁾.

4. Adım: Bit Kontrol: Denklem (2.26)'ya benzer şekilde mesaj, her bit düğümünden kontrol düğümüne gönderilir. Kontrol düğümü j deki bilgiyi kullanmaksızın i bitini j kontrol düğümüne yollar ve LLR hesaplanır.

$$L_{i,j} = \sum_{j' \in A_i, j' \neq j} E_{i,j'} + R_i \quad (2.28)$$

Örnek bir kodun denklem (2.25) ve (2.28)'e uygulaması Şekil-2.10'da gösterilmiştir. Kontrol düğümünden bit düğümüne geçen dışsal bilgi, bitin olasılık değerinden bağımsızdır. Kontrol düğümlerindeki dışsal bilgiler sonraki yinelemelerde bit düğümleri için önceki bilgi olarak kullanılır⁽²¹⁾.



Şekil 2.10. Örnek bir toplam-çarpım kod çözüm algoritması

3. ARAŞTIRMA BULGULARI VE TARTIŞMA

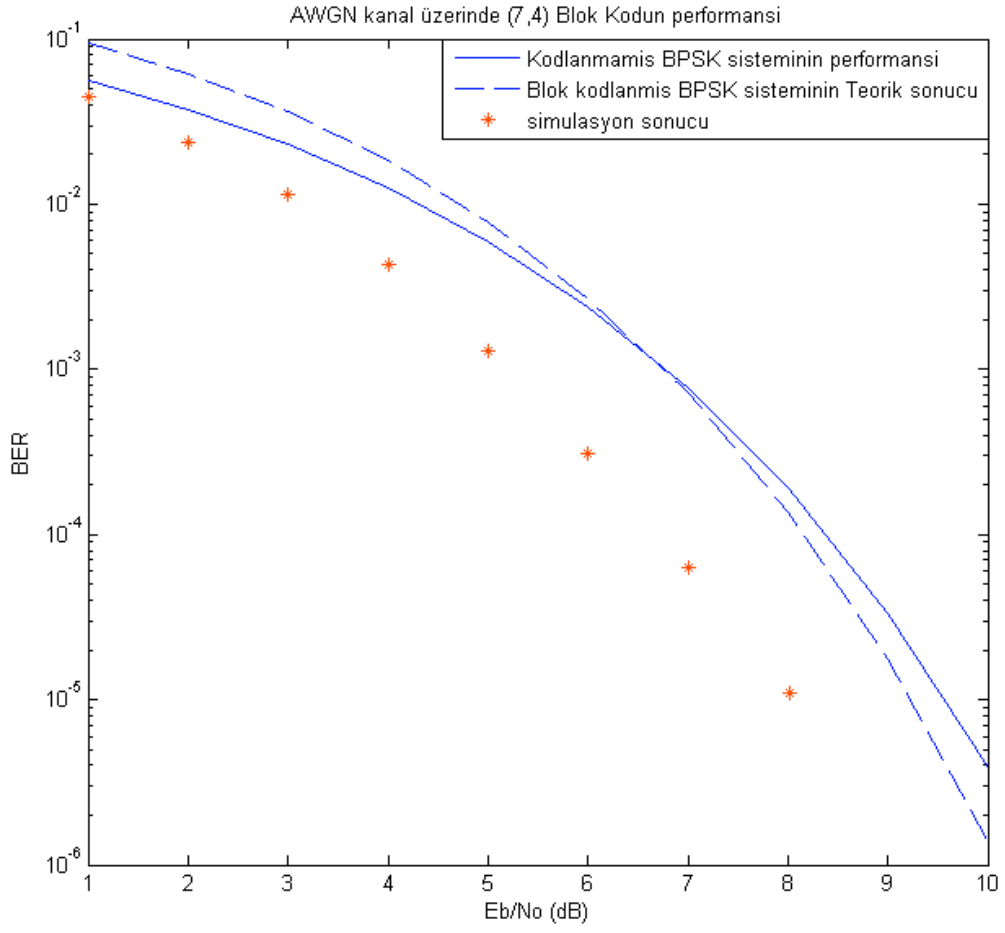
3.1. Blok Kodu Kullanan BPSK Sisteminin AWGN Kanal Üzerindeki Performans Analizi

Simülasyonda kullanılacak olan (7,4) lineer sistematik blok kodun G üreteç matrisi;

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

şeklindedir. Bu simülasyonun çalışma adımları şu şekildedir: Üreteç matrisi yardımıyla kodlanan mesaj BPSK modülasyonu ile modüle edilir. Daha sonra AWGN kanal üzerinden iletilir. Alıcı kısmında ise, kanaldan alınan işarete demodülasyon işlemi uygulanır ve kodu çözülür.

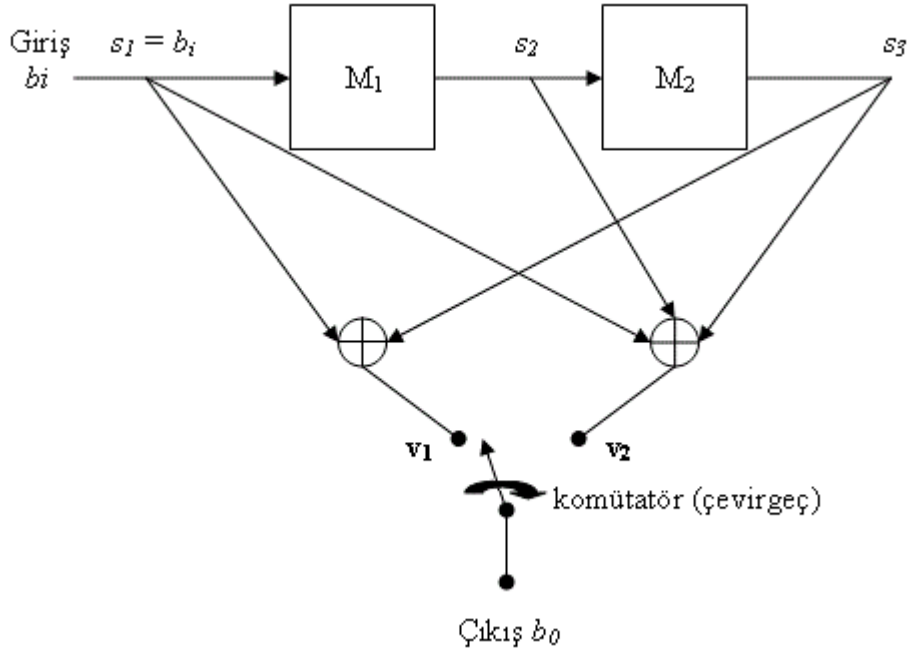
Simülasyonda, 50 KByte'lik ikili giriş mesaj dizisi (7,4) lineer sistematik blok kod tarafından kodlanmıştır. Karşılaştırma açısından kodlanmamış BPSK sistemi ve bu kodun teorik olarak performans sonuçları bulunmuştur.



Şekil 3.1: (7,4) Blok kodun AWGN kanal üzerindeki karşılaştırılmalı performansı

3.2. Konvolüsyon Kodu Kullanan BPSK Sisteminin AWGN Kanal Üzerindeki Performans Analizi

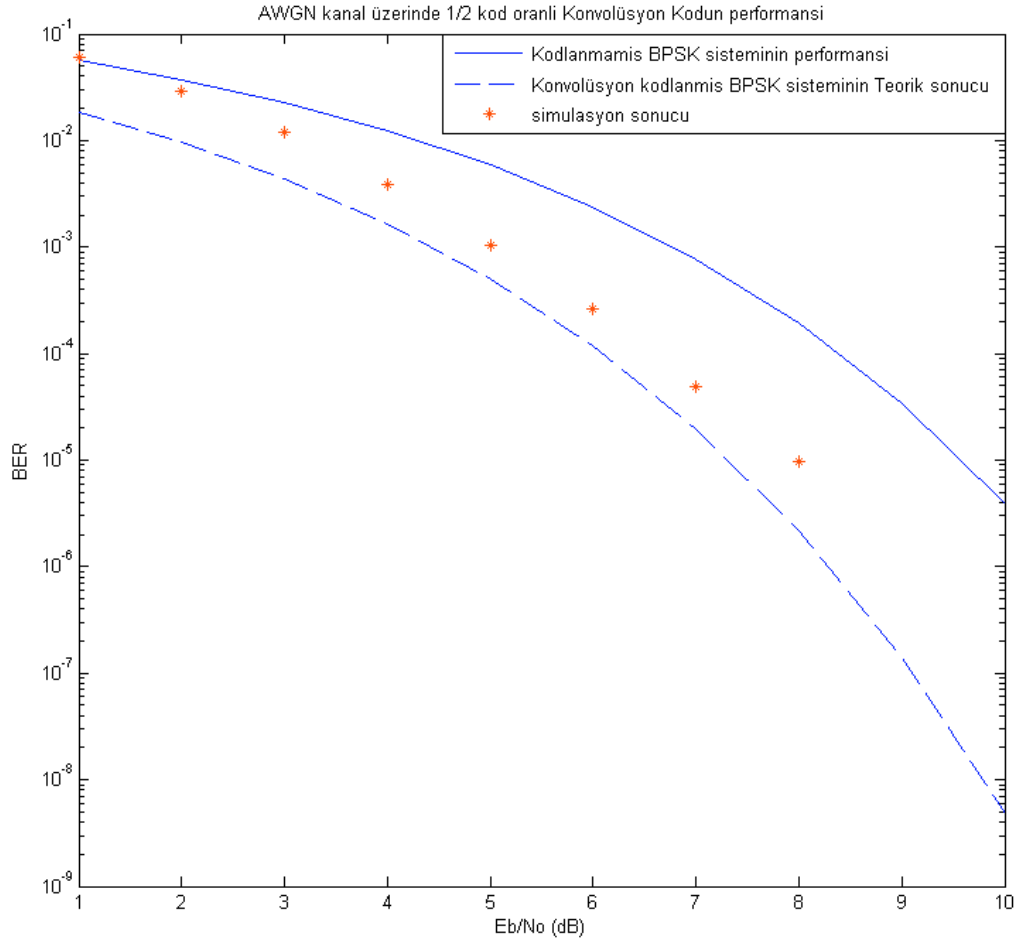
Simülasyonda kullanılacak 1/2 oranlı konvolüsyon kodlayıcı;



Şekil 3.2: 1/2 kod oranlı konvolüsyon kodlayıcısı

şeklindedir. Bu simülasyonda kodlayıcı tarafından kodlanan mesaj işareti BPSK modülasyonu ile modüle edilir. Daha sonra AWGN kanal üzerinden iletilir. Alıcı kısmında ise, kanaldan alınan işarete demodülasyon işlemi uygulanır ve viterbi algoritması kullanılarak kodu çözülür.

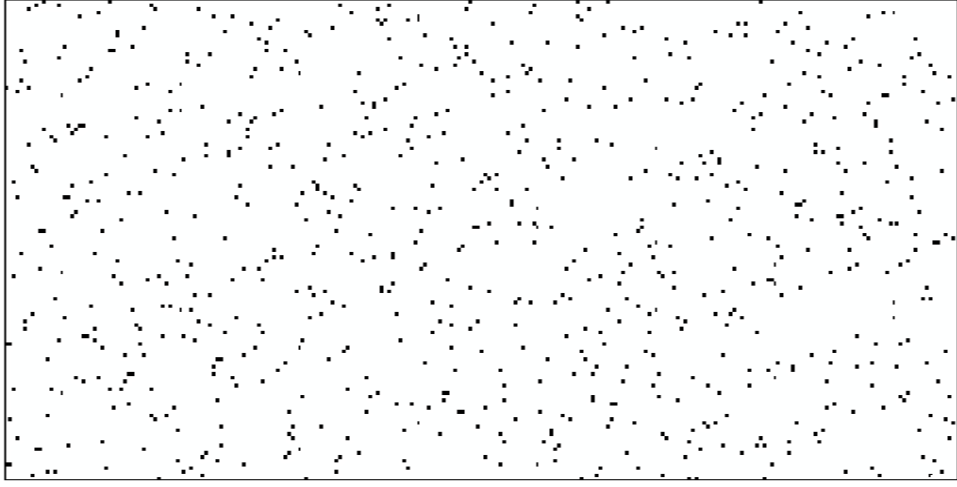
Simülasyonda, 50 KByte'lik ikili giriş mesaj dizisi 1/2 oranlı konvolüsyon kodlayıcı tarafından kodlanmıştır. Karşılaştırma açısından kodlanmamış BPSK sistemi ve bu kodun teorik olarak performans sonuçları bulunmuştur.



Şekil 3.3: 1/2 oranlı konvolüsyon kodun AWGN kanal üzerindeki karşılaştırılmalı performansı

3.3. LDPC Kodu Kullanan BPSK Sisteminin AWGN Kanal Üzerindeki Performans Analizi

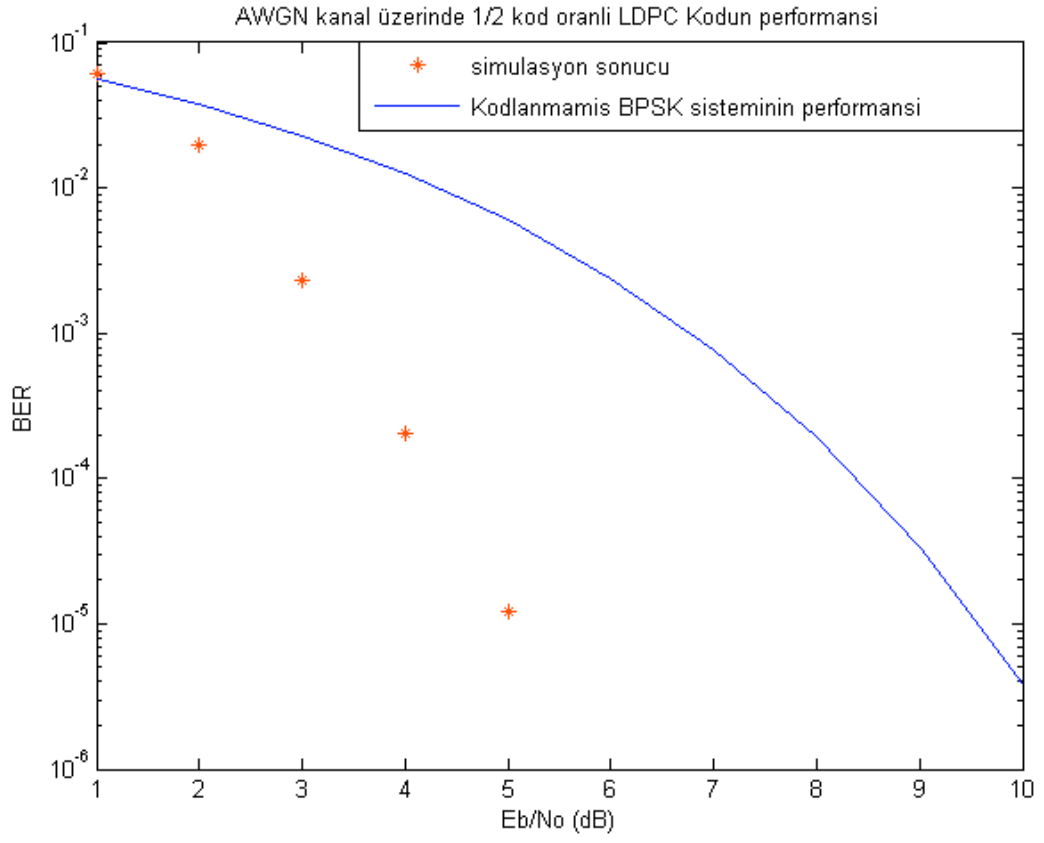
Bu simülasyonda H eşlik matrisi; 128×256 boyutunda, $w_c=3$ ve $w_r=6$ yapıdadır⁽¹⁸⁾.



Şekil 3.4: H eşlik kontrol matrisi

Bu simülasyonda kodlayıcı tarafından kodlanan mesaj işareti BPSK modülasyonu ile modüle edilir. Daha sonra AWGN kanal üzerinden iletilir. Alıcı kısmında ise, kanaldan alınan işarete demodülasyon işlemi uygulanır ve toplam-çarpım algoritması kullanılarak kodu çözülür.

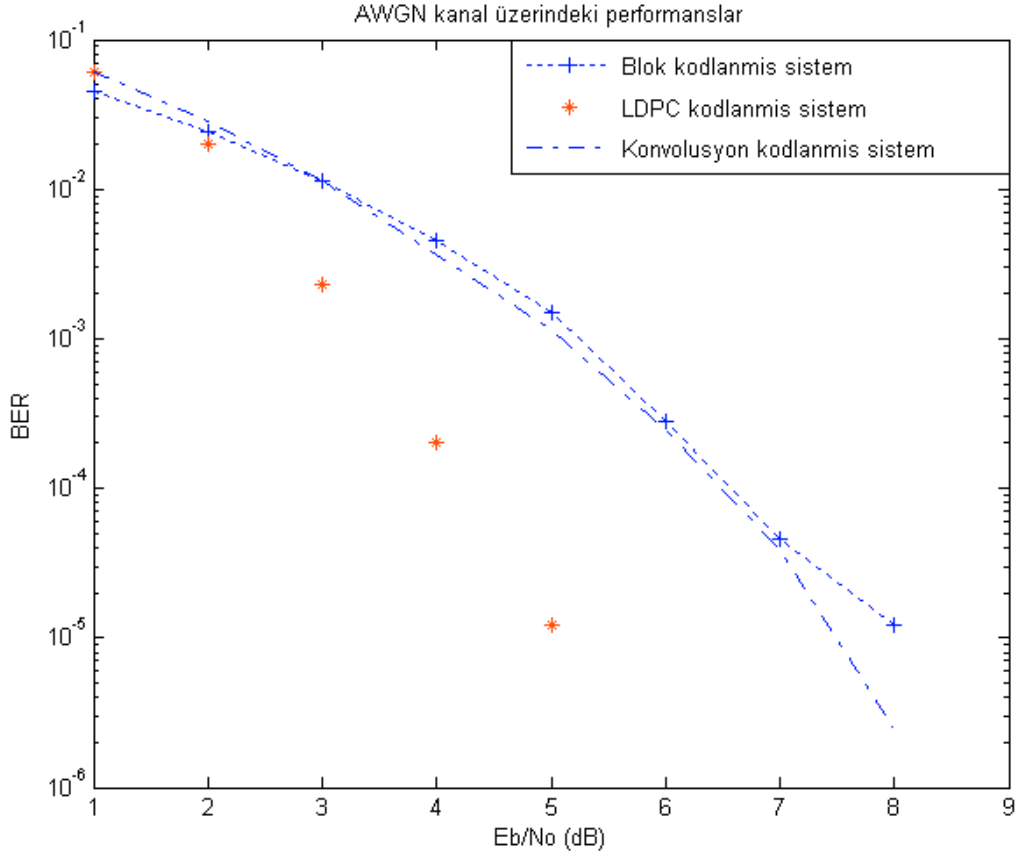
Karşılaştırma açısından kodlanmamış BPSK sistemi ve bu kodun simülasyon sonucu bulunmuştur.



Şekil 3.5: 1/2 oranlı LDPC kodun AWGN kanal üzerindeki karşılaştırılmalı performansı

4. SONUÇ

LDPC kodları ile blok ve konvolüsyon kodları kullanan BPSK sistemlerinin AWGN kanal üzerindeki bit hata olasılığı (BER) karşılaştırılması;



Şekil 4.1. Karşılaştırmalı simülasyon sonuçları

şeklindedir. Simülasyon sonuçlarına göre, LDPC kodu kullanan sistem blok ve konvolüsyon kodu kullanan sistemlere göre daha üstün bir performans özelliği göstermiştir. LDPC kodu kullanan sistemin blok kodu kullanan sistemlere göre yaklaşık 3 dB, konvolüsyon kodu kullanan sistemlere göre ise yaklaşık 2,5 dB'lik bir performans kazancı elde edilmiştir.

KAYNAKLAR

1. C. E. Shannon, A Mathematical Theory of Communication, Bell Syst. Tech. J., vol. 27, pp. 379-423, 623-656, 1948.
2. E. Tuna, Kafes Kodlamalı Modülasyon Kullanan Karma ARQ Sistemlerinin Toplanabilir Beyaz Gauss Gürültülü Ve Sönümlü Kanallar İçin Başarım Analizi. Doktora Tezi. Erciyes Üniversitesi, Kayseri, 2000.
3. R. G. Gallager, Low-Density Parity-Check Codes, IRE Trans. Inform. Theory, vol. IT-8, no. 1, pp. 21-28, 1962.
4. R. G. Gallager, Low-Density Parity-Check Codes, MIT Press, Cambridge, MA, 1963.
5. D. J. C. MacKay, Good Error-Correcting Codes Based on Very Sparse Matrices, IEEE Trans. Inform. Theory, vol. 45, no. 2, pp. 399-431, 1999.
6. R. M. Tanner, A Recursive Approach to Low Complexity Codes, IEEE Trans. Inform. Theory, vol. IT-27, no. 5, pp. 533-547, 1981.
7. S. Y. Chung, On the Construction of Some Capacity-Approaching Coding Scheme, PhD thesis, Massachusetts Institute of Technology, 2000.
8. T. J. Richardson and R. L. Urbanke, The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding, IEEE Trans. on Information Theory, vol. 47, pp. 599-618, 2001.
9. D. J. C. MacKay and R. M. Neal, Near Shannon Limit Performance of Low Density Parity Check Codes, Electron. Lett., vol. 32, no. 18, pp. 1645-1646, 1996, Reprinted Electron. Lett, vol. 33, no. 6, pp. 457-458, 1997.
10. T. Etzion, A. Trachtenberg, and A. Vardy, Which Codes Have Cycle-Free Tanner

- Graphs?, IEEE Trans. Inform. Theory, vol. 45, no. 6, pp. 2173-2181, 1999.
11. C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, Finitelength Analysis of Low-Density Parity-Check Codes on The Binary Erasure Channel, IEEE Trans. Inform. Theory, vol. 48, no. 6, pp. 1570-1579, 2002.
 12. M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, Efficient Erasure Correcting Codes, IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 569-584, 2001.
 13. T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, Design of Capacity Approaching Irregular Low-Density Parity-Check Codes, IEEE Trans. on Information Theory, Vol. 47, pp. 619–637, 2001.
 14. M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, Improved Low-Density Parity-Check Codes Using Irregular Graphs, IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 585-598, 2001.
 15. S.-Y. Chung, G. D. Forney, Jr., T. J. Richardson, and R. Urbanke, on the Design of Low-Density Parity-Check Codes Within 0.0045 dB of the Shannon Limit, IEEE Commun. Letters, vol. 5, no. 2, pp. 58-60, 2001.
 16. R. Lucas, M. P. C. Fossorier, Y. Kou, and S. Lin, Iterative Decoding of Onestep Majority Logic Decodable Codes Based on Belief Propagation, IEEE Trans. Commun., vol. 48, no. 6, pp. 931-937, 2000.
 17. Mei Yen, Cheong, Block Error Correction Codes and Convolution Codes, S-72.333 Postgraduate Course in Radio Communications - Wireless Local Area Network (WLAN)
 18. Shaikh Faisal Zaheer, faisal.zaheer@gmail.com, Non-Systematic LDPC Simulation
 19. G. D. Forney, Jr., The Viterbi Algorithm, Proc. IEEE, 61, pp. 268-278, 1973.

20. A. J. Viterbi and J. K. Omurga, Principles of Digital Communication and Coding, McGraw-Hill, New York, 1979.
21. Sarah J. Johnson, Steven R. Weller, Low-Density Parity-Check Codes: Design and Decoding, Technical Report EE02041, 2002.
22. William E. Ryan, An Introduction to LDPC Codes, 2003.
23. Stephan ten Brink, Gerhard Kramer, and Alexei Ashikhmin, Design of Low-Density Parity-Check Codes for Modulation and Detection, IEEE Trans. Commun., vol. 52, no. 4, pp. 670-678, 2004.
24. Pascal O. Vontobel and Ralf Koetter, Graph-Cover Decoding and Finite-Length Analysis of Message-Passing Iterative Decoding of LDPC Codes, IEEE Trans. Inform. Theory, submitted 2005.
25. Jose M. F. Moura, Jin Lu and Haotian Zhang, Structured Low-Density Parity-Check Codes, IEEE Signal Processing Mag. vol. 21, no. 1, pp. 42-55, 2004