KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK - ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

REED – SOLOMON KODLARIN AWGN ve RAYLEIGH KANALLARDA BAŞARIM ANALİZİ

YÜKSEK LİSANS TEZİ

Elektrik - Elektronik Mühendisi Yusuf ZORLU

TEMMUZ 2006
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

REED – SOLOMON KODLARIN AWGN ve RAYLEIGH KANALLARDA BAŞARIM ANALİZİ

Elektrik - Elektronik Mühendisi Yusuf ZORLU

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce
"Elektronik Yüksek Mühendisi"
Ünvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 09.06.2006
Tezin Savunma Tarihi : 06.07.2006

Tez Danışmanı : Yrd. Doç. Dr. İsmail KAYA
Jüri Üyesi : Doç. Dr. Temel KAYIKÇIOĞLU
Jüri Üyesi : Yrd. Doç. Dr. Cemal KÖSE

Enstitü Müdürü: Prof. Dr. Emin Zeki BAŞKENT

Trabzon 2006

ÖNSÖZ

Bu tez, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elektronik Mühendisliği Yüksek Lisans Programı'nda yapılan bir çalışmadır. "Reed – Solomon Kodların AWGN ve Rayleigh Kanallarda Başarım Analizi" konulu bu çalışmada, toplanır beyaz Gauss gürültülü (AWGN) kanal ve dar bantlı Rayleigh kanal baz alınarak önemli bir kod çözme tekniği olan Reed – Solomon Kod çözme detaylı bir şekilde incelenmiş ve performansı etkileyen faktörler belirlenmiştir. Ayrıca bir benzetim programı yazılarak Reed – Solomon kod çözücünün performansını etkileyen faktörler değiştirilmek suretiyle farklı benzetim sonuçları elde edilmiştir.

Yüksek lisans tezi danışmanlığımı üstlenerek, gerek konu seçimi ve gerekse çalışmalarımın yürütülmesi sırasında yardımlarını esirgemeyen değerli hocam Yrd. Doç. Dr. İsmail KAYA'ya, yerinde yardımları için KTÜ Elektrik-Elektronik Mühendisliği Bölümü öğretim üyeleri ve personeline, verdikleri teknik destek için KTÜ-DSP Laboratuarı çalışanlarına teşekkür ederim.

Ayrıca, uzun süre çalışmalarımızı beraber yürüttüğümüz değerli arkadaşım Bilgisayar Mühendisi Tuba ÖZLÜ'ye teşekkürü bir borç bilirim.

Son olarak benim için her türlü fedakârlığı yapan ve bana sabır gösteren aileme gösterdiği destek ve anlayış için çok teşekkür ederim.

Yusuf ZORLU

Trabzon,2006

İÇİNDEKİLER

Sayfa No

ÖZET

Sayısal iletişim sistemlerinde temel amaçlardan biri, birim zamanda alıcıya hatasız olarak aktarılan veri oranını yüksek tutmaktır. Bilgi, örneğin radyo veya uydu bağlantıları gibi ortamlarda iletilirken meydana gelen anlık değişimler bilgi işaretini bozarak veri kaybına neden olur. Bu anlamda bilginin bozulmadan, yüksek hızda iletilmesini veya saklanmasını sağlamak büyük önem taşımaktadır. Hemen her alanda, yüksek miktarda bilginin güvenilir bir şekilde iletilmesi, sunulan hizmet kalitesini artırır. Bilginin bozulmadan saklanması bilgi ve zaman kaybını önler.

Hata düzeltme kodları, iletim sırasında meydana gelebilecek sembol hatalarının algılanması ve düzeltilmesi için bilgi bitleri dizisine ek bit ilave etmektedir. Hata düzeltme kodlarının ortaya çıkması, gürültülü bir kanal üzerinden bilginin iletimi sırasında, Shannon' un kanal kapasitesine ulaşılabileceğini göstermiştir Reed – Solomon (RS) kodları, doğrusal blok kodlar ailesinin en güçlü kodlarıdır ve en geniş ölçüde kullanılan hata kontrol tipi olduğu tartışmalıdır. Özel olarak RS kodları ikili olmayan sistematik döngüsel doğrusal blok kodlardır. İkili olmayan bu kodlar birçok bitten oluşan semboller ile çalışırlar. İkili olmayan kodlar -öyle ki RS kodları – hata patlamalarını düzeltmede oldukça iyidirler. Çünkü bu kodların hata düzeltimi sembol seviyesinde yapılmaktadır. Ayrıca kodlama işlemi kod çözme işlemine göre oldukça basittir.

Bu çalışmada amaç, Reed – Solomon kodlar hakkında temel bilgileri vermek, kodların toplanır beyaz Gauss gürültülü (AWGN) kanaldaki ve dar bantlı Rayleigh kanaldaki performansını incelemek ve bunun sonucunda, AWGN ve Rayleigh kanallarda Reed – Solomon kod çözücünün performansını etkileyen faktörleri belirlemektir. Bunun için önce kodlama ve kod çözme yapıları incelenmiş ve Reed – Solomon kod çözme algoritması hakkında detaylı bilgi verilmiştir. Ardından, AWGN ve Rayleigh kanallarda Reed – Solomon kod çözücünün başarımını etkileyen faktörler değiştirilerek farklı benzetim sonuçları elde edilmiştir.

Anahtar Kelimeler: Reed - Solomon Kodlama, Hata Düzelten Kodlar, Reed Solomon Kod Çözme Algoritması

SUMMARY

Reed Solomon Codes Performance Analysis in AWGN and Rayleigh Channels

One of the main goals in a digital communication system is to increase the information rate at the receiver side without having erroneous data. Unexpected changes (e.g., multipath propagation effect), in the transmission mediums like radio or space links, corrupt the data signal and cause an information loss. Reliable data transfer at high speed increases the quality of service and is also an important problem in many cases. Storing the data without corrupting it prevents the data and time loss.

Error correction codes are a means of including redundancy in a stream of information bits to allow the detection and correction of symbol errors during transmission. The birth of error correction coding showed that Shannon's channel capacity could be achieved when transmitting information through a noisy channel. Reed – Solomon Codes (RS) are the most powerful in the family of linear block codes and are arguably the most widely used type of error control codes. To be specific, RS codes are non-binary systematic cyclic linear block codes. Non-binary codes work with symbols that consist of several bits. Non-binary codes such as RS are good at correcting burst errors because the correction of error the codes is done on the symbol level. Compared to decoding, channel encoding is relatively simple.

The dissertation specifically describes the structure of Reed – Solomon Codes gives the performance of Reed – Solomon Codes in additive white Gaussian noise (AWGN) channels and in Rayleigh fading channels and thus, determines the factors that influence the performance of Reed – Solomon decoding in AWGN and Rayleigh fading channels. Therefore, initially, encoding and decoding process is studied carefully and great amount of information about the Reed – Solomon decoding algorithm is given. Then, by changing the factors that influence the performance of Reed – Solomon decoder in AWGN and Rayleigh fading channels, various simulation results are obtained.

Key Words: Reed – Solomon Encoding, Error Correction Codes, Reed – Solomon Decoding Algorithm

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

SEMBOLLER DİZİNİ

ACK	Pozitif onay
A/D	Analog-sayısal dönüşüm
ADC	Analog-sayısal dönüştürücü
ARQ	Otomatik tekrar isteği
AWGN	Toplanır beyaz Gauss gürültüsü
BCH	Bose – Chaudri - Hocquenghem
BER	Bit hata oranı
BPSK	İkili faz kaydırmalı anahtarlama
BSC	İkili simetrik kanal
$b(X)$	Parite kontrol dizisinin polinomsal ifadesi
C	Kanal kapasitesi
CDPD	Hücresel sayısal paket veri
D_i	Katsayı matrisinin determinantı
D/A	Sayısal-analog dönüşüm
DMC	Ayrık hafızasız kanal
d_{min}	Minimum Hamming mesafesi
E_b/N_0	Bit başına düşen enerjinin gürültü gücüne oranı
E_s	Sembol başına düşen enerji
ESA	Avrupa uzay ajansı
e	Hata dizisi
e_i	i. hata değeri
e(X)	Hata dizisinin polinomsal ifadesi
FEC	İleri yön hata düzeltme
G	Üreteç matrisi
$GF(l)$	Galois alanı
$GF(p)$	p elemanlı Galois alanı
GSM	Küresel gezgin haberleşme sistemi
g(X)	Üreteç polinomu

H^T	Parity kontrol matrisinin transpozesi
I_k	Birim matris
$i(X)$	Bilgi dizisinin polinomsal ifadesi
K	Sınır uzunluğu
k	Kodlayıcıya giren veri sayısı
L	Giriş dizisinin uzunluğu (blok uzunluğu)
m	Kodlayıcıdaki, en uzun bellek dizisindeki, bellek elemanı sayısı
n	Kodlayıcıdan çıkan veri sayısı
N_0	Tek taraflı gürültü gücü spektrumu
NAK	Negatif onay
NASA	Amerikan uzay ve havacılık ajansı
QPSK	Quadrature faz kaydırmalı anahtarlama
P_e	Bit hata olasılığı
$p(x)$	Gauss gürültüsü olasılık dağılım fonksiyonu
q	Blok kodun derecesi
R	Kod oranı (Kod hızı)
RS	Reed - Solomon
r	Alınan kod vektörü
r(X)	Alınan kod dizisinin polinomsal ifadesi
S	Hata belirteci
S_i	i. hata belirteci
$S(X)$	Hata belirteçlerinin polinomsal ifadesi
SER	Sembol hata oranı
S_r	Serpiştirici faktörü
SNR	İşaret/gürültü oranı
t	Düzeltilebilecek hata sayısı
u	Bilgi dizisi vektörü
v	Kodlanmış dizi vektörü
v(X)	Kod dizisinin polinomsal ifadesi
X_t	Kanalın girişi
Y_t	Kanalın çıkışı

Z_t	Sıfır ortalamalı Gauss gürültüsü
δ	Gürültü varyansı
β_i	i. hata yeri
A_t	Kanal kazancı
α	İlkel eleman
$\sigma(X)$	Hata yeri polinomu
$C(X)$	Hata yeri polinomunun tersi
$\sigma'(X)$	Hata yeri polinomunun X' e göre türevi
μ	Adım sayısı
d_μ	μ . Uzaklık
l_μ	μ . Adımda elde edilen polinomun derecesi
$\Phi(X)$	$GF(2^m)$ için tanımlı minimal polinom
$\omega(X)$	Hata değerlendirme polinomu

1. GENEL BİLGİLER

1.1. Giriş

Son birkaç yılda, etkin, kaliteli ve güvenilir olması nedeniyle sayısal data iletimine olan talep giderek artmaktadır. Bu talep, askeri, resmi ve özel çalışma alanlarında, sayısal bilginin, yüksek hızlı data ağlarında değişimi, işlenmesi ve saklanması gerekliliğinin ortaya çıkması ile hızla artmıştır. Bu sistemlerin tasarımı için iletişim ve bilgisayar teknolojilerinin birleştirilmesi gerekmiştir. Dolayısıyla hata kontrolü ve data güvenliği, günümüz haberleşme araçlarının en önemli birkaç öğesinden biri haline gelmiştir.

Temel olarak bir iletişim sisteminde amaç, bilgiyi bir yerden başka bir yere iletmektir. Bu yapılırken, birim zamanda alıcıya hatasız olarak aktarılacak veri oranını yüksek tutmak gerekir. Bilgi; radyo bağlantıları, uydu bağlantıları, telefon hatları gibi ortamlarda iletilir veya manyetik bant, Compact Disk (CD) gibi ortamlarda saklanır. Bir iletim sisteminin performansı, iletim sırasında meydana gelen bilgi kaybı miktarı ile ölçülür.

İletim sırasında oluşan hatalar kanalın bozucu etkilerinden kaynaklandığından, haberleşme sistemi tasarlanırken kanalın gürültü özellikleri belirlenip uygun şekilde modellenmelidir. Seçilen kanal modeline uygun olarak doğru iletimi sağlamak için hata başarımını artıracak önlemler alınmalıdır. Kanal kodlaması yapılarak belirli bir hata olasılığıyla gürültülü bir kanaldan bilgi iletimi mümkündür. Bu konuyla ilgili olarak ilk çalışma 1948 yılında Shannon tarafından yapılmıştır. Shannon, gürültü seviyesi, işaret gücü ve bant genişliği gibi kanalın karakteristik özelliklerinden yola çıkarak kanal kapasitesinin hesaplanabileceğini ispatlamıştır. Kanal kapasitesi, birim zamanda kanaldan güvenilir bir şekilde iletilebilecek maksimum veri miktarıdır. Shannon, veri iletim hızı kanal kapasitesinin altında olduğu sürece bilginin uygun biçimde kodlanmasıyla, data iletim hızından taviz vermeden, gürültülü kanallarda ortaya çıkabilecek hata olasılığının azaltılabileceğini göstermiştir. Shannon'un çalışmasından bu güne, gürültülü ortamlarda etkin kodlama ve kod çözme teknikleri ile hata kontrolü için büyük çaba harcanmıştır.

Son yıllardaki gelişmeler, bugünün yüksek hızlı sayısal sistemlerinin gereksinim duyduğu güvenilirliğe yaklaşmaya yardımcı olmuş ve hata kontrol kodlaması modern iletişim sistem tasarımı içindeki yerini almıştır.

Sayısal iletişim sistemi tasarımcısının görevi, vericiden alıcıya, kullanıcıların istediği hızda ve doğrulukta veri iletimini sağlamaktır. Bu tip bir sistemin tasarım parametreleri ise iletim bant genişliği, işaret gücü ve seçilen gerçekleme tipinin karmaşıklığıdır. Veri iletim hızı ve doğruluğu ise uygulamaya göre değişir.

İletim bant genişliği iletim ortamının özellikleri ile sınırlanır. Örneğin yeryüzünde, başka kullanıcılarla etkileşimi engellemekte her bir haberleşme kanalı için belirli bir bant genişliği ayrılmıştır. İletim bant genişliğinin sınırlanmadığı uygulamalar olarak uzay araçlarıyla kurulan bağlantılar örnek verilebilir. Başka bir kullanıcıyla etkileşim olasılığı çok düşük olduğundan, bu tip iletimde çok geniş bant genişlikleri kullanılabilir. İşaret gücü ve gerçekleme karmaşıklığı ise daha çok tasarımcının kontrolünde olan parametrelerdir. Gerçeklenebilecek birkaç sistemden maliyet ve başarım açısından en uygun olanını seçmek bir mühendislik sorunudur. Örneğin bazı sistemlerde istenen doğrulukta veri iletimi, gönderilen işaretin gücünü arttırarak sağlanabilir. İşaret gücü, kanalda var olan gürültü gücünden yeterince yüksek ise, gönderilen işaret kanal içinde bozulmadan iletilir.

Güvenilir bir haberleşme bağlantısında gerekli yüksek işaret gürültü oranını elde etmek için daha yüksek verici gücü kullanmak yerine günümüzde hata düzeltme teknikleri tercih edilmektedir. Çünkü standartların en önemli sınırlaması güç üzerinde olmaktadır. Hata düzeltme teknikleri ile gönderilen işarete, sistematik olarak bazı kontrol işaretleri eklenir. Tıbbi ve ticari alanlarda kablosuz haberleşme ağları, her yeni üründe elektromagnetik güç emisyonunu önemli ölçüde azaltmaktadır.

Shannon'un çalışmasını takiben, R.W. Hamming, A. Hocquenghem, R.C. Bose ve D.K. Ray-Chaudri, I.S. Reed ve G. Solomon çalışmalarıyla gelişen hata düzelten kodlar ve kod çözme algoritmaları sağlam matematiksel temeller üzerine inşa edilmiştir. Hata düzeltme tekniklerinde karşılaşılan güçlük kod çözme algoritmalarının karmaşıklığından kaynaklanmaktadır. Kod çözme işlemi birçok karmaşık bağıntının beraberce çözülmesini gerektirdiğinden hızlı bir şekilde kod çözme işlemini gerçekleştirmekte güçlükler ortaya çıkabilir.

1.2. Sayısal Haberleşme Sistemleri

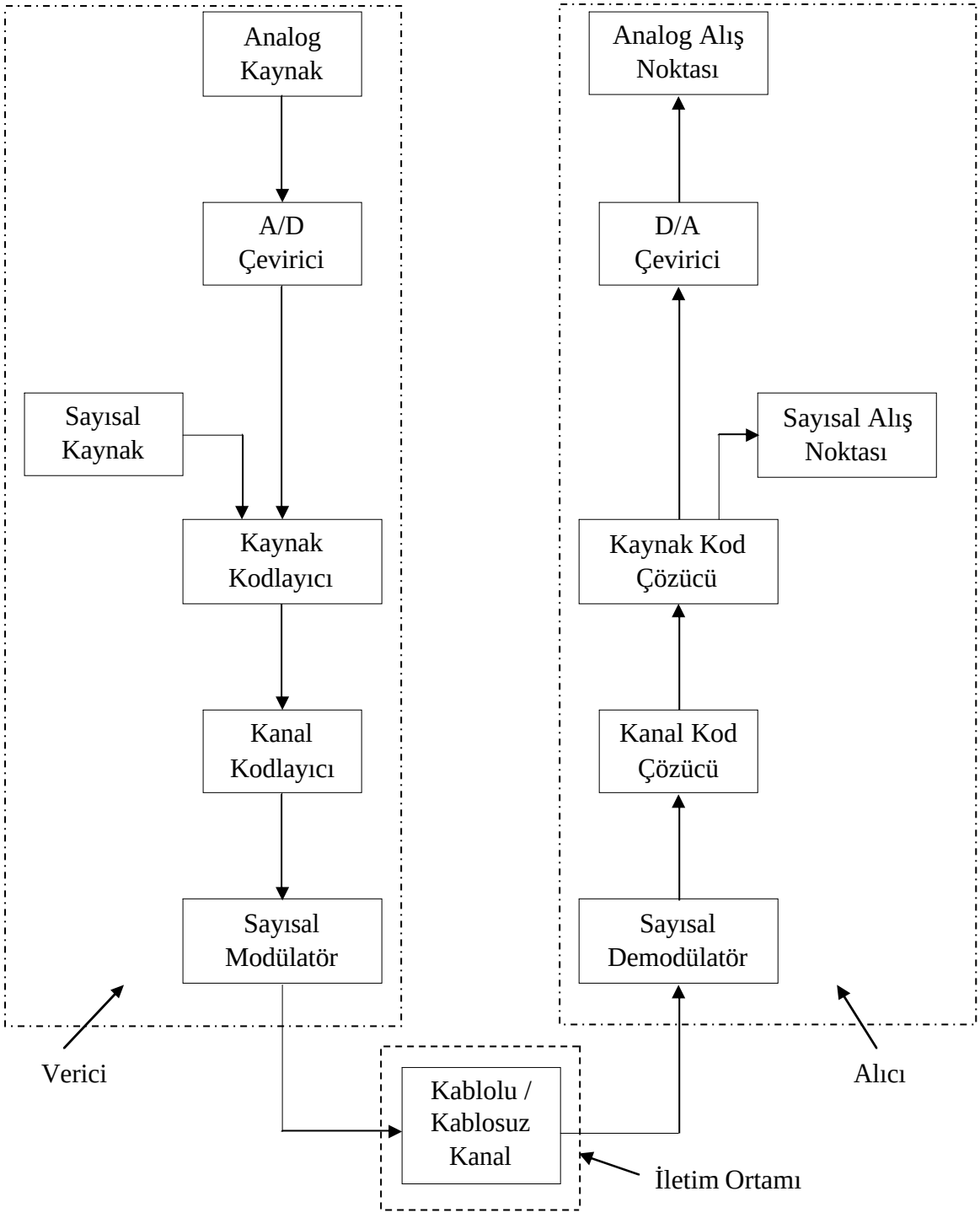
Günümüzde, hızla artan teknolojik gelişmelere paralel olarak bilgi trafiği de hızla artmıştır. ATM, ADSL, ISDN gibi yüksek hızlarda bilgi iletimine olanak veren çözümler sayesinde, yüzlerce sayfalık bir dökümanın dünyanın bir ucundan başka bir ucuna ulaştırılması saniyenin kesirleri mertebesinde sürmektedir. Hemen her alanda, yüksek miktarda bilginin güvenilir bir şekilde iletilmesi, sunulan hizmet kalitesini artırır. Bilginin bozulmadan saklanması bilgi ve zaman kaybını önler. Modern Sayısal İletişim Sistemlerinde bilginin bozulmadan, yüksek hızda iletilmesini ve saklanmasını sağlamak büyük önem taşımaktadır. Örneğin bankalar arası bir bilgi ağında hatalı iletime izin verilemez.

İletişimde, data bir bilgi kaynağından hedefe doğru gürültülü bir iletim ortamından transfer edilir. Tipik bir iletişim sistemi Şekil 1'deki blok diyagramdaki gibi ifade edilebilir. Bu sistem üç ana bölümden oluşmaktadır: bir verici (kaynak), bir alıcı (varış yeri) ve bir iletim ortamı (bir çift tel, koaksiyel kablo, fiber hat ya da boş alan)[1].Bilgi kaynağı insan veya bir makine olabilir.

Tipik bir sayısal iletişim sisteminde başlangıçtaki kaynak bilgi analog ya da sayısal darbeler halinde olabilir. Analog bilgiye örnek olarak ses, video, resim bilgisi ya da müzik; sayısal darbelere örnek olarak ise ikili kodlu sayılar, alfa sayısal kodlar, grafik semboller, mikroişlemci işlem kodları ya da veri tabanı bilgisi verilebilir. Sayısal iletişim sisteminde, eğer kaynak bilgi analog biçimde ise iletilmeden önce sayısal darbelere dönüştürülmelidir. Bu nedenle analog bilgi, bir analog-sayısal (A/D) çevirici yardımıyla sayısal darbelere dönüştürülmektedir.

Etkin bir sayısal iletişim sisteminin sağlanmasının en iyi yolu, veri sembollerinde fazlalıkların atılmasıdır. Yani kaynak bilgisini mümkün olabildiğince sıkıştırabilmektir. İşte bu nedenle, gönderilecek olan veri miktarını sıkıştırmak için kaynak kodlayıcı kullanılmaktadır. Örneğin, bir video sinyalinin A/D çeviriciden geçirilerek MPEG kaynak kodlayıcıya sürülmesiyle verinin sıkıştırılması sağlanmaktadır.

Dikkat edilirse, kaynak kodlama işlemi ile kanal kodlama işlemi arasında bir zıtlık olduğu görülmektedir. Kaynak kodlama işlemi ile kaynak bilgi sıkıştırılarak küçültülürken, kanal kodlama işlemi ile kaynak bilgiye fazlalık (artık bitleri) eklenmektedir. Sonuç olarak, kaynak kodlama işlemi kanal kodlama işleminin tam tersi bir işlem olarak düşünülebilmektedir.



Şekil 1. Sayısal iletişim sisteminin blok diyagramı

Her ne kadar kanal kodlama işlemi ile bilgi dizisine fazlalık katılmış olsa da, bu fazlalık orijinal bilgideki asıl fazlalıktan daha mantıksal ve kontrol edilebilir olduğundan, alıcı tarafında hataları düzeltmek için etkin bir şekilde kullanılabilmektedir.

Bilgi dizisinin fiziksel kanal üzerinden iletime uygun olmamasından dolayı sayısal bir modülatör yardımı ile kanal kodlayıcı çıkışındaki veri modüle edilip sürekli zaman dalga şekline çevrildikten sonra fiziksel kanal üzerinden iletilmektedir. Şekil 1'de görüldüğü gibi genellikle iki tür fiziksel kanal kullanılmaktadır: kablolu ve kablosuz. Kablolu kanala örnek olarak koaksiyel kablo, optik fiber, vb., kablosuz kanala örnek olarak ise serbest uzaydaki elektromagnetik kanallar verilebilir. Yalnız, bilgi sayısal kanal üzerinden iletilirken bir takım bozucu etkenden dolayı bilgi kaybı meydana gelmektedir.

Alıcının girişinde ise bozulmuş işaret alınmaktadır. Alınan bu işaret sayısal detektör yardımı ile demodüle edilerek kodlanmış bitlerin tahmini yapılmaktadır. Kanal kod çözücü, verici tarafındaki kanal kodlayıcının bilgi bitlerine eklediği artık bitleri kullanarak, mümkün olduğunca hataların algılanıp düzeltilmesini sağlamaktadır. Böylelikle, bilgi bitlerine verici tarafında kanal kodlayıcı tarafından eklenen fazlalık bu aşamadan sonra ortadan kalkmaktadır.

1.3. Hata Kontrol Yöntemleri

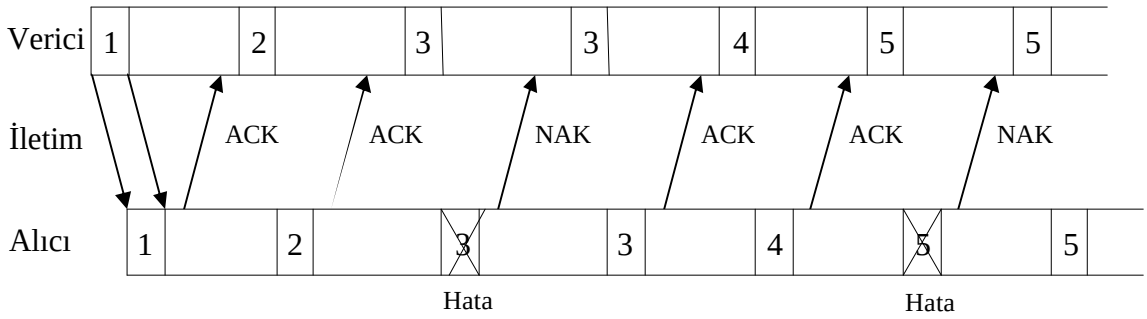
Hata kontrolünde temel olarak iki yöntemden söz edilebilir. Bunlardan biri, "otomatik tekrar isteği" denilen ARQ yöntemi, diğeri ise "ileri yön hata düzeltme tekniği" FEC'dir.

Şekil 1'deki blok diyagram, tek yönlü bir iletişim sistemini göstermektedir. Tek yönlü iletişim sisteminde hata kontrolü, ileri yön hata düzeltme tekniği kullanılarak yapılmalıdır. Bu teknikte bilgi mesajına hata düzeltme kodları ilave edilerek, alıcıda algılanan hatalı bitler otomatik olarak düzeltilir.

Bazı iletişim sistemleri iki yönlü olabilir. Bilgi iki yönde de gönderilir ve verici aynı zamanda bir alıcı gibi de davranır. İki yönlü iletişim sistemleri için hata kontrolü, "otomatik tekrar isteği" (ARQ) adı verilen, hatayı algılayan ve bilgiyi yeniden gönderen bir sistemle yapılabilir. ARQ sisteminde, alıcıda hata algılandığında, vericiye mesajı tekrarlaması için bir istek gönderilir. Bu istek, mesaj doğru alınana kadar devam eder.

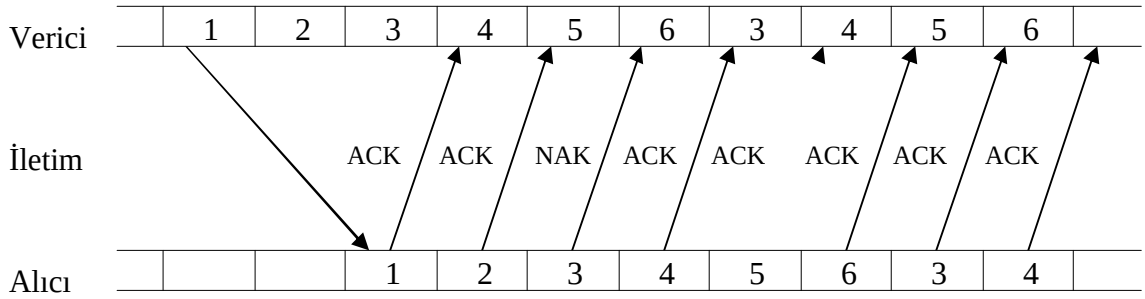
İki tip ARQ sistemi vardır.

- Dur ve bekle ARQ
- Sürekli ARQ



Şekil 2. Dur ve bekle ARQ

Dur ve bekle ARQ'da, verici alıcıya bir kod gönderir ve alıcıdan pozitif (ACK) veya negatif (NAK) bir onay almak için bekler. Eğer NAK cevabı alınırsa hata algılanmıştır ve önceki kod kelimesi yeniden gönderilir. Aynı kod, doğru olarak alınana kadar birçok kez gönderilir.



Şekil 3. Sürekli ARQ

Sürekli ARQ'da verici, kod kelimesini alıcıya sürekli olarak gönderir ve sürekli olarak bilgilendirme mesajını alır. Bir NAK algılandığında, verici hatalı olan kod kelimesine döner ve bu kelimeyi onu takip eden kelimelerle birlikte göndermeyi sürdürür. Buna geriye dönüşlü-N (go-back-N) ARQ denir. Alternatif olarak, verici sadece negatif bilgi mesajı alınan kod kelimelerini tekrar gönderebilir. Buna seçimli-tekrarlamalı (selective-repeat) ARQ denir. Seçimli-tekrarlamalı ARQ, geriye dönüşlü-N ARQ'dan daha etkindir ancak daha fazla lojik ve tamponlama (buffering) ister.

Sürekli ARQ, Dur ve bekle ARQ'ya göre daha etkindir fakat daha pahalıdır. İletim hızının yüksek ve dönüş yolu gecikmesinin uzun olduğu uydu iletişim sistemlerinde sürekli ARQ kullanılır.

Dur ve bekle ARQ sistemleri, bir kod kelimesini göndermek için geçen sürenin, bilgi mesajını almak için geçen süreden daha uzun olduğu iletişim sistemlerinde kullanılır. Dur ve bekle ARQ sistemleri yarı dubleks kanallarda, sürekli ARQ ise tam dubleks kanallarda kullanılmak için tasarlanmıştır.

ARQ'nun FEC'e göre en önemli avantajı, hata algılamanın hata düzeltmeye göre çok basit kod çözme algoritmasına sahip olmasıdır. Ayrıca ARQ adaptiftir. Yani sadece hata meydana geldiğinde bilgi yeniden gönderilir. Diğer taraftan, kanal hata oranı yüksekse yeniden gönderme çok sıklıkla tekrarlanmalıdır. Bu da hızı düşürecektir. Bu durumda sık karşılaşılan hata paterni için FEC ve daha uygunsuz hata paterni için hata algılama ve yeniden göndermenin kombinasyonu, tek başına olan ARQ'den daha etkin olacaktır.

1.4. Hata Türleri

Hafızasız kanallarda, gürültünün iletilen sembollere etkisi birbirinden bağımsızdır. Örnek olarak, ikili simetrik kanalı (binary simetric channel, BSC) düşünelim. Her iletilen bit, diğer iletilen bitlerden bağımsız olarak, " p " kadar yanlış alınma olasılığına ve "$1-p$" kadar doğru alınma olasılığına sahiptir. Bu sebepten dolayı, alınan dizide iletim hataları rastgele oluşur ve hafızasız kanallara rastgele hata kanalları denir. Derin uzay kanalları ve birçok uydu kanalı rastgele hata kanalına örnektir. Direkt görüş iletişim sistemlerinin çoğu, asıl olarak rastgele hatadan etkilenir. Rastgele hataları düzeltmek için tasarlana kodlara rastgele hata düzeltme kodları denir.

Hafızalı kanallarda gürültü, iletimden iletime bağımsız değildir. Bu model iki durum içerir. Hatanın nadiren ortaya çıktığı iyi durumda $p_1 \approx 0$ ve hata olasılığının yüksek olduğu kötü durumda $p_2 \approx 0$.'dir. Kanal genelde iyi durumdadır. Ancak ara sıra kanalın iletim karakteristiğindeki değişimlerden dolayı, örneğin çok yollu iletimden kaynaklanan derin bayılmalar, kanal kötü duruma geçebilir. Sonuç olarak iletim hataları ardışıl meydana gelir. Hafızalı kanallara, hata patlamalı kanallar denir. Hata patlamasını gidermek için tasarlanan kodlara, hata patlaması düzeltme kodları denir.

1.5. Hata Düzeltme Kodlaması

Herhangi bir iletişim kanalı üzerinden bilgi iletilmeye çalışılırken, esas amaçlardan biri güvenilirliği sağlamaktır ve bu alıcıdaki doğru algılama oranı ile ölçülmektedir[2]. Bu amaçlar doğrultusunda, iletişim sistemlerinde güvenilirliği sağlamak ve alıcıdaki doğru algılama oranını artırmak için hata düzeltme kodlamasından yararlanılmaktadır. Bilgi teorisinin ve hata düzeltme kodlamasının temelini, Shannon'un 1940'lı yıllarda ortaya koyduğu "kanal kapasite teoremi" oluşturmaktadır. Bu teoremde, kod oranına bağlı olarak gürültülü bir ortamda kanal kapasitesinin teorik olarak ulaşabileceği üst sınır belirtilmiştir. Bunun doğrultusunda, kodlama teorisine olan ilgi hızla artmaya başlamıştır. Günümüzde pratik olarak Shannon'un kanal kapasite sınırına ulaşabilmek için etkin kanal kodlama tekniklerini içinde barındıran sayısal iletişim sistemlerinin tasarımı üzerinde durulmaktadır.

1.5.1. Kod Oranı (Kod Hızı)

Kanal kodlayıcılarının performansını belirleyen en önemli kavramlardan biri kod oranıdır. Bir kanal kodlayıcısı, k bitlik veri dizisini n bitlik kod kelimesine dönüştürmekte ($n > k$) ve buna bağlı olarak $R = k/n$ oranı kod oranı olarak adlandırılmaktadır. $n > k$ olmasından dolayı kod oranı birden küçüktür ($R < 1$). Ayrıca, güvenilir bir haberleşme sağlamak için kod oranı kanal kapasitesini aşmamalıdır ($R \leq C$). Sıkça kullanılan kod oranları $1/4$, $1/3$, $1/2$, $2/3$ ve $3/4$'tür. Fakat pratikte, teorikten farklı olarak kod oranı, ek kodlayıcı bitleri olarak nitelendirilen kuyruk bitlerinden ötürü yukarıda verilen kod oranı değerlerinden çok az da olsa küçüktür.

1.5.2. Shannon'un Kanal Kapasitesi Teoremi

Kod oranı, kanal kapasitesi olarak adlandırılan bir rakamdan daha küçük olduğu sürece, gürültülü kanallar üzerinden de güvenilir iletişim sağlamak mümkün olacaktır [3].1940'lı yıllarda Claude Shannon tarafından sunulan bu bulgu, gürültülü kanal kapasitesi teoremi [4] olarak bilinmektedir.

Bu teoreme göre, toplanır beyaz Gauss gürültüsünün (AWGN) getirdiği kısıtlama, iletişim güvenirliliği ile ilgili değil de iletişim hızı ile bağlantılıdır. Bu teorem, hatanın oluşmadığı yada yok olduğu bir kanal ile desteklenebilen maksimum data oranını verir. AWGN kanal için kapasite,

$$C = \frac{1}{2}\log_2\left(1+\frac{2E_s}{N_0}\right) \tag{1.1}$$

ile bulunur. Burada C bilgi kapasitesi, E_S sembol enerjisi, N_0'da kanalın çift taraflı gürültü güç yoğunluk spektrumudur. Bu kapasiteye yakın değerlere pratikte, hata düzeltme kodlamasıyla (*error correction coding*) ulaşılabilir. Güvenilir bir haberleşme sağlamak için kod oranı, kanal kapasitesini aşmamalıdır. ($R \le C$). Bu durumda gerekli minimum E_b/N_0 aşağıdaki denklemden bulunur.

$$\frac{E_b}{N_0} \ge \frac{1}{2R}\left(2^{2R}-1\right) \tag{1.2.}$$

1.2. denkleminde eşitlik, sadece girişin Gauss dağılımlı olması durumunda geçerlidir. Shannon'un teoreminde, $R \le C$ şartına bağlı olan bir rastgele kod için kod uzunluğu (n) sonsuza giderken, bit hata olasılığı da (P_e) sıfıra yaklaşır. Ancak pratikte rastgele kodlar gerçekleştirilemez. Ayrıca, kodlama ve kod çözme işlemlerinin verimli yapılabilmesi için kodların yapılandırılmış olması gerekir. Ancak Shannon, kanal kapasitesine pratik olarak nasıl ulaşılacağına açıklık getirmemiş, sadece elde edilebilir olduğunu göstermiştir.

1.6. Matematiksel Temeller

Bu bölümde hata algılayan ve düzelten kodların teorik temellerini oluşturan grup, alan ve Galois Alanı ile ilgili temel bilgiler verilmiştir. Bölümde geçen teoremlerin ispatları ve tanımlarla ilgili daha ayrıntılı bilgi çeşitli kaynaklarda bulunabilir[50 – 53].

1.6.1 Grup Tanımı

Hata düzelten kodların özelliklerini belirleyen en basit cebirsel sistem gruptur.

Bir küme ve o kümenin elemanları üzerine tanımlanmış bağıntı ve işlemlerin oluşturduğu kümeye “Cebirsel Sistem” denir; kümenin iki elemanına uygulanan işlem sonunda elde edilen eleman yine kümenin bir elemanıdır; dolayısıyla kapalılık otomatik olarak sağlanır.

Tanım 1: G bir elemanlar kümesi, a,b,c ε G olmak üzere, * , G üzerinde tanımlı bir matematiksel işlem olsun. * işlemi G kümesi içinde aşağıdaki özellikleri sağlıyorsa G gruptur denir.

i. G, * işlemine göre kapalıdır.

$$c = a * b$$

ii. * işleminin G içinde birleşme özelliği vardır.

$$a*(b*c) = (a*b)*c$$

iii. G’nin bir birim elemanı vardır ve e ile gösterilir.

$$a*e = e*a = a$$

iv. a, G’nin bir elemanı ise a’nın * işlemine göre tersi vardır ve $a^{'}$ ile gösterilir.

$$a * a^{'} = a^{'} * a = e$$

Bir grubun eleman sayısı o grubun derecesidir ve bir grup sonlu veya sonsuz sayıda elaman içerebilir. Örneğin *0,1,2 ……. M-1* tam sayılarının oluşturduğu grup mod M toplama işlemi altında sonlu bir gruptur. *1,2,….. M-1* tam sayılarının oluşturduğu grup, M asal sayı ise mod M çarpma altında sonlu bir gruptur. Tablo 1.’de M=3 için tanımlı çarpma ve bölme işlemi verilmiştir.

Tablo 1.a) Mod 3 için toplama ve çarpma

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

*	1	2
1	1	2
2	2	1

1.

6.2 Alan Tanımı

Alan adı verilen cebirsel sistemde, kümenin elemanları üzerine iki işlem tanımlanmıştır. Bunlara toplama ve çarpma işlemleri denilecektir. Alan, üzerinde toplama ve çarpma işlemlerinin tanımlı olduğu elemanlar kümesidir. Çıkarma ve bölme işlemleri, alan içindeki her elemanın toplama ve çarpma işlemlerine göre tersi olduğundan yola çıkılarak gerçeklenir. F alanın elemanları toplama ve çarpma işlemleriyle aşağıdaki koşulları sağlamalıdır.

i. Toplama işleminin sıra değişme özelliği vardır ve toplama işleminin birim elemanı 0'dır.

$a + b = b + a$

$a + 0 = 0 + a = a$

ii. Çarpma işleminin sıra değişme özelliği vardır ve çarpma işleminin birim elemanı 1'dir

$a . b = b . a$

$a . 1 = 1 . a = a$

iii. Çarpmanın toplama üzerine dağılma özelliği vardır.

$a . (b + c) = a .b + a . c$

Gerçel sayılar toplama ve çarpma işlemi altında bir alan oluştururlar. Tamsayılar kümesi 1 dışındaki elemanların çarpmaya göre tersini içermediğinden alan değildir. Alanın eleman sayısı sonlu veya sonsuz olabilir. Toplam *q* tane eleman içeren alana. sonlu alan veya Galois Alanı adı verilir ve *GF(q)* ile gösterilir, p bir asal sayı ve m bir tamsayı olmak üzere her zaman bir $GF(p^m)$ vardır. Sonlu alanların en basit örneği asal alan *GF(p)*'dir.

Burada *p*, 1'den büyük herhangi bir asal sayı, toplama ve çarpma işlemleri ise mod *p* toplama ve mod *p* çarpma işlemleri olarak tanımlıdır. Örnek olarak Tablo 2'de GF(7) üzerinde toplama ve çarpma işlemleri görülebilir.

Tablo 2. GF(7) üzerinde toplama ve çarpma işlemleri

+	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

*	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

1.6.3. Galois Alanı Üzerinde Matematiksel İşlemler

$GF(q=p^m)$ alanı içinde matematiksel işlemlerin tanımı bir önceki bölümde verilmişti, Hata düzelten kodların tasarımında ve gerçeklenmesinde *GF(q)* üzerinde oluşturulan polinomlar ve bu polinomlar arasındaki matematiksel işlemler büyük önem taşımaktadır. Bu bölümde katsayıları *GF(q)*'nun elemanı olan polinomlar asındaki matematiksel işlemlerin gerçekleştirilmesi ve *GF(q)*'nun oluşturulması açıklanacaktır.

Katsayıları GF(2)'nin elemanı olan polinomlar f(X) ve g(X) aşağıdaki biçimdedir.

$f(X)=f_0+f_1X+f_2X^2+...f_nX^n$

$g(X)=g_0+g_1X+g_2X^2+...g_mX^m$

Bu iki polinomun toplamı, m < n olmak üzere

$$f(X) + g(X) = (f_0 + g_0) + (f_1 + g_1)X + \ldots\ldots + (f_m + g_m)X^m + f_{m+1}X^{m+1} + \ldots.. f_nX^n \quad (1.3)$$

ifadesi ile belirlenir.

f(X) ve g(X) polinomlarının çarpımı ise aşağıdaki şekilde hesaplanabilir.

$f(X) \cdot g(X) = c_0+c_1X+c_2X^2+ \ldots\ldots c_{n+m}X^{n+m}$ (1.4)

$c_0 = f_0 \cdot g_0$

$c_1 = f_0 \cdot g_1 + f_1 \cdot g_0$

$c_2 = f_0 \cdot g_2 + f_1 \cdot g_1 + f_2 \cdot g_0$

.

.

$C_{n+m} = f_n \cdot g_m$

Örneğin $f(X) = 1+X$, $g(X) = 1+X^2$ için ,

$f(X) + g(X) = X+X^2$, $f(X) \cdot g(X) = 1+X+X^2$ olur.

Tanım 2: GF(2) üzerinde oluşturulan m. Dereceden p(X) polinomu m'den daha küçük dereceli polinomlara bölünemiyorsa p(X) GF(2) üzerinde azaltılamaz denir.

Tanım 3: GF(2) üzerinde azaltılamaz p(X) polinomunun böldüğü $X^n + 1$ polinomu için $n=2^m - 1$ ise p(X) ilkeldir denir.

Örneğin $p(X) = X^4 + X + 1$ polinomu 4'ten daha az dereceli bir polinoma bölünemez ve bu nedenle GF(2) üzerinde azaltılamaz polinomdur. p(X) , $X^{15} + 1$ i böler fakat $1=n=15$ olmak üzere başka bir X^n+1 polinomunu bölemez. Bu nedenle p(X) ilkel bir polinomdur.

İlkel polinomlar için Galois Alanlarının oluşturulmasında kullanılırlar. Galois alanının oluşturulması için GF(2) içinde α sembolü tanımlansın, öyle ki ,$\alpha .0= 0$,$\alpha .1=\alpha$ olsun. $p(\alpha) = 0$ olarak verilir[52]. Örnek olarak $p(X) = X^4 + X + 1$ polinomundan $GF(2^4)$ oluşturulsun.

$$p(\alpha)=\alpha^4+\alpha+1=0$$

$$\alpha^4=\alpha+1$$

$$\alpha^5=\alpha.\alpha^4=\alpha.(\alpha+1)=\alpha^2+\alpha$$

$$\alpha^6=\alpha.\alpha^5=\alpha(\alpha^2+\alpha)=\alpha^3+\alpha^2$$

$$\alpha^7=\alpha.\alpha^6=\alpha(\alpha^3+\alpha^2)=\alpha^4+\alpha^3=\alpha+1+\alpha^3=1+\alpha+\alpha^3$$

Bu işlemler $GF(2^4)$'ün tüm elemanları için tekrarlanırsa, her elemana bir polinom karşı düşer. p_i, bir polinomun i. katsayısı olmak üzere, her eleman 4 bitlik $p_0\ p_1\ p_2\ p_3$ vektörüyle de ifade edilebilir. Tablo 3 te $GF(2^4)$'ün tüm elamanlarının polinomsal ve vektörel gösterimleri verilmiştir.

Tablo 3. GF(2^4)'ün elemanlarının farklı gösterilişleri

Eleman	Polinomsal Gösterim	Vektörel Gösterim	Ondalık Gösterim
0	0	0000	0
1	1	0001	1
α	α	0010	2
α^2	α^2	0100	4
α^3	α^3	1000	8
α^4	$\alpha + 1$	0011	3
α^5	$\alpha^2 + \alpha$	0110	6
α^6	$\alpha^3 + \alpha^2$	1100	12
α^7	$\alpha^3 + \alpha + 1$	1011	11
α^8	$\alpha^2 + 1$	0101	5
α^9	$\alpha^3 + \alpha$	1010	10
α^{10}	$\alpha^2 + \alpha + 1$	0111	7
α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1110	14
α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1111	15
α^{13}	$\alpha^3 + \alpha^2 + 1$	1101	13
α^{14}	$\alpha^3 + 1$	1001	9

GF(2^4)'ün herhangi iki elemanının çarpımı 1.4. eşitliğiyle bulunabileceği gibi elemanların kuvvetleri mod 15'e göre toplanarak da bulunabilir. Örneğin $\alpha^8.\alpha^{13} = \alpha^{21} = \alpha^6$ olur. İki elemanı bölmek için bölenin çarpmaya göre tersi bölünenle çarpılır. α^i 'nin çarpmaya göre tersi α^{15-i} olduğundan, örneğin

$\frac{\alpha^6}{\alpha^{14}} = \alpha^6.\alpha^{15-14} = \alpha^6.\alpha = \alpha^7$ olur.

GF(2^4)'ün iki elemanının toplamı, elemanların polinomsal gösterilimi ve 1.3. eşitliğinden yararlanılarak bulunabileceği gibi vektörel XOR işlemi ile de hesaplanabilir. Örneğin, $\alpha^6 + \alpha^9 = (0011) \oplus (1101) = (1110) = \alpha^{10}$ olur.

Katsayıları gerçel sayılar olan bir polinomun sanal kökleri olması gibi, katsayılar GF(2)'den alınan bir polinomun kökleri GF(2^m)'den olabilir. Örneğin $p(X)=X^4+X^3+1$, GF(2) üzerinde azaltılamaz bir polinomdur. p(0)=p(1)=1 olduğundan p(X)'in GF(2) de bir kökü yoktur. Fakat GF(2^4)'ün elemanı olan $\alpha^7,\alpha^{11},\alpha^{13},\alpha^{14}$ p(X)'in kökleridir ve bunlar,

$$p(\alpha^7)=p(\alpha^{11})=p(\alpha^{13})=p(\alpha^{14})=0$$
$$p(X)=(X+\alpha^7)(X+\alpha^{11})(X+\alpha^{13})(X+\alpha^{14})$$

olarak belirlenir.

Teorem 1: f(X), GF(2) üzerinde tanımlı bir polinom , $\beta\varepsilon GF(2^m)$ olmak üzere, eğer β f(X)'in kökü ise i = 0 için β^{2^i} de f(X)'in bir köküdür[53].

Tablo 4. GF(2^4)'ün elemanlarının minimal polinomları

Eleman	**Minimal Polinom**
0	X
1	$X+1$
$\alpha,\alpha^2,\alpha^4,\alpha^8$	X^4+X+1
$\alpha^3,\alpha^6,\alpha^9,\alpha^{12}$	$X^4+X^3+X^2+X+1$
α^5,α^{10}	X^2+X+1
$\alpha^7,\alpha^{11},\alpha^{13},\alpha^{14}$	X^4+X^3+X+1

Tanım 4: Eğer $\Phi(X),\Phi(\beta)=0$ koşulunu sağlayan minimum dereceli polinom ise $\Phi(X)$'e β 'nin minimal polinomu denir. GF(2^4)'ün elemanlarının minimal polinomları Tablo 4'te verilmiştir.

Galois Alanının yukarıda belirtilen özellikleri hata düzelten kodların tasarımında, kodlama ve kod çözme aşamalarında kullanılmaktadır. Kodlama ve kod çözme sırasında karşılaşılacak bütün denklemlerin katsayıları GF(2^m)'in elemanı olup eşitliklerin çözülmesinde Galois Alanı içinde tanımlı işlemler kullanılacaktır.

1.7. Kanal Kodlama Teknikleri

Sayısal iletişim sistemlerinde kullanılan kanal kodlama teknikleri bilgiyi, gürültü ve diğer bozucu etkenlerden korumak ve bit hata oranını düşürmek için kullanılır. Kodlama tekniğinde en büyük kaygı, güvenilir bir iletişim sağlayabilmek için hataların kontrolünün sağlanmasıdır. Kanal kodlama, genellikle gönderilecek bilgiye belirli bir seçicilikle artık bitler eklenerek gerçekleştirilir. Bu artık bitler hataların algılanmasına ve düzeltilmesine olanak verir ve daha güvenilir bir bilgi akışı gerçekleştirmeyi sağlar. Bilgiyi korumak için kullandığımız kanal kodlamanın kullanıcıya maliyeti, veri hızında bir azalma veya bant genişliğinde istenmeyen bir artışdır.

Kodlama tekniklerinden bahsetmeye başlamadan önce birkaç tanımdan söz etmek gerekir. Bir kodun Hamming ağırlığı, kod kelimesinin içerdiği 1 değerli bitlerin sayısıdır. d ile gösterilen, iki kod arasındaki Hamming mesafesi ise, aynı bit pozisyonundaki farklı değerli bitlerin sayısı olarak tanımlanmaktadır. Minimum Hamming mesafesi ($d_{\mathrm{m\,i}}$) de, bir kod içindeki iki kod kelimesinin arasındaki minimum Hamming mesafesini ifade eder. Bir kodun ağırlık dağılımı veya mesafe spektrumu, mümkün olan her ağırlıktaki kod kelimelerinin sayısıdır.

Kanal kodlama teknikleri başlıca dört ana başlık altında incelenebilir: Blok kodlar, konvolüsyonel kodlar, ardışıl kodlar ve turbo kodlar.

1.7.1. Blok Kodlar

Blok kodlar, yeniden iletime gerek kalmadan, sınırlı sayıdaki hatanın algılanmasına ve düzeltilmesine olanak sağlar. Bir blok kod, *k* adet ikili giriş sembolünü, *n* adet ikili çıkış sembolüne dönüştürür. Blok kodlayıcı hafızasız bir birimdir. Yani kod kelimesi içindeki her bit, bir önceki bitten bağımsızdır. $n > k$ olduğundan dolayı, seçilen kod, kod kelimesi uzunluğunda bir artışa sebep olacaktır. Bu artık bitler, eşlik bitlerinde olduğu gibi, kod çözücü tarafından hata algılanmasında ve düzeltilmesinde kullanılacaktır. Kod hızı $R = k/n$ şeklinde tanımlandığında, kodlar da (n, k) şeklinde gösterilir. Kod hızı R'nin pratikteki değeri $1/2$ ila $7/8$ arasında değişmektedir. Blok kodların hata düzeltme kapasitesi kod mesafesinin bir fonksiyonudur. Blok kodlar aşağıdaki özelliklerle sınıflandırılırlar.

Doğrusallık: Bir kodun doğrusal olduğunu söyleyebilmemiz için, kodda ki iki kod kelimesinin toplamının başka bir kod kelimesi oluşturması gerekir. Doğrusal bir kod, tümü sıfır olan bir kod kelimesi içermelidir.

Sistematiklik: Sistematik kod, eşlik bitlerinin bilgi bitlerinin sonuna eklendiği bir koddur. Gönderilen datayı alıcı aynı formda içerir. Dolayısı ile ses yada görüntü türünden sistemler için alıcı kod çözme işlemi devre dışı bırakılabilir.

Cyclic: Cyclic kodlar, blok kodlar ailesinin bir üyesidir. Bir kodun cyclic olabilmesi için kod kelimesi bir bit sağa kaydırıldığında ve en sağdan düşen bit sol başa eklendiğinde yeni bir kod kelimesi oluşması gerekir.

İkili aritmetikte, modül - 2 toplama ve çıkartma kullanılmaktadır. Bu aritmetik gerçekte, 2'nin 0 olarak düşünülmesi hariç, bilinen aritmetikle aynıdır.

Pratikte Hamming kodları, Hadamard kodları, Golay kodları, Cyclic kodlar, BCH kodları ve Reed-Solomon kodları gibi blok kodlarla sıklıkla karşılaşılmaktadır. Burada bunların dört tanesinden bahsedelim.

Hamming kodlar: Hamming kodlar ilk önemli hata düzeltme kodları arasındadır[5]. İkili ve ikili olamayan kodlara sahiptir. İkili bir Hamming kod şu özelliğe sahiptir.

$$(n,\ k) = (2^m - 1,\ 2^m - m - 1) \tag{1.5}$$

Burada k, n bitlik kod kelimesi oluşturmak için kullanılan bilgi bitleri sayısı ve $m \geq 2$'dir.

Eşlik sembolleri sayısı $m = n - k$'dır. Örneğin, $m = 3$ ise $(7,\ )$ bir koda sahibiz demektir. Bu ikili kodların minimum mesafesi 3'tür. Blok içindeki tek bir hatayı düzeltme veya iki yada daha az sayıdaki hataların bütün kombinasyonlarının algılama yeteneğine sahiptir[6].

Golay kodları: Golay kodlar minimum mesafesi 7 olan doğrusal, ikili $(23,12)$ kodlardır. (23,12)'lik koda bir eşlik biti eklenerek genişletilmiş Golay kodu (24,12) elde edilir. (24,12)'lik bir kodun minimum mesafesi 8'dir. (23,12) Golay kodun ve genişletilmiş (24,12) Golay kodun ağırlık dağılımı bilinmektedir[7]. Genişletilmiş Golay kodlar, Hamming kodlara göre oldukça güçlüdür. Genişletilmiş Golay kod üç hatayı düzeltme garantisi verir.

Bose – Chaudhuri – Hocquenghem (BCH) kodlar: BCH kodlar en önemli blok kodlar arasındadır. Çünkü geniş bir data hızı aralığına sahiptir ve önemli kodlama kazançlarına ulaşabilmektedir. Ayrıca yüksek hızlarda da uygulanabilmektedir[7]. BCH kodlar, Hamming kodların birden çok hata düzeltimine imkan veren genelleştirilmiş bir şeklidir. Kodun blok uzunluğu $n = 2^m - 1$'dir $(m \geq 3)$. Düzeltebileceği hata sayısı t ile sınırlandırılmıştır. Burada $t < \left[\left(2^m - 1\right)/2\right]$'dir. İkili olmayan BCH kodların en önemli ve yaygın sınıfı Reed-Solomon kodları olarak bilinen kod ailesidir. (63,47) Reed-Solomon kodu kullanan U.S. Hücresel Sayısal Paket Veri (Cellular Digital Packet Data (CDPD)) sistemi, kod sembolü başına altı bit kullanır $(m = 6)$ [3].

Reed – Solomon kodları: Reed – Solomon kodları BCH kodunun özel bir sınıfı olup ikili olmayan bir setten oluşur. Reed – Solomon kodlar ardışıl hataları düzeltme yeteneğine sahiptir ve genellikle ardışıl kodlama sistemlerinde kullanılırlar. Kodun minimum mesafesi[6] $d_{min} = n - k + 1$'dir. Burada k kodlanacak data sembol sayısı, n ise kodlanan bloktaki toplam sembol sayısıdır. Bu kod t veya daha az sayıdaki sembol hatasının bütün kombinasyonlarını düzeltebilecek kapasitededir. Burada $t = (d_{min} - 1)/2 = (n - k)/2$'dir. Buradan t adet hatayı düzeltmek için kullanılacak eşlik sembollerinin sayısı $n - k = 2t$ olarak bulunur.

1.7.2. Konvolüsyonel Kodlar

Blok kodların kullanımıyla performans başarımında önemli bir artış elde edilmesine rağmen, blok kodların kendi içyapısından kaynaklanan birkaç dezavantaja sahip olması nedeniyle, bu tür kanal kodlama tekniğinin iletişim sistemlerinin tasarımında kullanılması sıkıntı meydana getirmektedir.

1. Blok kodlar çerçeve uyumlu olması nedeniyle kod çözme işlemi başlamadan önce iletilen bütün blokların alıcı tarafından alınması gerekmektedir. Bunun sonucunda, özellikle büyük blok uzunluklarında sistemde tahammül edilemeyecek bir gecikme oluşmaktadır.

2. Blok kodlar, kesin (çok iyi) bir çerçeve senkronizasyonuna gerek duymaktadır.

3. Blok kodlarda kullanılan kod çözücüler, sıfır-bir kararına dayalı olarak çalışmaktadır. Sıfır-bir kararına dayalı olarak çalışan kod çözücülerde, kanal çıkışında alınan bilgi ikili düzende (0 veya 1) olmasına rağmen, yumuşak-tahminli kod çözücülerde kanalın çıkışında alınan bilgi sürekli (reel) bir değer olacaktır. Oysa Shannon tarafından tanımlanan teorik performans sınırına ulaşabilmek için sürekli değerde kanal çıkışına ihtiyaç duyulmaktadır. Bu nedenden dolayı, genel olarak blok kodlar, iyi kanallarda etkileyici bir performans oranı yakalamasına rağmen, güç tüketiminde verimli olamadıkları için işaret/gürültü oranının düşük olduğu değerlerde kötü bir performans sergilemektedir. Yalnız unutulmamalıdır ki, blok kodların düşük işaret/gürültü oranı değerlerinde kötü bir performans sergilemesi, blok kodların kendi içyapılarından dolayı değil de daha çok bu kodların kod çözme işleminde kullanmış olduğu sıfır-bir kararına dayalı olarak çalışan kod çözücülerden kaynaklanmaktadır. Gerçekte, blok kodlarda yumuşak-tahminli kod çözücüler kullanma, olası bir ihtimal olmasına rağmen, işlem karmaşıklığının artacağı düşünülmesinden dolayı tercih edilmemektedir.

Blok kodların dezavantajlarından, kodlama için farklı bir yaklaşım olan ve ilk kez 1955'te Elias tarafından ileri sürülen konvolüsyonel kodların ele alınması ile kaçınılmıştır [8]. Konvolüsyonel kodlar günümüzde kullanımı yaygınlaşmış güçlü kodlar olup giriş bilgisi bloklar halinde gruplanmayıp, bir kaç tane giriş bilgisi üzerinden koda özgü fonksiyonlar kullanarak katlanması ile kodlanmaktadır.

Bir başka deyişle konvolüsyonel kodlar, veri bitlerinin sonlu duruma sahip doğrusal kaydırmalı kaydedicilerden geçirilerek iletilmesi ile oluşmaktadır. Bu özelliği sayesinde konvolüsyonel kodlarda, hem blok kodlarda olduğu gibi veri bloğunun hazır olup olmaması gibi bir sorunla karşılaşılmamakta hem de blok kodlara oranla konvolüsyonel kodlarda kodlama işlemi daha kısa bir süre içerisinde gerçekleşmektedir. Genel olarak konvolüsyonel kodlayıcının çıkışındaki n adet çıkış biti, k adet giriş bitinin, kaydırmalı kaydediciler içerisinde saklanan m adet bit ile doğrusal kombinasyonu sonucunda elde edilmektedir. Bu durumda kodlayıcı hızı $R = k/n$ olmaktadır. Her bir çıkışın bağlı olduğu bitlerin toplam sayısı, sınır uzunluğu, K, olarak adlandırılmaktadır. Ayrıca, konvolüsyonel kod çözücüde kullanılan algoritma tamamen demodülatör çıkışındaki yumuşak bilgiye dayalıdır.

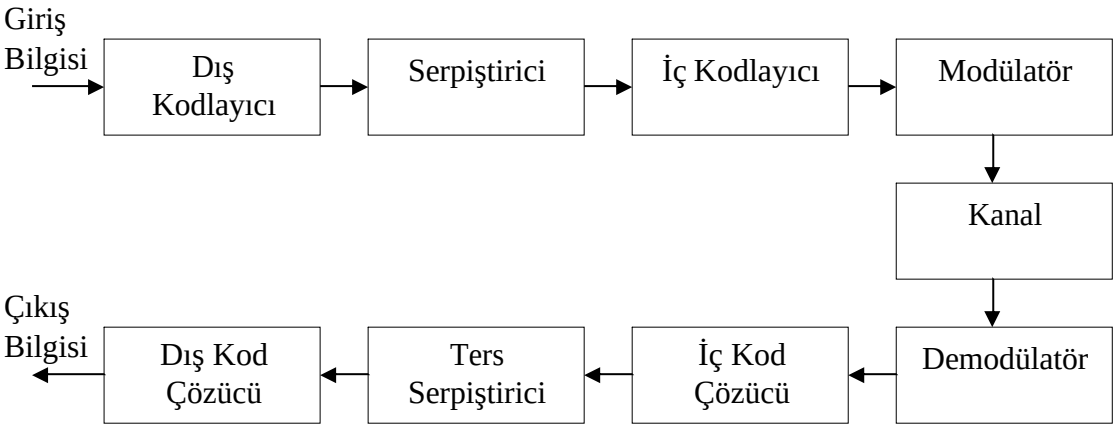
İlk pratik kod çözücü algoritması 1961'de Wozencraft ve Reiffen'in [9] ardışık kod çözücü algoritmasıdır. Bu algoritma daha sonra 1963'te Fano [10] ve 1969'da Jelinek [11] tarafından geliştirilmiştir. Yalnız bu algoritma, kodlamada büyük bir başarı sağlamasına rağmen 1967 yılında ortaya çıkan Viterbi algoritması [12] ile popülerliğini kaybetmiştir. Günümüzde Viterbi algoritması konvolüsyonel kodlar için optimum çözümü teşkil etmektedir.

Viterbi algoritmasının keşfinden sonra konvolüsyonel kodlayıcılar, iletişimde geniş bir uygulama alanı bulmuştur. Sınır uzunluğu $K = 5$, kod oranı $r = 1/2$ ve $r = 1/3$ olan "Odenwalder" konvolüsyonel kodu ticari uydular arası iletişim uygulamalarında bir standart olmuştur [13]. "Voyager" ve " Pioneer" gibi birkaç konvolüsyonel kod derin uzay araştırmalarında kullanılmıştır [14]. Benzer şekilde, ikinci nesil tüm sayısal hücresel standartlar konvolüsyonel kodları kullanmaktadır. GSM, $r = 1/2$, $K = 5$ konvolüsyonel kod [3], Globalstar $r = 1/2$, $K = 9$ konvolüsyonel kod, Iridium $r = 3/4$, $K = 7$ konvolüsyonel kod [15] kullanmaktadır. Ayrıca, şu an düşünce aşamasında olan üçüncü nesil standartlarının belli bir kısmında konvolüsyonel kodlama tekniğinden yararlanılmaktadır.

1.7.3. Ardışıl Kodlar

Konvolüsyonel kodlar günümüzde, sıkça kullanılan bir kodlama tekniği olmasına rağmen blok kodlardaki kadar çok olmasa da önemli bir dezavantaja sahiptir. Konvolüsyonel kodlar, hata patlaması (ard arda hatalı bitlerin gelmesi) olması durumunda istenilen düzeyde bir performans sergileyememektedir. Bu dezavantajı ortadan kaldırmak için yapılan çalışmalar sonucunda, 1966 yılında David Forney tarafından [16], ardışıl kod olarak adlandırılan yeni bir kodlama tekniği tasarlanmıştır. Ardışıl kodlar, iki veya daha fazla basit kodlayıcının yüksek kod kazancına ulaşmak için birleşmesinden ibarettir.

Ardışıl kodlar, istenen hata performansına ulaşmak için, biri iç diğeri dış kod olmak üzere iki seviyeli kod kullanırlar. Şekil 4'de basit bir ardışıl kod yapısı görülmektedir. Bu şekilde, kodlama ve kod çözme sırası da belirtilmektedir. İç kod genellikle kanal hatalarının çoğunu düzeltmek için tasarlanır. İç kod bloğunun çıkışında oluşabilecek bir ardışıl hatayı yayabilmek için, iç ve dış kod arasına bir serpiştirici bloğu yerleştirilmiştir.



Şekil 4. Basit bir ardışıl kod yapısı

Tipik örnekler, bize iç kodun daha güçlü ve düşük kod hızlı tasarlandığını, dış kodun ise daha yüksek kod hızına sahip olduğunu göstermektedir[17]. Massey[18], katlamalı kodların, kanal durum bilgisini ve yumuşak karar bilgisini kanaldan kolayca elde edebileceğini belirterek bu kodun kod çözme işleminin ilk adımı olması gerektiğini belirtmiştir. Reed – Solomon kodları Viterbi kod çözücüden artan hataları temizlemek için kullanılır. Viterbi ve Reed – Solomon kodları birbirini çok iyi tamamlayan kodlardır.

Viterbi kod çözücüyü iç kod, Reed – Solomon'u dış kod olarak kullanan ardışıl kodlama sistemlerinden en popüler olanlarından biri, iki kod adımı arasında serpiştirici kullanan sistemdir[13]. Bu sistemin uygulanmasıyla 2 ila 2.5 dB'lik E_b / N_0 oranı için $p = 10^{-5}$'lik bir hata olasılığına ulaşılmıştır ki bu Shannon sınırından sadece 4 dB uzaktadır. Bu özellik, ardışıl sistemlerin, derin uzay görevlerinde, NASA ve ESA'nın Uzay Bilgi Sistemleri için Danışma Komiteleri tarafından seçilmesinin nedenlerinden biridir[19].

Ardışıl kodları, kısa kodlardan uzun kodlar oluşturma metodu olarak özetleyebiliriz. Bu yapı, kod çözümü için uzun kodların karmaşık kod çözme algoritmalarına gerek duymayan uzun blok kod yapımını doğurmuştur.

1.7.4. Turbo Kodlar

Kodlama kuramında, son yılların belki de en ilginç ve en önemli olayı turbo kodların keşfidir [20]. Turbo kodlar, 1993 yıllarının başlarında önerilen ve hata başarımının Shannon limit değer eğrilerine yaklaşmasına neden olan çağdaş kodlama tekniğidir. Bu kodlama tekniği, Berrou, Glavieux ve Thitimajshima tarafından ortaya atılmıştır [20]. Bu üç bilim adamı yaptıkları çalışmalar sonucunda, 10^{-5} bit hata olasılığı (*probability of bit error* – P_e) için sinyal/gürültü oranını 0.7 dB bulmuşlardır. Ortaya çıkardıkları bu yeni kod türünde kodlayıcı, bir tane serpiştiriciyle iki tane özyinelemeli sistematik konvolüsyonel kodlayıcının (RSC) paralel olarak bağlanmasından oluşmaktaydı. Serpiştirici, bilgi kaynağının ürettiği L uzunluklu bilgi bloklarında simgelerin yerini değiştirerek ikinci özyinelemeli sistematik kodlayıcı için özgün L uzunluklu bilgi dizisi oluşturmaktaydı. Ayrıca kodlama genelinde kod oranı $1/3$ olmasına rağmen, delme işlemiyle kod oranını, 1'den küçük olmak şartıyla, daha yüksek oranlara yükseltmek mümkündü. Konvolüsyonel kodlarda maksimum olasılığın elde edilmesi için kullanılan Viterbi algoritması, serpiştiricilerden dolayı hesap karmaşıklığının artmasına neden olacağından bu kodda kullanılmamıştır. Bu nedenden dolayı kod çözücüde Bahl algoritmasından yaralanılmıştır.

1994 yılında, P. Robertson [21] ve Hagenauer [22], turbo kodlar hakkında çok ilginç bir görüş sunarak, turbo kodlarda işlem karmaşasını önemli ölçüde azaltan yeni bir Bahl algoritmasını ortaya koymuştur. 1996 yılında S. Benedetto ve G. Montorsi turbo kodların yapısına açıklık kazandırmış [23], [24] ve aynı yıl içerisinde L. Perez ortaya koyduğu "Spectral Thinning" adlı teoremiyle turbo kodların anlaşılmasına katkıda bulunmuştur. Yapmış oldukları analizler, turbo kodların, daha önceleri bilinen çeşitli kavramların üzerine inşa edilen zekice bir tasarım sonucu ortaya çıktığını göstermiştir [23].

Ne yazık ki, günümüze kadar turbo kodları teorik olarak açıklayan pek fazla çalışma bulunmamaktadır. Ancak, turbo kodlara farklı açılardan zenginlik kazandırmak için bir çok simülasyon çalışmasına ihtiyaç vardır.

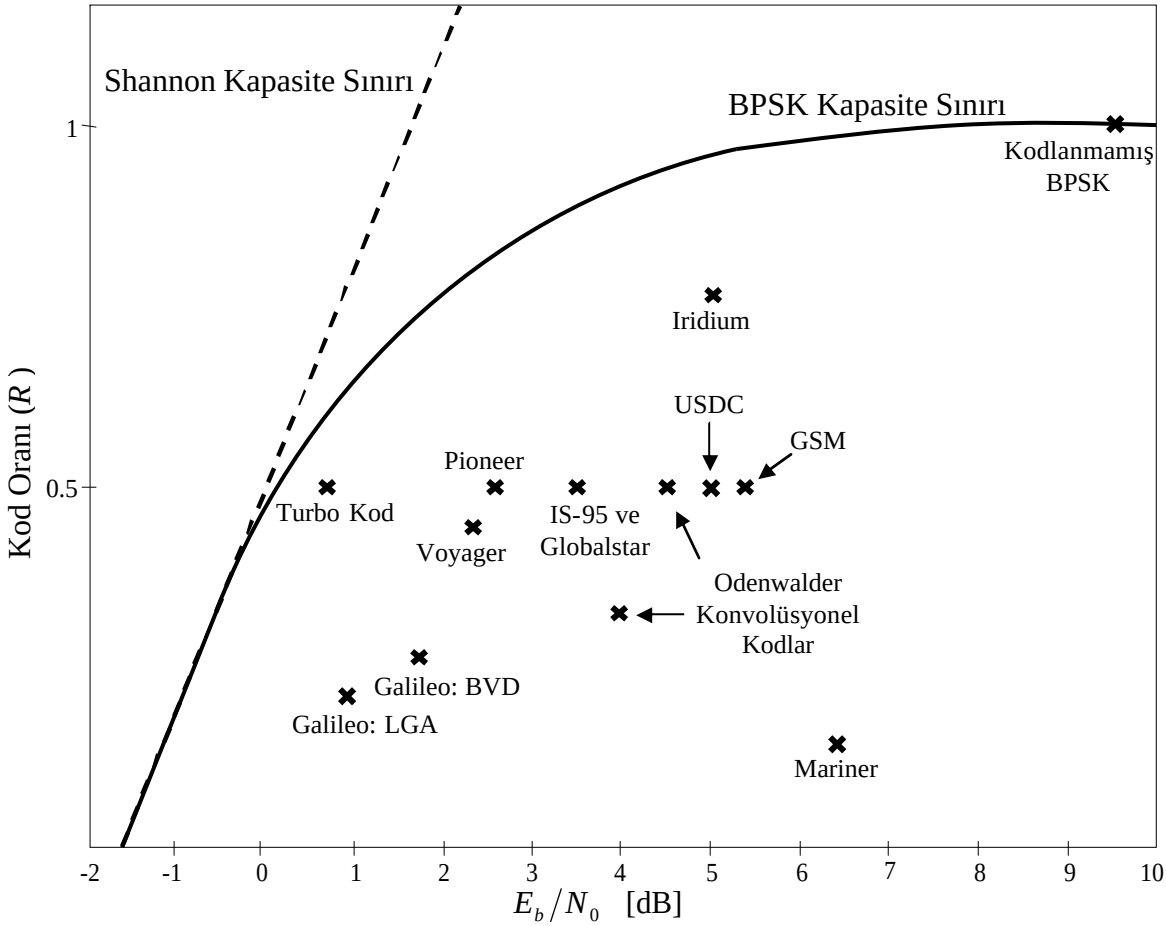
1.7.5. Kodlama Sistemlerinin Karşılaştırılması

Şekil 5'de birçok uygulama alanında kullanılmış olan sistemlere ve standartlara ait hata düzeltme kodlarının performans karşılaştırılması görülmektedir.

Burada, x-ekseni güç verimliliğini temsil etmekte ve bit başına düşen enerjinin tek taraflı güç yoğunluğuna oranı, E_b/N_0 , şeklinde ifade edilmektedir. Benzer şekilde y-ekseni ise spektral verimliliği temsil etmekte ve kod oranı, *R*, ile ifade edilmektedir. İletim ortamının, toplanır beyaz Gauss gürültülü (AWGN) kanal olduğu varsayılmaktadır.

Eğri üzerindeki her bir noktanın, kullanılan modülasyon türünün ikili faz kaydırmalı anahtarlama (BPSK) ve bit-hata oranının (BER) $P_e = 10^{-5}$ olması durumunda elde edildiği farz edilmektedir. "Shannon Kapasite Sınırı" olarak adlandırılan eğri, kod oranı fonksiyonuna bağlı olarak gerçeklenebilir bir iletişim sağlamak için gerekli olan minimum E_b/N_0 'ı ifade etmektedir. Yalnız kullanılan modülasyon türü BPSK olmasından dolayı gerçek sınır, "BPSK Kapasite Sınırı" olarak adlandırılan eğridir.

BPSK modülasyonunda, herhangi bir hata düzetme kodlaması kullanılmadığı zaman $P_e = 10^{-5}$ elde etmek için $E_b/N_0 = 9.6$ dB'ye ihtiyaç vardır. Eğer herhangi bir kodlama tekniği kullanılmış olsaydı, gerekli olan E_b/N_0 değeri azalacaktı. Fakat bunun akabinde, spektral verimlilikte bir azalma ve alıcının karmaşıklığında bir artış gözlenecekti. Kodlama tekniği kullanıldığı zaman istenilen P_e değerini elde etmek için gerekli olan E_b/N_0 değeri ile kodlanmamış BPSK'da istenilen P_e değerini elde etmek için gerekli olan E_b/N_0 değeri arasındaki fark, kodlama kazancı olarak adlandırılmaktadır.



Şekil 5. AWGN kanalda hata düzeltme kodlaması ve BPSK modülasyonu kullanan bazı sistem ve standartların karşılaştırılması [25].

Şekil 5'de gösterilen en eski kod, 1969'da Mars'ta Mariner Görevi'nde kullanılan (36, 6) Reed-Muller kodudur. Bu kod 3.6 dB'lik bir kodlama kazancı sağlamasına rağmen, kod oranı sadece $R = 0.1875$ idi. Bu değer, o kadar önemli bir kodlama kazancı olarak görülmese de, o zamanlarda her bir dB'lik kodlama kazancı sistem maliyetinde yaklaşık $1,000,000 Amerikan Doları kadar bir azalmaya neden olmaktaydı [15]. Pioneer 10 (1972 - 1973 Jüpiter), Pioneer 11 (1973 - 1977 Satürn), Pioneer 12 (1978 Venüs) görevlerinde, Mariner görevine oranla daha iyi bir kodlama kazancı elde edilmiştir. Pioneer görevlerinde, kod oranı $R = 1/2$ ve sınır uzunluğu $K = 32$ olan konvolüsyonel kod kullanılarak, 6.9 dB'lik kodlama kazancına ulaşılmıştır.

Şekil 5'de gösterilen sistemlerin çoğu, konvolüsyonel kodları kullanmaktadır. Sınır uzunluğu $K = 7$ ve kod oranları $R = 1/2$ ve $R = 1/3$ olan Odenwalder kodları birçok uydu haberleşmesi uygulamasında kullanılmıştır.

Kod oranı $R = 1/2$ olan Odenwalder kodu, 5.1 dB'lik bir kodlama kazancı sağlarken, kod oranı $R = 1/3$ olan Odenwalder kodu, 5.6 dB'lik bir kodlama kazancı sağlamaktadır. GSM standardında sınır uzunluğu $K = 5$, USDN standardında sınır uzunluğu $K = 6$ ve IS - 95 standardında ise sınır uzunluğu $K = 9$ olan konvolüsyonel kodlar kullanılarak sırasıyla 4.3 dB, 4.6 dB ve 6.1 dB'lik kodlama kazançları elde edilmiştir [26][1]. Globalstar'da kullanılan kod, IS - 95 standardında kullanılan kod ile özdeş olup aynı kodlama kazancını verirken, Iridium'da sınır uzunluğu $K = 7$ ve kod oranı $R = 3/4$ olan konvolüsyonel kodlayıcı kullanılarak 4.6 dB'lik kodlama kazancı elde edilmiştir. Büyük Viterbi Kod Çözücüsü (*Big Viterbi Decoder*-BVD), Jüpiter'deki Galileo görevinde kullanılmak üzere tasarlanmış bir sistem olup, bu sistemde sınır uzunluğu $K = 15$ ve kod oranı $R = 1/4$ olan konvolüsyonel kod kullanılarak 7.9 dB'lik kodlama kazancı sağlanmıştır [27].

Voyager görevlerinde (Voyager 1-2, 1979, Jüpiter, Satürn, Uranüs, Neptün) kullanılan sistemde, kod oranı $R = 1/2$ olan Odenwalder kodunun ve ($k = 223$, $t_e = 16$, $q = 2^8$, $n = 255$) Reed-Solomon kodunun seri bir biçimde bağlanması sonucunda elde edilen ardışıl kod kullanılmıştır [15]. Kod hızı $R = 0.44$ olan bu sistemde 7.1 dB'lik bir kazanç elde edilmiştir. Ayrıca, Jüpiter'deki Galileo görevinde kullanılan sistemde, konvolüsyonel kod ile ($q = 2^8$, $n = 255$, $k = 223$, $t_e = 16$) Reed-Solomon kodları arasına bir serpiştirici yerleştirmek suretiyle elde edilen bir ardışıl kod türü kullanılmıştır. Bu ardışıl kod ile konvolüsyonel kodlayıcının doğal yapısından ötürü ortaya çıkan hata patlamalarına karşı önlem alınmaya çalışılmıştır.

Aynı zamanda Şekil 5'de, orijinal turbo koda özgü performans da gösterilmektedir [28]. Şekilden de görüldüğü gibi turbo kod, diğer kodlara oranla BPSK kapasite sınırına oldukça yaklaşmıştır. Ayrıca, turbo kodda kullanılan kod çözücü ile BVD kodda kullanılan kod çözücü aynı karmaşıklığa sahiptir [29].

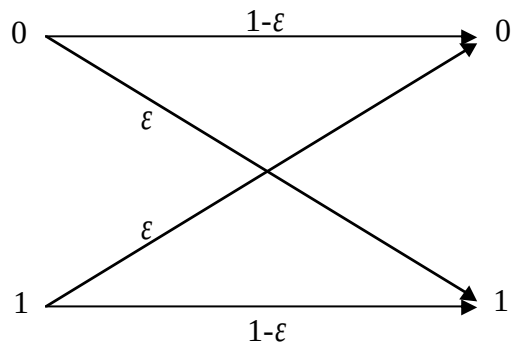
[1] Bu standartlar, normal koşullar altında BPSK modülasyonunu kullanmamaktadır. Şekil 5'de gösterilen performans noktaları, sistemlerde kullanılan gerçek kodların sonucunda elde edilmesine rağmen, performans kıyaslaması yapabilmek için tüm veri noktalarının, BPSK modülasyonun kullanımı sonucunda elde edildiği varsayılmaktadır.

1.8. Kablosuz Kanal Modelleri

1.8.1. İkili Simetrik Kanal

İkili simetrik kanal (*binary symmetric channel*-BSC), ayrık hafızasız kanalın (*discrete memoryless channel* - DMC) özel bir türünü oluşturmaktadır. BSC, ayrık giriş ve çıkışlara sahip olmasından dolayı, belirli bir andaki çıkış sadece o andaki girişe bağlı olup kanalın önceki giriş veya çıkışlarından bağımsızdır [30], [31].

BSC'de, "0" bilgi biti gönderildiği zaman ikili simetrik kanalın girişinde bilgi bitinin "1" olarak alınma olasılığı ε ile ifade edilirse, bilgi bitinin "0" olarak alınamama olasılığı $1-\varepsilon$ olarak tanımlanabilir. Benzer şekilde, kanalın simetrik olmasından dolayı, gönderilen bilgi bitinin "1" olması durumunda kanalın girişinde bilgi bitinin "0" olarak alınma olasılığı ε ile ifade edilirse, bilgi bitinin "1" olarak alınamama olasılığı $1-\varepsilon$ olarak tanımlanabilir. Şekil 6'da bu durum gösterilmektedir.

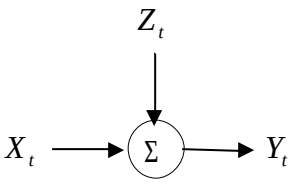


Şekil 6. İkili simetrik kanal modeli

1.8.2. Toplanır Beyaz Gauss Gürültülü Kanal

Toplanır beyaz Gauss gürültülü (AWGN) kanal, iletişim sistemlerinin modellenmesinde oldukça sık kullanılan bir kanal türüdür. AWGN kanalında, iletilen işaretler Gauss dağılımına sahip gürültü tarafından bozulmaktadır. Şekil 7'de de görüldüğü gibi, t anında kanalın çıkışını Y_t, girişini de X_t olarak tanımlarsak, kanalın çıkışını aşağıdaki gibi ifade edebiliriz[32].

$$Y_t = X_t + Z_t \tag{1.6}$$



Şekil 7. AWGN kanal modeli

Burada Z_t, sıfır-ortalamalı Gauss gürültüsünü temsil etmekte ve olasılık dağılım fonksiyonu $p(z)$ ile tanımlanmaktadır.

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z^2}{2\sigma^2}} \tag{1.7}$$

Varyans, σ^2, gürültü ve işaretin birbirinden bağımsız olduğu düşünülerek, gürültü tayfının gücü cinsinden

$$\sigma^2 = \frac{N_0}{2} \tag{1.8}$$

ifade edilmektedir.

1.8.3. Yavaş Sönümlü Dar Bantlı Rayleigh Kanalı

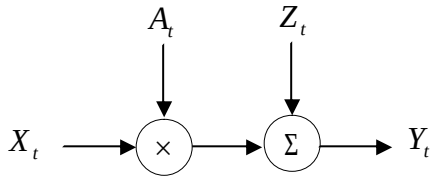
Kablosuz haberleşmedeki birçok uygulamada kanallar, AWGN kanala oranla daha karmaşık bir yapıya sahiptir. Bu karmaşıklığa sebep olan en büyük etken bayılmadır[33]. Bayılma, gönderilen işaretin kanal içerisinde yayılımı sırasında meydana gelmektedir. Kablosuz iletişim sisteminde, gürültü nedeniyle bir bozulma olmadığı varsayılsa bile, gönderilen işaret çok-yollu yayılımdan, atmosferik olaylardan vb. etkenlerden dolayı git gide zayıflayacaktır. Örneğin, bir gök dalgası iyonosferden toprağa dönüp, sonra yansıyarak tekrar iyonosferde kırıldıktan sonra antene ulaşıyor.

Aynı zamanda başka bir gök dalgası iyonosferden kırılarak alıcı antene ulaşıyorsa, faz farkından dolayı alıcıda alınan işaretin gücünde zaman içerisinde artma veya azalma yani bir dalgalanma meydana gelir.

Şekil 8'de de görüldüğü gibi yavaş sönümlü Rayleigh kanalında, t anında kanalın girişi X_t ve çıkışı Y_t olarak adlandırılır ise, Y_t ve X_t arasındaki ilişki denklem (1.9) yardımıyla ifade edilebilir.

$$Y_t = A_t . X_t + Z_t \tag{1.9}$$

Burada Z_t, sıfır ortalamalı toplanır beyaz Gaussian gürültüsünü temsil etmekte ve gürültü varyans $\sigma^2 = N_0/2$ şeklinde hesaplanmaktadır.



Şekil 8. Yavaş sönümlü dar bantlı Rayleigh kanal modeli

A_t ise, Rayleigh dağılımına sahip rastgele bir değişken olup kanal kazancı veya bayılma katsayısı

$$p_A(a) = 2a e^{-a^2}, \qquad a \geq 0 \tag{1.10}$$

olarak adlandırılmaktadır.

1.9. Serpiştirici

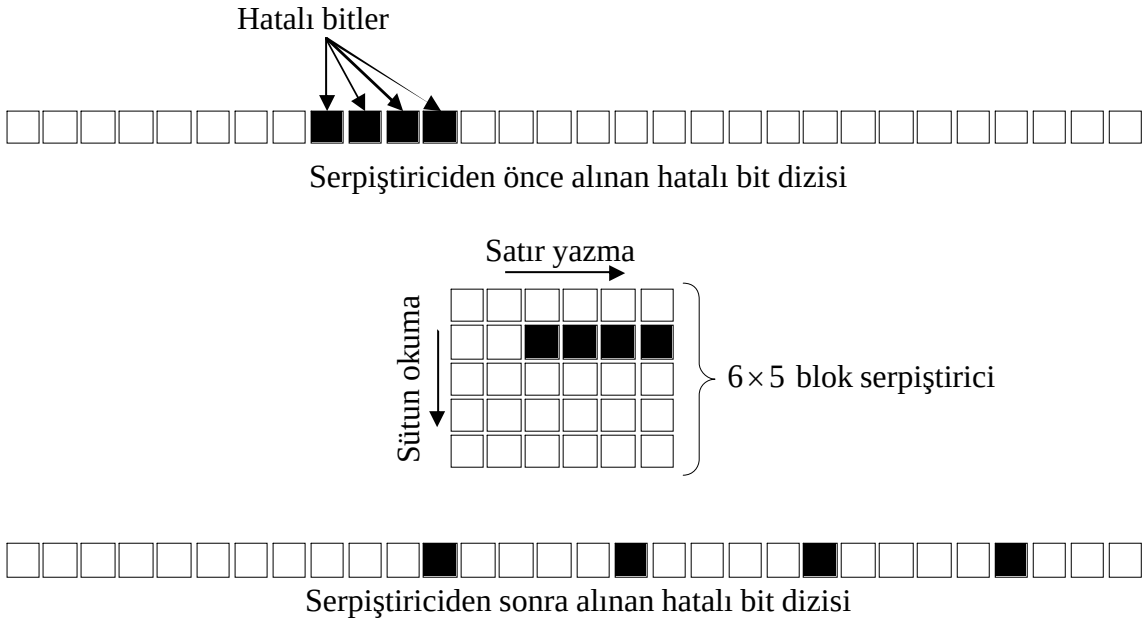
Konvolüsyonel kodlar ve blok kodların büyük bir çoğunluğu tek şekilde dağılmış hatalara karşı dayanıklı olup hataların gruplar halinde gelmesi durumunda hata düzeltme kapasiteleri oldukça düşmektedir. Pratikte, özellikle gezgin haberleşmede bayılmalardan dolayı oluşan düşük SNR oranları hataların gruplar halinde gelmesine sebep olmakta, ve kullanılan kodlama türüne göre bitlerin uygun bir şekilde serpiştirilmesini zorunlu kılmaktadır.

Bir serpiştirici, önceden belirlenmiş fonksiyonu sayesinde girişinde bulunan giriş dizisindeki bitlerin yerlerini değiştirerek çıkışında giriş dizisiyle olabildiğince ilişkisiz bir dizi üreten bir devredir. Giriş dizisinde zamanda birbirine yakın olan bitler serpiştiricinin çıkışında birbirinden uzaklaştırılarak giriş dizisiyle çıkış dizisi arasındaki ilişki küçültülmektedir. Genellikle bir serpiştirici hata patlamalarını düzgün dağıtmak için kullanılır. Düzgün dağıtmakta amaçlanan, simge bloklarının haberleşme kanalından iletiminde, bilgi taşıyan simgeleri bozan kanal gürültüsünün yeniden şekillendirilmesidir. Bu şekillendirme alıcıda hatalı olarak alınan ardışıl simgelerin birbirinden serpiştirici sayesinde uzaklaştırılmasıyla yapılmaktadır. Kanal içerisinde gruplar halinde hatalar oluşacağı dikkate alınırsa, en mantıklı olanı klasik kodlama tekniklerinde olduğu gibi serpiştiriciyi kanal kodlayıcı ile kanal arasına yerleştirmektir.

Bu amaçla literatürde çeşitli serpiştirici oluşturma yöntemleri geliştirilmiştir. Bu serpiştirici yöntemlerinden üç tanesi sıkça kullanılmaktadır [51], [52]:

1.9.1. Blok Serpiştirici

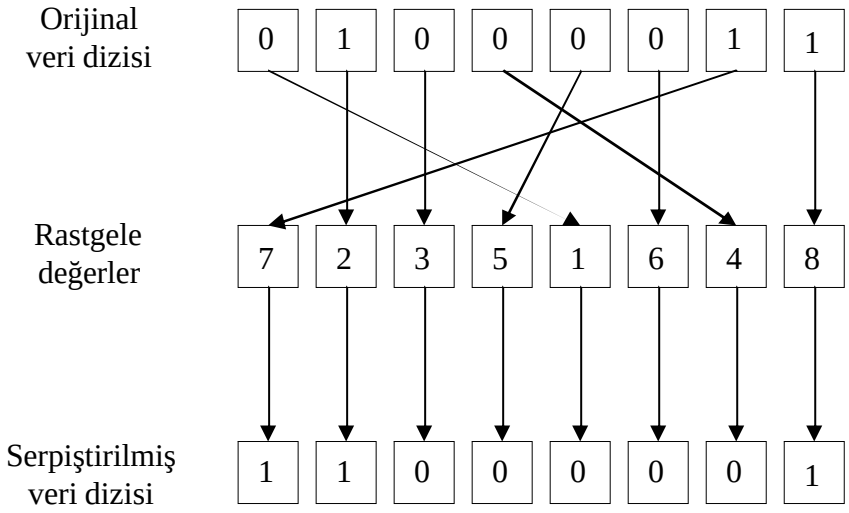
En çok kullanılan basit serpiştirici yöntemidir. Bu yöntemde $a \times b$ boyutlu bir giriş dizisi, serpiştirme için kullanılan $a \times b$ boyutlu bir matrise satır satır yazılıp sütun sütun okunmaktadır. Böylelikle, hata patlaması olması durumunda hatalı gelen bitler serpiştirici vasıtasıyla farklı yerlere serpiştirilmektedir. Şekil 9'da, grup halinde gelen dört tane hatalı bitin blok serpiştirici yardımıyla nasıl dağıtıldığı gösterilmektedir.



Şekil 9. (6×5) blok serpiştirici

1.9.2 Rastgele Serpiştirici (*pseudo-random interleaver*)

Rastgele serpiştirici, orijinal veri dizisini belli bir kural çerçevesinde düzensiz olarak dağıtmaktadır.



Şekil 10. Rastgele serpiştirici

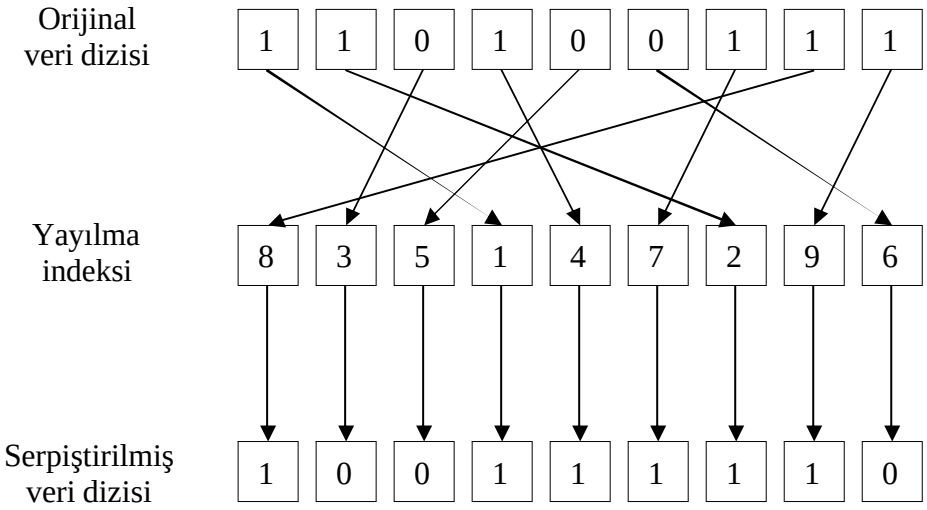
Şekil 10'dan da görüldüğü gibi, rastgele serpiştirici kullanılarak orijinal veri dizisinden rastgele i. sıradaki veri alınarak, serpiştirilmiş veri dizisinde j. sıraya denk düşen yere yerleştirilmektedir.

1.9.3. Yarı-rastgele Serpiştirici (*semi-random interleaver*)

Yarı-rastgele serpiştirici, rastgele serpiştiriciye iyileştirme uygulanarak elde edilmektedir. Bu yöntemde, giriş dizisindeki bütün komşu bitler en az çerçeve uzunluğu L ile ilişkili olan S_r faktörü kadar birbirinden uzaklaştırılmaktadır.

$$S_r < \sqrt{\frac{L}{2}} \tag{1.11}$$

Bu serpiştirici yönteminde, çerçeve boyutundan küçük olması şartıyla daha önce seçilmemiş bir sayı rastgele seçilmekte ve bir önceki rastgele seçilmiş sayı ile karşılaştırılmaktadır. Eğer rastgele seçilen sayı, daha önce belirlenmiş olan sayının $\pm S_r$ aralığında ise tekrar yeni bir sayı seçilmektedir. Aksi takdirde, seçilen sayı hafızada tutulmaktadır. Bu işlemler, çerçeve boyutundan küçük tüm sayılar bir kere seçilene dek devam etmektedir. Şekil 11'de $L = 9$ ve $S_r = 2$ olan bir yarı rastgele serpiştirici örneği verilmiştir.



Şekil 11. Yarı-rastgele serpiştirici

Bu noktada diagonal serpiştiriciden de biraz bahsetmek gerekmektedir. Diagonal serpiştirici, esas olarak, yardımcı bir serpiştirici olarak düşünülebilir. Diagonal serpiştirici, farklı bloklardaki bitlerin birbirleri içine serpiştirilmesini sağlar. Örneğin bilgi dizimiz 256 bitlik bloklar halinde ise, diagonal serpiştirici, öncül bloğun son 128 biti ile ikinci bloğun ilk 128 bitini birbiri içine serpiştirir. Bu işlem ikinci blok ile üçüncü blok ve daha sonra gelen bilgi blokları arasında sürdürülür. Buradaki amaç, bayılmaya uğrayan bir blokta oluşabilecek ardışıl hataların dağıtılmasıdır. Ardışıl hataların dağıtılması Reed – Solomon kod çözücünün başarımını arttıracaktır. Bu serpiştirici GSM'de uygulanan kodlama sisteminde de kullanılmaktadır.

2. YAPILAN ÇALIŞMALAR

2.1. Doğrusal Blok Kodlar

Önceki bölümlerde kanal kodlayıcının, bilgi kaynağından gelen dizilere semboller ekleyerek koda hata algılama ve düzeltme yeteneği kazandırdığından bahsedilmişti. Bilgi kaynağı çıkışının "0", "1" şeklinde ikili sayılar olduğunu varsayalım. Bu ikili bilgi dizisi k bitlik mesaj bloklarına ayrılsın. Mesaj blokları u ile gösterilirse her u blok toplam k bilgi biti içerir ve k bitlik toplam 2^k mesaj bulunabilir. Kanal kodlayıcı belli matematiksel kurallarla u mesajına kontrol bitleri ekleyerek n bitlik bir sözcük oluşturur. Bu n bitlik sözcük, u mesajına ait v kod sözcüğüdür. Her bir u sözcüğüne karşı düşen bir v kod sözcüğü vardır. Bu nedenle 2^k tane kod sözcüğü (n , k) blok kodunu oluşturur.

Tanım 2.1. : n uzunluklu 2^k kod sözcüğü, eğer n uzunluklu vektör uzayının k boyutlu bir alt uzayı ise (n , k) doğrusal blok kodunu oluşturur. Doğrusallıktan kasıt kod vektör uzayının, temel vektörlerin doğrusal kombinezonlarıyla üretilebileceğidir. Eğer doğrusal bağımsız n bitlik k vektör $g_1 , g_2 , g_3 , ….. , g_k$ ise, bunların doğrusal kombinezonlarıyla (n , k) blok kodu oluşturulabilir. u_i bilgi bitleri, olmak üzere v kod sözcüğü 2.1 bağıntısı ile bulunabilir.

$$v = (u_1, u_2, ...u_k) \begin{bmatrix} g_1 \\ g_2 \\ . \\ . \\ g_k \end{bmatrix} = u.G \qquad (2.1)$$

2^k bilgi dizisi 2.1'deki yerine konursa 2^k kod sözcük, dolayısıyla (n , k) elde edilir.G matrisinin satırları blok kodu belirlediğinden G matrisine üreteç matris denir. Örneğin üreteç matris,

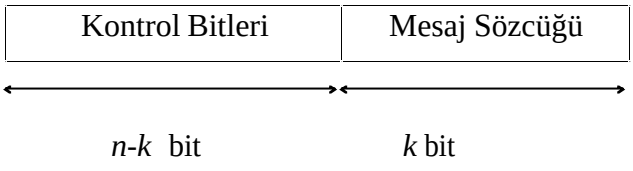
$$G = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

olan (7 , 4) doğrusal blok kodu Tablo 5'de verilmiştir.

Tablo 5. (7 , 4) doğrusal blok kodu

Mesaj				**Kod Sözcüğü**						
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	1	0	1	0	0
1	1	0	0	1	0	1	1	1	0	0
0	0	1	0	1	1	1	0	0	1	0
1	0	1	0	0	0	1	1	0	1	0
0	1	1	0	1	0	0	0	1	1	0
1	1	1	0	0	1	0	1	1	1	0
0	0	0	1	1	0	1	0	0	0	1
1	0	0	1	0	1	1	1	0	0	1
0	1	0	1	1	1	0	0	1	0	1
1	1	0	1	0	0	0	1	1	0	1
0	0	1	1	0	1	0	0	0	1	1
1	0	1	1	1	0	0	1	0	1	1
0	1	1	1	0	0	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1

(n , k) kodu, gerçekte bir vektör uzayı olduğundan, 2^k kod vektörü vektörel toplama işlemi üzerinde bir grup oluşturur. Grup tanımından yola çıkarak herhangi iki kod sözcüğünün toplamının da bir kod sözcük olacağı ve sıfır vektörünün de bir kod sözcüğü olduğu söylenebilir. Doğrusal blok kodlar Şekil 12'deki gibi sistematik hale sokulabilirler. Sistematik haldeki blok kodun *k* bitini mesaj, geriye kalan *n-k* bitini ise kontrol bitleri oluşturur.



Şekil 12. Kod sözcüğünün sistematik hali

Sistematik haldeki *(n , k)* kodunun üreteç matrisi 2.2'de verilmiştir.

$$G=\begin{bmatrix} g_0 \\ g_1 \\ . \\ . \\ g_k \end{bmatrix}=\begin{bmatrix} p_{00} & p_{01} & . & p_{0,n-k-1} & 1 & 0 & . & 0 \\ p_{10} & . & . & p_{1,n-k-1} & 0 & 1 & . & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ p_{k-1,0} & p_{k-1,1} & . & p_{k-1,n-k-1} & 0 & 0 & . & 1 \end{bmatrix} \qquad (2.2)$$

P I

2.2'de I_k , k x k boyutlu birim matris olup $G=[P \quad I_k]$'dır. 2.1'den yola çıkarak *v* kod sözcüğü hesaplanabilir.

$$v = (v_0, v_1,\ldots\ldots,v_{n-1}) = (u_0, u_1,\ldots\ldots,u_{k-1}) .G \qquad (2.3)$$

2.2. ve 2.3. kullanılarak kod sözcüğün elemanları,

$$v_{n-k+i} = u_i, \quad 0 \le i \le k \qquad (2.4a)$$

$$v_i = u_0 p_{0j} + u_1 p_{1j} + ... + u_{k-1} p_{k-1,j} \qquad 0 \le j \le n-k \qquad (2.4b)$$

olarak bulunur. 2.4a'dan, kod sözcüğün sağında kalan *k* bitin bilgi dizisinin aynısı olduğu, 2.4b'den ise *n-k* kontrol bitinin bilgi bitlerinin doğrusal bir işlevi olduğu görülebilir. 2.4b ile verilen *n-k* eşitlik parite kontrol denklemleri olarak adlandırılır. Parite kontrol denklemleri ayrıca *H* parite kontrol matrisi ile de ifade edilebilir. Sistematik haldeki *(n , k)* kodunun parite kontrol matrisi 2.5 ile bulunabilir.

$$H=[I_{n-k} \quad P^T]=\begin{bmatrix} 1 & 0 & 0 & . & 0 & p_{00} & p_{10} & . & p_{k-1,0} \\ 0 & 1 & 0 & . & 0 & p_{01} & p_{11} & . & p_{k-1,1} \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & . & 0 & 1 & p_{0,n-k-1} & p_{1,n-k-1} & . & p_{k-1,n-k-1} \end{bmatrix} \qquad (2.5)$$

G üreteç matrisi ile elde edilen *v kodu,*

$v.H^T = 0$ koşulunu sağlar. (2.6)

Ayrıca 2.3. ve 2.6.'dan

$v_j + u_0 p_{0j} + u_1 p_{1j} + ... + u_{k-1} p_{k-1,j} = 0$ elde edilebilir. (2.7)

2.7 tekrar düzenlenirse 2.4b ile verilen parite kontrol denklemi bulunabilir. Bu nedenle *(n , k)* doğrusal kodu parite kontrol matrisi ile tanımlanır. Parite kontrol matrisi alınan kod sözcüğünün hatalı iletilip iletilmediğini algılamakta kullanılır. Örneğin *v* gürültülü bir kanaldan iletilen kod sözcük, *e* iletim kanalında *v*'ye etkiyen gürültü olmak üzere, alıcı tarafa ulaşan sözcük, *r*

$$r = v + e \tag{2.8}$$

ile verilir. *e*'nin sıfırdan farklı olduğu yerlerde bir hata oluşmuş ve iletilen sözcük bozulmuştur. Alıcı taraf *v* ve *e* sözcüklerini bilmediğinden kod çözücü öncelikle *r*'de hata olup olmadığına karar vermelidir. Eğer hata olmuşsa kod çözücü ya hatayı düzeltir ya da gönderene tekrar iletim isteği (ARQ) gönderir *r* sözcüğü alındığında kod çözücü 2.9 ile verilen *n-k* tane hata belirtecini hesaplar.

$$S = r.H^T = (s_0, s_1, \ldots\ldots\ldots s_{n-k-1}) \tag{2.9}$$

r bir kod sözcük ise *S=0* olur. $S \neq 0$ ise *r* bir kod sözcük değildir ve hata algılanır. Ancak hata sözcüğü sıfırdan farklı bir kod sözcüğe eşit ise, herhangi iki kod sözcüğün toplamı da bir kod sözcük olacağından bu hata dizisi kod çözücü de algılanamaz. Toplam 2^k-1 tane algılanamaz hata vardır.

2.8 eşitliği 2.9'da kullanılırsa hata belirteci

$$S = r.H^T = (v + e).H^T = v.H^T + e.H^T \tag{2.10}$$

olur. 2.6 ve 2.10'dan hata belirtecinin sadece hata dizisine bağlı olduğu görülebilir.

$$S = e.H^T \tag{2.11}$$

Parite kontrol matrisinin sistematik hali 2.5 ve 2.11'den hata belirteci ve hata bitleri arasında bir ilişki kurulabilir.

$$
\begin{aligned}
s_0 &= e_0 + e_{n-k}\,p_{00} + e_{n-k+1}\,p_{10} + \ldots\ldots\ldots + e_{n-1}\,p_{k-1,0} \\
s_1 &= e_1 + e_{n-k}\,p_{01} + e_{n-k+1}\,p_{11} + \ldots\ldots\ldots + e_{n-1}\,p_{k-1,0} \\
&\ldots \\
&\ldots \\
s_{n-k-1} &= e_{n-k-1} + e_{n-k}\,p_{0,n-k-1} + e_{n-k+1}\,p_{1,n-k-1} + \ldots\ldots\ldots + e_{n-1}\,p_{k-1,n-k-1}
\end{aligned}
\qquad (2.12)
$$

Hata belirtecinin elemanları, hata bitlerinin bir doğrusal kombinezonudur ve hatalı bitler hakkında bilgi taşırlar. Bu nedenle hata belirteci hata düzeltme işleminde kullanılabilir. 2.12'deki *n-k* eşitliğin çözümünü veren herhangi bir yöntem hata düzeltme yöntemidir. Hata dizisi *e*, bulunduktan sonra 2.8 yardımıyla *r+e* dizisi gerçekte iletilen kod sözcüğü *v'* yi verir. Fakat aynı hata belirtecine neden olacak 2^k hata dizisi vardır. Hatalı kod çözme olasılığını azaltmak için, kod çözücü 2^k hata dizisinden gerçek hata dizisi olma olasılığı en yüksek olanını seçmelidir. Örneğin ikili simetrik kanalda gerçek hata dizisi, tüm hata dizileri içinde Hamming ağırlığı en az olan hata dizisidir.

Bir *v* kod sözcüğünün Hamming ağırlığı, kod sözcüğünde yer alan sıfırdan farklı bitlerin sayısıdır ve *w*(*v*) ile gösterilir. Herhangi iki kod sözcük *v, w*'nin birbirlerinden farklı oldukları bit sayısı ise Hamming uzaklığıdır ve d (*v,w*) ile gösterilir.

Örneğin, *v* = (1001), *w* = (0011) için *w(v) = w(w) = 2, d(v , w) = 2*'dir. İki kod sözcüğün Hamming uzaklığı toplamlarının Hamming ağırlığına eşittir.

$$d(v , w) = w(v + w) \qquad (2.13)$$

İki kod sözcüğün toplamı da bir kod sözcük olduğundan bir *C* kodunun minimum uzaklığı d_{min}, sıfırdan farklı kod sözcüklerin en küçük Hamming ağırlığı kadardır.

$$d_{min} = min\,(w(x) : x \varepsilon C, x \neq 0) = w_{min} \qquad (2.14)$$

v kod sözcüğü gürültülü bir kanaldan iletimi sırasında I hata oluşursa, alınan *r* kod sözcüğü, gönderilen *v* sözcüğünden I yerde farklıdır. (*d(v,r)=I)*. Eğer *C* kodunun minimum uzaklığı d_{min} ise herhangi iki kod sözcük d_{min} yerde farklıdır. Hata sözcüğü ancak bir kod sözcüğüne eşit ise kod çözme hatası olacağından ve d_{min} -1 veya daha az ağırlıklı hata dizileri kod sözcük olamayacağından, *C* kodu bu hata dizilerini algılayabilir. 2^k mesaj sözcüğüne karşı düşen 2^k kod sözcüğü vardır. *n* bitlik sözcük sayısı 2^n ve kod sözcüğü sayısı 2^k olduğunda *C* kodu toplam 2^n-2^k hata dizisini algılayabilir.

Eğer hata dizisi bir kod sözcüğe eşit ise yine bir kod sözcük oluşturacağından bu tip hatalar algılanamaz. Sıfırdan farklı kod sözcüğü sayısı 2^{k-1} tane algılanamaz hata vardır. Büyük n değerleri için 2^{k-1}, 2^n yanında çok küçük kalacağından az miktarda hatalı dizi algılanmadan kod çözücüden geçer. d_{min} uzaklığına sahip C kodu $d_{\min} \geq 2t+1$ olmak üzere t veya daha az hatalı sözcükleri kesinlikle düzeltebilir. t, rastgele hata düzeltim kapasitesi ve C koduna t hata düzelten kod denir. Örneğin, Tablo 5'de verilen (7 , 4) kodunun minimum uzaklığı 3 dolayısıyla $t=1$ dir. Bu kod 1 hatalı dizileri düzeltebilir.Görüldüğü gibi bir kodun rastgele hata algılama ve düzeltme özellikleri o kodun minimum uzaklığa sahip blok kodu kullanmak hatalı kod çözme olasılığını azaltır.

Doğrusal blok kodlamanın önemli bir alt grubu Hamming kodlardır. m = 3 olan herhangi bir tamsayı için aşağıdaki parametrelerle tanımlı bir Hamming kodu tanımlanabilir.

Kod uzunluğu : $n = 2^m - 1$

Bilgi bit sayısı : $k = 2^m - m - 1$

Parite kontrol bit sayısı : $n - k = m$

Minimum uzaklık : $d_{\min} = 3$

Hamming kodları $t=1$ hata düzeltebilirler.

2.2. Bose – Chaudri – Hocquenghem Kodları

Bose, Chaudri ve Hocquenghem (BCH) kodları rastgele hata düzelten kodlar içinde önemli bir gruptur. BCH kodları, Hamming kodlarının birden fazla hata düzelten, genelleştirilmiş hali olarak düşünülebilir. BCH kodları 1959 yılında Hocquenghem, 1960 yılında Bose ve Chaudri tarafından bulunmuştur[44]. Kodların çevrimsellik özelliği 1960 yılında Peterson tarafından kanıtlanmış ve Gorenstein ve Zierler, BCH kodlarının p^m sembollerinde genelleştirilmiş halini elde etmiştir[42].

BCH kodları ikili ve ikili olmayan kodlar olmak üzere iki gruba ayrılır. İkili BCH kodlarının üreteç polinomlarının katsayıları GF(2)'nin, ikili olmayan kodların üreteç polinomunun katsayıları p bir asal sayı olmak üzere GF(p^m)'nin elemanıdır. Yani ikili

olmayan kodlama m bitlik bilgi sembolleri üzerinde yapılır.

Reed – Solomon kodları ikili olmayan BCH kodlara örnektir.

$m = 3$ ve $t < 2^{m-1}$ olmak üzere ikili BCH kodunun parametreleri aşağıdaki gibidir.

Blok uzunluğu : $n = 2^m - 1$

Parite kontrol bit sayısı : $n - k \le m.t$

Minimum uzaklık : $d_{\min} \ge 2.t + 1$

Parametreleri yukarıda verilen BCH kodu *n* bitlik bir blokta, *t* veya daha az sayıda hatayı düzeltebilir. Bu nedenle bu kod *t* hata düzelten BCH kodu olarak adlandırılır. Bu kodun üreteç polinomu g(x)'in katsayıları GF(2)'nin elemanı olduğu halde, kökleri $GF(2^m)$'in elemanıdır.$\alpha, GF(2^m)$'de bir asal eleman olmak üzere *t* hata düzelten, *n* uzunluklu BCH kodunun üreteç polinomu, kökleri $\alpha, \alpha^2, \alpha^3 .....\alpha^{2t}$ olan en düşük dereceli polinomdur[43].

$$g(x) = OKEK\{\Phi_1(x), \Phi_3(x), ...\Phi_{2t-1}(x)\} \quad (2.15)$$

u(x) bilgi dizisine karşı düşen kod sözcük *v(x)* 2.16 ile bellidir.

$$v(x) = u(x).g(x) \quad (2.16)$$

Kodun tanımı gereği parite kontrol bitlerinin sayısı en fazla *m.t* kadardır. Parite kontrol bit sayısı ile kodun düzeltebildiği hata sayısı arasında tam olarak belli bir bağıntı olmamakla beraber küçük *t* değerleri için *m.t* tam olarak *n-k* değerine eşit olur. Örneğin tek hata düzelten BCH kodu için *t=1* olduğundan $g(x)=\Phi_1(x)$ olur. Bu kodun parametreleri,

Blok uzunluğu : $n=2^m-1$

Parite kontrol bit sayısı : $n-k= m$

Minimum uzaklık : $d_{min}=3$

olduğundan tek hata düzelten BCH kodu aynı zamanda 2^m-1 uzunluklu Hamming kodudur. α , Tablo 3'te verilen $GF(2^4)$'ün elemanı ise α ve α^3 için

$$\Phi_1(X) = 1 + X + X^4$$

$$\Phi_3(X) = 1 + X + X^2 + X^3 + X^4$$

olarak verilir. 2.15'ten

$$g(X) = \Phi_1(X).\Phi_3(X) = 1 + X^4 + X^6 + X^7 + X^8$$

bulunur. Bu kod (15 , 7) BCH kodu olup bu kod için $n\text{-}k = m.t$ ve $d_{min} = 5$ 'dir. Bu nedenle (15 , 7) BCH kodu tam olarak 2 hata düzeltebilir.

İkili BCH kodların çözümünde parite kontrol matrisi yardımıyla hata belirteci hesabı gerekir. Hata belirtecinin hesabının ardından hata yerinin hesaplanması için Peterson yöntemi, Berlekamp'ın iteratif algoritması veya Chien metodu kullanılabilir. t hata düzelten BCH kodunun parite kontrol matrisi 2.17'de verilmiştir. Görüldüğü gibi H matrisinin elemanları GF(2^m)'den alınmıştır. Eğer her eleman m bitlik sözcüklerle ifade edilirse ikili parite kontrol matrisi elde edilir.

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & . & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & . & (\alpha^2)^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & . & (\alpha^3)^{n-1} \\ . & . & . & . & . \\ . & . & . & . & . \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & . & (\alpha^{2t})^{n-1} \end{bmatrix} \qquad (2.17)$$

Doğrusal blok kodlarda olduğu gibi BCH kodlarında da hata belirteci sadece hata sözcüğünün bir fonksiyonudur. Bu nedenle bulunan hata belirteci, alınan sözcüğün hatalı bitleri hakkında bilgi taşır. t hata düzelten BCH kodu için $2t$ hata belirteci,

$$S_i = (S_1, S_2, ...S_{2T}) = r.H^T \qquad (2.18)$$

olur. Bu eşitlikte r, doğrusal blok kodları için geçerli olan 2.8'den hesaplanabilir. 2.17 ve 2.18'den hata belirteci bulunabilir.

$$S_i = r(\alpha^i) = r_0 + r_1\alpha^i + r_2\alpha^{2i} + ... + r_{n-1}\alpha^{(n-1)i} \qquad (2.19)$$

2.16'ya göre kod polinomunun çarpanlarından biri üreteç polinomudur.

Üreteç polinomun kökleri de $\alpha, \alpha^2, \alpha^3 ..... \alpha^{2t}$ olduğundan $v(\alpha^i) = 0$ olmalıdır, iletim ortamında hata oluşmamışsa 2.19 sonucu hata belirteçleri sıfıra eşittir. Buna göre hata belirteçleri hata sözcüğünün bir işlevidir.

$$S_i = e(\alpha^i) \tag{2.20}$$

Hata sözcüğü $e(x), X^{j_1}, X^{j_2}, ... X^{j_v}$ olmak üzere toplam v yerde hata üretmişse $0 \leq j_i \leq n$ olmak üzere,

$$e(X) = X^{J_1} + X^{J_2} + ... + X^{J_V} \tag{2.21}$$

olur. Böylece 2.20 ve 2.21'den hata belirteçleri için *2t* eşitlik elde edilir.

$$\begin{aligned} S_1 &= \alpha^{j_1} + \alpha^{j_2} + ... + \alpha^{j_V} \\ S_2 &= (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + ... + (\alpha^{j_V})^2 \\ &.. \\ &.. \\ S_{2t} &= (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + ... + (\alpha^{j_v})^{2t} \end{aligned} \tag{2.22}$$

Bu denklemlerde $\alpha^{j_1}, \alpha^{j_2}, ... \alpha^{j_V}$ bilinmeyenlerdir ve bu eşitlikleri çözecek her yöntem BCH kodlar için kod çözme algoritmasıdır. $\alpha^{j_1}, \alpha^{j_2}, ... \alpha^{j_V}$ bulunduğunda j_1 hatalı sembollerin yerini belirler. 2.22'nin birden fazla çözümü vardır. En az sayıda hata içeren hata dizisini veren çözüm en uygun çözümdür. Büyük t değerleri için 2.22'nin doğrudan çözümü oldukça zordur. *l = 1,2,...,v* olmak üzere,

$$\beta_l = \alpha^{j_1} \tag{2.23}$$

ile tanımlanan β_l hata yerlerini belirtir. 2.23, 2.22'de kullanılırsa

$$\begin{aligned} S_1 &= \beta_1 + \beta_2 + ... + \beta_v \\ S_2 &= (\beta_1)^2 + (\beta_2)^2 + ... + (\beta_v)^2 \\ &.. \\ &.. \\ S_{2t} &= (\beta_1)^{2t} + (\beta_2)^{2t} + ... + (\beta_v)^{2t} \end{aligned} \tag{2.24}$$

bulunur. Hata yerleri β 'ların, S_i 'lerden bulunması için kullanılabilecek yöntemlerden ikisi **Peterson'un Doğrudan Çözüm Yöntemi** ve **Berlekamp'ın İteratif Yöntemi**'dir.

2.2.1. Peterson'un Doğrudan Çözüm Yöntemi

BCH kodlarının Peterson yöntemiyle çözümüne 2.25 ile verilen $\sigma(X)$, hata yeri polinomu tanımlansın.

$$\sigma(X) = (X + \beta_1)(X + \beta_2).....(X + \beta_L)$$
$$= X^L + \sigma_1 X^{L-1} + ...\sigma_l \qquad (2.25)$$

$\sigma(X)$'in kökleri $\beta_1, \beta_2.....\beta_v$ sayıları, hata yerleridir. Bu nedenle $\sigma(X)$, hata yeri polinomu olarak adlandırılır. $\sigma(X)$'in katsayıları ve hata yerleri arasında aşağıdaki ilişki vardır.

$$\sigma_1 = \sum_i \beta_i$$
$$\sigma_2 = \sum_{i<j} \beta_i \beta_j$$
$$\sigma_3 = \sum_{i<j<k} \beta_i \beta_j \beta_k$$
.. (2.26)
..
..
$$\sigma_l = \beta_1.\beta_{2.}\beta_2.....\beta_l$$

2.24'den ve 2.26'dan σ_i'lerin hata belirteçlerine bağlı olduğu görülebilir. Hata yeri $\sigma(X)$'in kökü olduğundan $\sigma(\beta_i) = 0$'dır.

$$\sigma(\beta_i) = \beta_i^{\,t} + \sigma_1 \beta_i^{\,t-1} + ...\sigma_t = 0 \qquad i = 1,2.....t \qquad (2.27)$$

2.27'nin her iki tarafını $\beta_i^{\,j}$ ile çarpılsın.

$$\beta_i^{\,t+j} + \sigma_1 \beta_i^{\,t+j-1} + ...\sigma_t \beta_i^{\,j} = 0 \qquad (2.28)$$

j'yi sabit tutup $i = 1,2.....t$ için 2 28'den elde edilecek *t* eşitliği toplarsak 2.24. yardımıyla hata belirteçleri ve hata yeri polinomunun katsayıları arasında bir ilişki buluruz.

$$S_{t+j} + \sigma_1 S_{t+j-1} + ... + \sigma_t S_j \qquad (2.29)$$

2.29'la verilen eşitlikler Newton Özdeşlikleri olarak adlandırılır. İkili BCH kodu için σ_i '1er, S_i'lere aşağıdaki Newton özdeşlikleri ile bağlıdır.

$$
\begin{aligned}
&S_1 + \sigma_1 = 0 \\
&S_2 + \sigma_1 S_1 + 2\sigma_2 = 0 \\
&S_3 + S_2\sigma_1 + S_1\sigma_2 + 3\sigma_3 = 0 \\
&.. \qquad\qquad i\sigma_i = \begin{cases} \sigma_i & ,i \ \ tek \\ 0 & ,i \ \ çift \end{cases} \\
&.. \\
&S_{t+j-1} + \sigma_1 S_{t+j-2} + ... + t\sigma_t = 0 \\
&S_{t+j} + \sigma_1 S_{t+j-1} + ... + \sigma_t S_j = 0
\end{aligned}
\tag{2.30}
$$

BCH kodlarının çözümü için 2.30 yardımıyla hata yeri polinomu bulunur. Hata yeri polinomunun kökleri hata yerlerini verir. Örneğin *t=1* hata düzelten BCH kodu için $\sigma_2 = 0$ olacağından ilk iki özdeşliğin çözümü σ_1'i verir.

$$S_1 + \sigma_1 = 0 \Rightarrow \sigma_1 = S_1$$

$$\sigma(X) = X + S_1$$

olur. Yani hata yeri $\beta_1 = S_1$'dir. Örneğin,$S_1=\alpha^3$ ise alınan sözcüğün 3. biti hatalıdır demektir ve bu bit evirilerek hata düzeltilebilir.

t=2 hata düzelten BCH kodunun çözümü için $\sigma_3 = 0$ olacağı dikkate alınıp 2.30'daki 1. ve 3. eşitlikler matris formunda yazılırsa,

$$\begin{bmatrix} 1 & 0 \\ S_2 & S_1 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_3 \end{bmatrix} \tag{2.31}$$

Bu iki eşitlik çözülürse hata yeri polinomu $\sigma(X)$'in katsayıları,

$$\sigma_1 = S_1 \quad \text{ve} \quad \sigma_2 = \frac{S_3 + S_1^{\,3}}{S_1} \tag{2.32}$$

olarak bulunur.

$\sigma(X)$'in kökleri bulunarak hata yerleri belirlenir ve hatalı bitler düzeltilir.

2.2.2. Berlekamp Yöntemi

Peterson'un doğrudan çözüm yöntemiyle az sayıda hata düzelten ikili BCH kodları çözmek kolaydır. Altıdan fazla hata düzelten kodların çözümünde $\sigma(X)$'in katsayılarının hesaplanması oldukça karmaşık işlemler gerektirir. Bu kodların çözümü için Berlekamp iteratif bir yöntem geliştirmiştir. Daha sonra Massey, bu yöntemi bir ötelemeli yazıcı sentezine indirgemiştir[45].

Berlekamp yönteminde hata belirteçleri aşağıdaki gibi bir polinomla ifade edilir.

$$S(X) = S_1 X + S_2 X^2 + ... + S_{2t} X^{2t} \tag{2.33}$$

Yöntemin amacı, kökleri hata yerlerinin tersi olan bir *C(X)* polinomu bulmaktır. β_i hata yerleri ve $i = 1.2.....t$ olmak üzere *C(X)*,

$$C(X) = (1+\beta_1 X)(1+\beta_2 X).....(1+\beta_t X) \tag{2.34}$$

ile verilir. 2.34 açık biçimde yazılırsa hata yeri polinomu $\sigma(X)$ ile *C(X)* arasındaki ilişki görülebilir.

$$C(X) = 1+\sigma_1 X + \sigma_2 X^2 + ... + \sigma_i X^t \tag{2.35}$$

Yöntemin ilk adımı katsayıları birinci Newton özdeşliğini sağlayan minimum dereceli $C^1(X)$ polinomunu bulmaktır. İkinci adımda $C^1(X)$'in katsayılarının ikinci Newton özdeşliğini de sağlayıp sağlamadığı kontrol edilir. Katsayılar ikinci Newton özdeşliğini sağlıyorsa,

$$C^2(X) = C^1(X)$$

olur. Eğer katsayılar ikinci özdeşliği sağlamıyorsa $C^1(X)$'e, ilk iki özdeşliği sağlayacak ve minimum dereceli polinomu verecek bir terim eklenir. Böylece ikinci adımda ilk iki Newton özdeşliğini sağlayan $C^2(X)$ polinomu elde edilir. Bir sonraki adımda aynı yöntemle, $C^2(X)$'den, ilk üç Newton özdeşliğini sağlayan $C^3(X)$ elde edilmeye çalışılır. Iterasyon bütün Newton özdeşliklerini sağlayan $C^{2t}(X)$ polinomu elde edilene kadar devam eder. Sonunda $C(X) = C^{2t}(X)$ olur.

C(X), 2.22'yi sağlayan minimum ağırlıklı hata dizisi *e(X)*'i verir. Alınan kod sözcüğü *r(X)*'deki hata sayısı *t* veya daha az ise *C(X)* gerçek hata dizisini verir.

$$C^{(\mu)}(X) = 1 + \sigma_1^{(\mu)} X + \sigma_2^{(\mu)} X^2 + ... + \sigma_{l_\mu}^{(\mu)} X^{l_\mu} \quad (2.36)$$

$C^{(\mu)}(X)$, μ adımda elde edilen polinomdur ve derecesi l_μ 'dür. $C^{(\mu+1)}(X)$'i belirlemek için μ .uzaklık, d_μ hesaplanır.

$$d_\mu = S_{\mu+1} + \sigma_1^{(\mu)} S_\mu + \sigma_2^{(\mu)} S_{\mu-1} + ... + \sigma_{l_\mu}^{(\mu)} S_{\mu+1-l_\mu} \quad (2.37)$$

$d_\mu = 0$ ise $C^{(\mu)}(X)$'in katsayıları $(\mu+1)$. Newton özdeşliğini de sağlar ve

$C^{(\mu+1)}(X) = C^{(\mu)}(X)$

olur. $d_\mu \neq 0$ ise $C^{(\mu)}(X)$'in katsayıları $(\mu+1)$. Newton özdeşliğini sağlamaz ve $C^{(\mu)}(X)$'de düzeltme yapılmalıdır. Düzeltme terimini belirlemek için $d_p \neq 0$ ve $p - l_p$ değeri en yüksek olan *p*.adımdaki $C^{(p)}(X)$ polinomu belirlenir. $(\mu+1)$. Newton özdeşliğini sağlayan polinom

$$C^{(\mu+1)}(X) = C^{(\mu)}(X) + d_\mu d_p^{-1} X^{u-p} C^{(p)}(X) \quad (2.38)$$

ile verilir.

Tablo 6**.** Berlekamp'ın iteratif algoritması

μ	$C^{(\mu)}(X)$	d_μ	l_μ	$\mu - l_\mu$
-1	1	1	0	-1
0	1	S_1	0	0
1				
..				
..				
2t				

Hata yeri polinomunun bulunabilmesi için Tablo 6'daki başlangıç koşullarından itibaren 2.36, 2.37 ve 2.38 ile verilen kurallara uyularak, iteratif algoritma uygulanır. Tablo 6'nın son satırında elde edilecek polinom hata yeri polinomunu verir.

Eğer bu polinomun derecesi *t*'den büyükse, alınan kod sözcükte *t*'den hazla hata var demektir ve bu hataların yerini belirlemek mümkün olamaz.

BCH kodlarının çözümündeki son adım hata yerlerinin bulunmasıdır. Hata yerleri, *C(X)* polinomunun köklerinin tersleridir. Kökler, GF(2^m)'in tüm elemanları, $1,\alpha,\alpha^2,....\alpha^{n-1}$, *C(X)'*de X yerine konarak bulunabilir. $C(\alpha^i)=0$ ise, α^i bir köktür ve tersi α^{n-i} bir hata yeridir. Alınan kod sözcüğünün r_{n-i} . biti hatalıdır. *C(X)* polinomunun köklerinin bulunması için GF(2^m)'in elemanlarının *C(X)* polinomuna yerleştirilmesi önce Peterson tarafından kullanılmış, daha sonra Chien bu işlemlerin gerçekleştirilmesi ve hatanın düzeltilmesi için bir prosedür geliştirmiştir[44-53]. Alınan kod sözcük,

$$r(X)=r_0+r_1X+r_2X^2+.....+r_{n-l}X^{n-l}$$

için kod çözümüne en yüksek anlamlı bitten başlanır. α^{n-l} , hata yerlerinden biriyse, r_{n-l} . biti hatalıdır ve α^{n-l}'in tersi α , *C{X)*'in köküdür. Bu durumda

$$1+\sigma_1\alpha+\sigma_2\alpha^2+.....+\sigma_v\alpha^v$$

olur. r_{n-l} hatalıdır. r_{n-l} 'nin hatalı olup olmadığını test etmek için

$$1+\sigma_1\alpha^l+\sigma_2\alpha^{2l}+.....+\sigma_v\alpha^{vl}$$

toplamının sıfıra eşit olup olmadığına bakılır. r_{n-l} hatalı alınmışsa, r_{n-l} **,** 1 ile ikili toplama işlemine sokulur ve böylece hata düzeltilmiş olur.

2.3. Reed – Solomon Kodları

2.3.1 Reed – Solomon Kodlarının Tanımı

Reed-Solomon (RS) kodları ikili olmayan BCH kodlarının bir alt kümesidir. RS kodları çevrimsel kodlardır. Reed ve Solomon tarafından 1960 yılında tanımlanmıştır[47]. RS kodları sabit *k* ve *n* değerleri için en yüksek minimum Hamming uzaklığına sahiptir.

Bölüm 2.2.'de anlatılan ikili BCH kodları, ikili olmayan kodlara genelleştirilebilir. *p* bir asal sayı, *q*, *p*'nin bir kuvveti, *s* ve *t* herhangi iki tamsayı olmak üzere, blok uzunluğu $n = q^s - 1$, parite kontrol sembolü sayısı, *n-k=2st* olan bir ***İkili Olmayan BCH Kodu*** vardır.

Bu kod, *2st* parite kontrol sembolü yardımıyla, *t* veya daha az sayıda hatalı sembolü düzeltme yeteneğine sahiptir. İkili BCH kodlarının üreteç polinomunun katsayıları GF(2)'nin elemanı iken, ikili olmayan BCH kodunun üreteç polinomunun katsayıları $GF(q^s)$'in elemanıdır. Üreteç polinomun kökleri ise $\alpha, \alpha^2, ... \alpha^{2t}$ dir. $\Phi_i(X)$ α^i'nin minimal polinomu olmak üzere üreteç polinom,

$$g(X) = OKEK\{\Phi_1(X), \Phi_2(X), ... \Phi_{2t}(X)\} \quad (3.1)$$

ile verilir. Her bir minimal polinomun derecesi *s* veya daha azdır. Bu nedenle $g(X)$'in derecesi *2st* olabilir. Dolayısıyla $g(X)$ tarafından üretilen kod *2st* parite kontrol sembolü içerir.

RS kodları, ikili olmayan BCH kodları içinde *s=1* alınarak elde edilir, *t* hata düzelten RS kodunun parametreleri,

Blok uzunluğu : $n = q - 1$

Parite kontrol bit sayısı : $n - k = 2t$

Minimum uzaklık : $d_{\min} = 2t + 1$

olarak verilir. RS kodlarının önemli bir özelliği minimum Hamming uzaklığının her zaman parite kontrol sembolü sayısından bir fazla olmasıdır. α^i $GF(q = p^m)$'in sıfırdan farklı bir elemanı ve $\Phi(X)$'in bir kökü olmak üzere *t* hata düzelten ilkel RS kodunun üreteç polinomu,

$$g(X) = (X + \alpha)(X + \alpha^2).....(X + \alpha^{2t})$$

$$= g_0 + g_1 X + g_2 X^2 + ... + g_{2t-1} X^{2t-1} + X^{2t} \quad (3.2)$$

ile verilir. g_i'ler $GF(q = p^m)$'nin elemanıdır ve $\alpha, \alpha^2, ...\alpha^{2t}$, $g(X)$'in kökleridir. $g(X)$'in derecesi en fazla *2t*'dir. $g(X)$ tarafından üretilen kod *(n , n-2t)* kodudur.

Örnek 3.1: Tablo 6'da verilen GF(2^4) üzerinde, üç hata düzelten (15 , 9) RS kodu için üreteç polinom,

$$g(X) = (X + \alpha)(X + \alpha^2)(X + \alpha^3)(X + \alpha^4)(X + \alpha^5)(X + \alpha^6)$$

$$= \alpha^6 + \alpha^9 X + \alpha^6 X^2 + \alpha^4 X^3 + \alpha^{14} X^4 + \alpha^{10} X^5 + X^6$$

kullanılır.

2.3.2 Kodlama

k=n-2t olmak üzere, kodlanacak bilgi sözcüğü,

$$i(X) = i_0 + i_1 X + i_2 X^2 + ... + i_{k-1} X^{k-1} \quad (3.3)$$

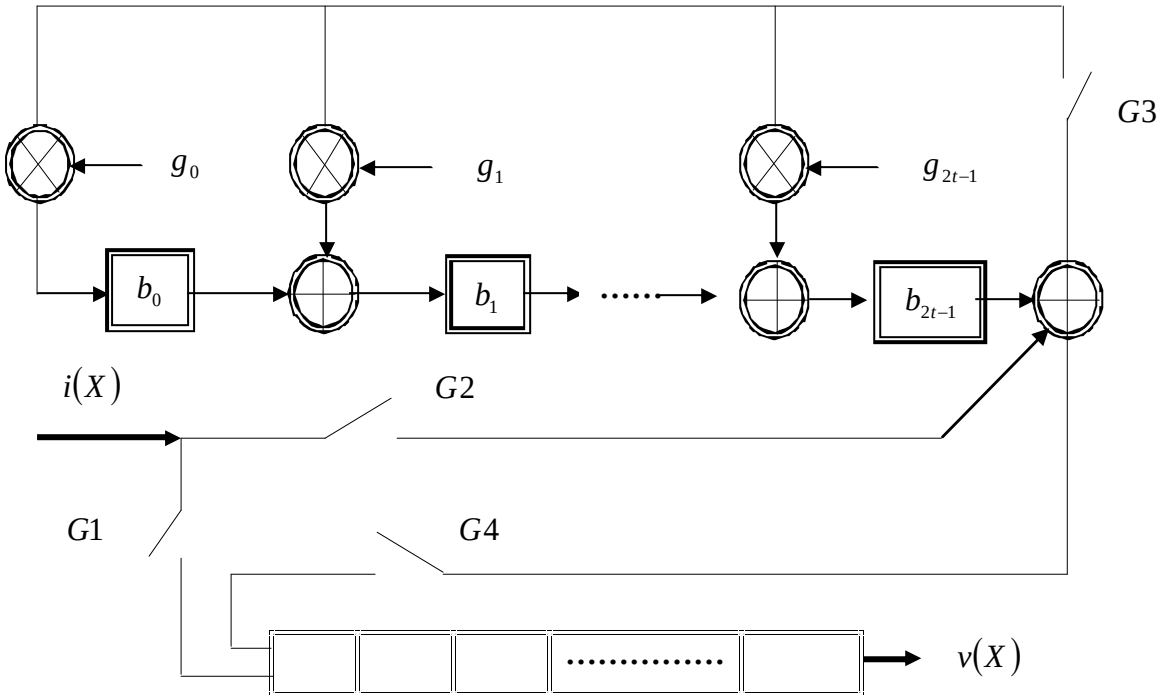
olsun. RS kod sözcükleri *k* sembollük bilgi sözcüğü *i(X)'*in *g(X)* ile çarpılması sonucu elde edilebilir; fakat elde edilen kod sözcük sistematik halde olmayacaktır. Sadece $GF(p^m)$ içinde *g(X)'*e tam olarak bölünen sözcükler RS kod sözcükleri olabileceği dikkate alınırsa kodlama, bir bölme işlemiyle gerçekleştirilebilir. *i(X)* polinomu, X^{2t} ile çarpılıp *g(X)'*e bölünsün. Bölümden kalan *b(X)*, $X^{2t}.i(X)$ ifadesine eklenirse elde edilecek polinom *g(X)'e* tam olarak bölünebilir ve bu polinomla ifade edilen kod sözcük *v(X)*sistematik haldedir.

$$b(X) = X^{2t} i(X) \bmod g(X)$$

$$= b_0 + b_1 X + ... + b_{2t-1} X^{2t-1} \quad (3.4)$$

$$v(X) = X^{2t} i(X) + b(X) \quad (3.5)$$

Bu işlemle *i(X) 2t* kadar ötelenip, *i(X)'*in *g{X)'*e bölümünden kalan eklenerek kod sözcük oluşturulur. Bu işlemi gerçekleştiren bir devrenin blok diyagramı Şekil 13. de verilmiştir.



Şekil 13. Reed – Solomon kodlayıcı devrenin blok diyagramı

Bu devrede *2t* adet yazıcı eleman bulunmaktadır. Toplama ve çarpma işlemleri *GF(p^m)* üzerinde yapılır. *i(X)* sözcüğünü kodlamak için önce *G1*.*G2* ve *G3* anahtarları kapanır, *G4* anahtarı açılır ve *i(X)* sözcüğü kanala verilir. *i(X)* bilgi sözcüğü iletim kanalına girdiği anda bölme işlemi bitmiş ve parite kontrol sembolleri yazıcılarda saklanmıştır. *G1*, *G2* ve *G3* anahtarları açılır ve *G4* anahtarı kapanır ve parite kontrol sembolleri kanala verilir. Tüm semboller kanala girdiğinde kodlama işlemi bitmiştir. Bütün yazıcılara sıfır değeri yüklenir ve devre yeni bilgi sözcüklerinin kodlanmasına hazır duruma getirilir.

2.3.3. Kod Çözme

Bu bölümde RS kodlarının çözümü için kullanılan yöntemler açıklanacaktır. Doğrusal blok kodlarında ve ikili BCH kodlarında kod çözme işlemine hata belirtecinin hesabı ile başlanır. Hata belirtecinin hesabının ardından hata yerleri bulunur ve hatalı bitler düzeltilir. İkili olmayan BCH kodlarında dolayısıyla RS kodlarında da kod çözme işleminin ilk adımı hata belirteci hesabıdır, fakat hata yerleri bulunduktan sonra ikili kodlardan farklı olarak hata değerlerinin de bulunması gerekir. Çünkü RS kodları bitler üzerinde değil bir kaç bitten oluşan sözcükler üzerinde çalışır. RS kodlarının çözümünde izlenecek adımlar aşağıdaki gibidir.

1. Hata belirteçlerinin hesabı
2. Hata yeri polinomu $\sigma(X)$'n bulunması
3. $\sigma(X)$'in kökleri olan hata yerlerinin bulunması
4. Hata değerlerinin hesaplanması
5. Hatalı sözcüklerin düzeltilmesi

Hata belirteçlerinin hesabında tek bir yöntem kullanılırken hata yeri polinomunun belirlenmesinde bir kaç yöntem kullanılabilir. Bu bölümde her bir yöntem örneklerle incelenecektir.

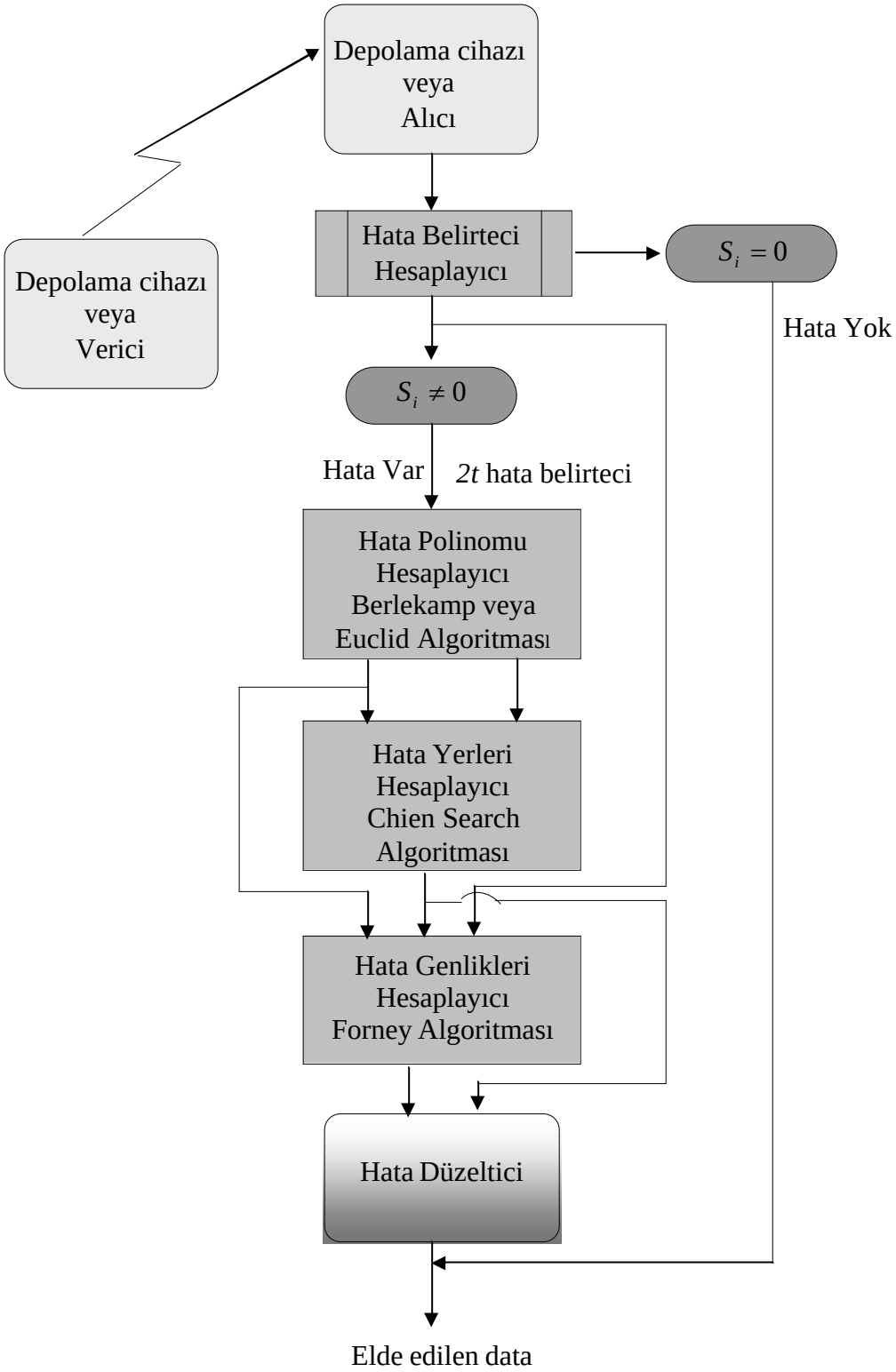
2.3.3.1. Hata Belirteci Hesabı

Gönderilen kod sözcük *v(X)* , alınan sözcük *r(X)* olmak üzere kanal tarafından üretilen gürültü *e(X),*

$$v(X) = v_0 + v_1 X + \ldots\ldots\ldots\ldots + v_{n-1} X^{n-1}$$
$$r(X) = r_0 + r_1 X + \ldots\ldots\ldots\ldots + r_{n-1} X^{n-1}$$
$$e(X) = r(X) + v(X) = e_0 + e_1 X + \ldots\ldots\ldots + e_{n-1} X^{n-1} \qquad (3.6)$$

olarak tanımlanır.

Şekil 14'te Reed Solomon kod çözücü devrenin blok diyagramı verilmektedir.



Şekil 14**.** Reed – Solomon kod çözücü devrenin blok diyagramı

3.6. ifadesinde $e_i = r_i + v_i$ ile bellidir ve alınan sözcüğün ilgili yerinde oluşan hata değerini göstermektedir. Hata sözcüğü *e(X),* $X^{j_1}, X^{j_2}, \ldots\ldots\ldots X^{j_v}$, olmak üzere toplam *v* yerde hata üretmiş ise $0 \le j_i \le n-1$ olmak üzere, hata polinomu,

$$e(X) = e_{j_1} X^{j_1} + e_{j_2} X^{j_2} + \ldots\ldots\ldots e_{jv} X^{j_v} \tag{3.7}$$

olarak bulunur. Hatanın düzeltilebilmesi için hata yerleri X^{j_i}'ler ve hata değerleri e_{j_1}'ler bilinmelidir. BCH kodlarında olduğu gibi $l = 1,2,\ldots\ldots\ldots v$ olmak üzere,

$$\beta_l = \alpha^{j_1} \tag{3.8}$$

tanımlanır ise hata belirteçleri 3.9 eşitlikleriyle hesaplanabilir.

$$\begin{aligned}
&S_1 = r(\alpha) = e_{j_1}\beta_1 + e_{j_2}\beta_2 + \ldots\ldots\ldots\ldots e_{j_v}\beta_v \\
&S_2 = r(\alpha^2) = e_{j_1}\beta_1^{\,2} + e_{j_2}\beta_2^{\,2} + \ldots\ldots\ldots e_{j_v}\beta_v^{\,2} \\
&.. \\
&.. \\
&S_{2t} = r(\alpha^{2t}) = e_{j_1}\beta_1^{\,2t} + e_{j_2}\beta_2^{\,2t} + \ldots\ldots\ldots e_{j_v}\beta_v^{\,2t}
\end{aligned} \tag{3.9}$$

Alınan sözcükte *t* hata varsa 3.9 eşitlikleri 3.10 ile ifade edilebilir.

$$S_k = \sum_{i=1}^{t} e_{j_i}\beta_i^{\,k} \tag{3.10}$$

Kod çözme işleminin amacı 3.10 ile verilen hata belirteçlerini üreten maksimum *t* hatadan oluşan hata dizisini bulmaktır.

2.3.3.2. Peterson'un Doğrudan Çözüm Yöntemi

Daha önce Peterson'un doğrudan çözüm yönteminin ikili BCH kodlarına uygulanması açıklanmıştı. Bölüm 2.2.1 de elde edilen sonuçlar birkaç değişiklikle ikili olmayan BCH kodları ve RS kodlarına uygulanabilir. 2.29. ile verilen Newton özdeşlikleri hatırlanırsa,

$$S_{t+j} + \sigma_1 S_{t+j-1} + ... + \sigma_t S_j = 0 \tag{3.11}$$

3.11 ile verilen Newton özdeşliklerinde geçen σ_i'ler hata yeri polinomu $\sigma(X)$'in katsayılarıdır.

$$\sigma(X) = X^t + \sigma_1 X^{t-1} + ... + \sigma_t \tag{3.12}$$

Kod çözme işlemini ilk adımında $2t$ belirteci $S_1, S_2, ...S_{2t}$ hesaplanır. Daha sonra $1 \le j \le t$ için 3.12'den t eşitlik elde edilir. Elde edilen eşitliklerin çözümü hata yeri polinomu $\sigma(X)$'in katsayılarını verir. Örnek olarak $t=3$ hata düzelten RS kodunu ele alalım. 3.11.'den;

$$\begin{aligned} S_1\sigma_3 + S_2\sigma_2 + S_3\sigma_1 &= -S_4 \\ S_2\sigma_3 + S_3\sigma_2 + S_4\sigma_1 &= -S_5 \\ S_3\sigma_3 + S_4\sigma_2 + S_5\sigma_1 &= -S_6 \end{aligned} \tag{3.13}$$

elde edilir. Hata yeri polinomunun katsayılarının bulunması için 3.13 formunda verilen eşitliklerin çözülmesi gerekir. Kullanılacak denklem sayısı alınan sözcükteki hata sayısına eşittir. 3.13'le verilen denklemleri matris biçiminde yazılsın.

$$\begin{bmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \end{bmatrix} \begin{bmatrix} \sigma_3 \\ \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} -S_4 \\ -S_5 \\ -S_6 \end{bmatrix} \tag{3.14}$$

Tek ve çift hatalı durumlar için 3.14 daha basit bir hal alır.

$$[S_1][\sigma_1] = [-S_2] \tag{3.15}$$

$$\begin{bmatrix} S_1 & S \\ S_2 & S_3 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} -S_3 \\ -S_4 \end{bmatrix} \tag{3.16}$$

3.14, 3.15 ve 3.16'dan farklı hatalı durumlar için hata yeri polinomunun katsayıları bulunabilmesi için bu ifadelerde yer alan katsayı matrislerinin determinantının sıfırdan farklı olması gerekir. D_2, 3.16'daki katsayı matrisinin determinantı, D_3, 3.14'teki üç hatalı durumun katsayı matrisinin determinantıdır ve bu determinantlar 3.17 ve 3.18 ile bellidir.

$$D_2 = S_1 S_3 - S_2^{\,2} \tag{3.17}$$

$$D_3 = S_1 S_3 S_5 + S_2 S_3 S_4 + S_2 S_3 S_4 - S_3^{\,3} - S_1 S_4^{\,2} - S_2^{\,2} S_5 \tag{3.18}$$

Alınan sözcükte kaç hata olduğunun belirlenebilmesi için önce D_3 ve D_2 hesaplanır Tek hatanın olduğu durumda bu iki ifade sıfıra eşittir. Çift hatalı sözcükler için ise sadece D_3 sıfır olur. σ_i'ler belirlendikten sonra hata yeri polinomunun kökleri, hata yerleri bulunur. Hata yerleri hata belirteci denklemi 3.10'da kullanılırsa hata değerleri bulunur ve kod çözme işlemi tamamlanır. Eğer kod $GF\left(2^m\right)$ alanı üzerinde tanımlı ise toplama ve çıkarma işlemleri eşdeğer olduğundan 3.11 – 3.18 denklemlerinde basitleştirmeler mümkündür. Peterson'un doğrudan çözüm yönteminin adımlarını üç hata düzelten ve $GF\left(2^4\right)$ üzerinde tanımlı (15, 9) RS kodu üzerinde incelensin.

1. Hata belirteçleri 3.9 yardımıyla S_k, $1 \le k \le 6$ için hesaplanır.

$$S_k = r\left(\alpha^k\right)$$

$S_k = 0$, $1 \le k \le 6$ ise alınan sözcük bir kod sözcüktür ve iletim sırasında hata oluşmadığı varsayılır.

2. Alınan sözcükteki hata sayısının belirlenmesi:

a. $D_3 = S_1 S_3 S_5 + S_3^{\,3} + S_1 S_4^{\,2} + S_2^{\,2} S_5 \neq 0$ ise üç hata olduğu varsayılır.

b. $D_3 = 0$ ve $D_2 = S_1 S_3 + S_2^{\,2} \neq 0$ ise iki hata olduğu varsayılır.

c. $D_3 = D_2 = 0$ ve $S_1 \neq 0$ ise bir hata olduğu varsayılır.

3. Hata yeri polinomunun katsayılarının bulunması

a. Üç hata olması hali:

$$\sigma_1 = \frac{1}{D_3}\left[S_1S_3S_6 + S_1S_4S_5 + S_2{}^2S_6 + S_2S_3S_5 + S_2S_4{}^2 + S_3{}^2S_4\right]$$
$$\sigma_2 = \frac{1}{D_3}\left[S_1S_4S_6 + S_1S_5{}^2 + S_2S_3S_6 + S_2S_4S_5 + S_3S_4{}^2 + S_3{}^2S_5\right] \quad (3.19)$$
$$\sigma_3 = \frac{1}{D_3}\left[S_2S_4S_6 + S_2S_5{}^2 + S_3{}^2S_6 + S_4{}^3\right]$$

b. İki hata olması hali:

$$\sigma_1 = \frac{1}{D_2}\left[S_1S_4 + S_2S_3\right]$$
$$\sigma_2 = \frac{1}{D_2}\left[S_2S_4 + S_3{}^2\right] \quad (3.20)$$

c. Bir hata olması hali:

$$\sigma_1 = \beta_1 = \frac{S_2}{S_1} \quad (3.21)$$

4. Hata yerlerinin bulunması:

a. Üç hata olması hali

$$\sigma(X) = X^3 + \sigma_1 X^2 + \sigma_2 X + \sigma_3 \quad (3.22)$$

b. İki hata olması hali

$$\sigma(X) = X^2 + \sigma_1 X + \sigma_2 \quad (3.23)$$

polinomlarının kökleri bulunur. Tek hata olması halinde hata yeri polinomu $\sigma(X) = X + \sigma_1$, hata yeri polinomunun kökü 3.21'den de görülebileceği gibi $\sigma_1 = \frac{S_2}{S_1}$'dir.

Eğer doğru sayıda kök, yani hata yeri bulunamazsa düzeltilemez bir hata oluşmuş demektir.

5. Hata yerleri belirlendikten sonra hata belirteci denklemleri 3.10 çözülerek hata değerleri hesaplanır.

a. Üç hata olması hali:

$$C = \beta_1\beta_2{}^2\beta_3{}^3 + \beta_1{}^3\beta_2\beta_3{}^2 + \beta_1{}^2\beta_2{}^3\beta_3 + \beta^3{}_1\beta_2{}^2\beta_3 + \beta_1\beta_2{}^3\beta_3{}^2 + \beta_1{}^2\beta_2\ \beta_3{}^3$$

$$e_{j_1} = \frac{1}{C}\left[S_1\beta_2{}^2\beta_3{}^3 + S_2\beta_2{}^3\beta_3 + S_3\beta_2\ \beta_3{}^2 + S_1\beta_2{}^3\beta_3{}^2 + S_2\beta_2\ \beta_3{}^3 + S_3\beta_2{}^2\beta_3\right]$$

$$e_{j_2} = \frac{1}{C}\left[S_1\beta_1{}^3\beta_3{}^2 + S_2\beta_1\ \beta_3{}^3 + S_3\beta_1{}^2\beta_3 + S_1\beta_1{}^2\beta_3{}^3 + S_2\beta_1{}^3\beta_3 + S_3\beta_1\ \beta_3{}^2\right]$$

$$e_{j_3} = \frac{1}{C}\left[S_1\beta_1{}^2\beta_2{}^3 + S_2\beta_2\ \beta_1{}^3 + S_3\beta_2{}^2\beta_1 + S_1\beta_1{}^3\beta_2{}^2 + S_2\beta_2{}^3\beta_1 + S_3\beta_2\ \beta_1{}^2\right] \qquad (3.24)$$

b. İki hata olması hali:

$$e_{j_1} = \frac{S_1\beta_2 + S_2}{\beta_1\beta_2 + \beta_1{}^2}$$

$$e_{j_2} = \frac{S_1\beta_1 + S_2}{\beta_1\beta_2 + \beta_2{}^2} \qquad (3.25)$$

c. Tek hata olması hali:

$$e_{j_1} = \frac{S_1{}^2}{S_2} \qquad (3.26)$$

6. Alınan sözcüğün hatalı yerlerindeki semboller, hesaplanan hata değerleri ile toplanarak düzeltme işlemi yapılır.

7. Düzeltilmiş sözcük için hata belirteci hesaplanır. Eğer hata belirteci sıfırdan farklı ise düzeltilmiş sözcük de hatalıdır ve gerçekte hata düzeltilememiştir.

Örnek 3.2. Örnek 3.1'de üreteç polinomu verilen (15 , 9) RS kodu için gönderilen kod vektörü $v=(000000000000000)$, alınan kod vektörü $r=(000\alpha^{7}00\alpha^{3}00000\alpha^{4}00)$ olsun. Alınan sözcüğün polinomsal ifadesi $r(X)=\alpha^{7}X^{3}+\alpha^{3}X^{6}+\alpha^{4}X^{12}$ olur. 3.9'dan hata belirteçleri,

$S_1=\alpha^{12}$ $\quad$ $S_4=\alpha^{10}$
$S_2=1$ $\quad$ $S_5=0$
$S_3=\alpha^{14}$ $\quad$ $S_6=\alpha^{12}$

olarak bulunur. 3.18'den $D_3=\alpha^{7}\neq 0$ olarak bulunacağından üç hata olduğu varsayılır.

3.19 eşitliklerinden,

$$\sigma_1=\alpha^{7},\sigma_2=\alpha^{4},\sigma_3=\alpha^{6}$$

$$\sigma(X)=X^{3}+\alpha^{7}X^{2}+\alpha^{4}X+\alpha^{6}$$

bulunur. $GF(2^4)$'ün tüm elemanları $\sigma(X)$'te X yerine konursa

$$\sigma(\alpha^{3})=\sigma(\alpha^{6})=\sigma(\alpha^{12})=0$$

olduğu görülebilir. Buradan 3. , 6. ve 12. sembollerin hatalı alındığı ortaya çıkar. 3.24 eşitliklerinden hata değerleri hesaplanabilir.

$$e_3=\alpha^{7},e_6=\alpha^{3},e_{12}=\alpha^{4}$$
$$e(X)=\alpha^{7}X^{3}+\alpha^{3}X^{6}+\alpha^{4}X^{12}$$

Hata dizisi 3.6.'da yerine konursa

$$v(X)=r(X)+e(X)$$

bulunur. Düzeltilen sözcük için hata belirteçlerinin sıfıra eşit olacağı açıktır. Üç yerde hatalı iletilen sıfır kod sözcüğündeki hatalar düzeltilmiştir.

2.3.3.3. Berlekamp Yöntemi

Peterson yöntemiyle az sayıda hatalı sembolü düzelten kodların çözümü basittir. Altıdan fazla hatalı kodların çözümünde $\sigma(X)$'in katsayılarının hesaplanması oldukça karmaşık işlemler gerektirdiğinden ikili BCH kodlarının çözümünde olduğu gibi RS kodlarının çözümünde de Berlekamp yönteminden yararlanılır. İkili BCH kodlarından farklı olarak RS kodlarının çözümünde hata değerinin de bulunması gerekir. Hata yerleri belirlendikten, hata değerlendirme polinomu, $\omega(X)$ bulunarak hata değerleri hesaplanır. $\omega(X)$ ve $\sigma(X)$ arasındaki ilişki 3.27 bağıntısıyla bellidir.

$$\omega(X) = \sigma(X)[1 + S(X)] \bmod X^{2t+1} \tag{3.27}$$

Bu bağıntı anahtar eşitlik olarak bilinir ve BCH kodların Berlekamp yöntemiyle çözümünde kullanılır.

Bağıntıda $S(X)$ hata belirteci polinomu olarak bilinir ve

$$S(X) = S_1 X + S_2 X^2 + ... + S_{2t} X^{2t} \tag{3.28}$$

biçimindedir. Ayrıca anahtar eşitlikteki hata yeri polinomu $\sigma(X)$'in tanımı Peterson yönteminden farklıdır. Artık $\sigma(X)$, β_i hata yerleri ve $i. = 1,2.....t$ olmak üzere;

$$\sigma(X) = (1 + \beta_1 X)(1 + \beta_2 X).....(1 + \beta_t X) \tag{3.29}$$

İfadesiyle tanımlanmıştır. Yani $\sigma(X)$'in köklerinin tersi hata yerlerini vermektedir. Berlekamp yöntemi $\sigma(X)$'in bulunması için geliştirilmiş iteratif bir yöntemdir. Hata yeri polinomu bulunduktan sonra polinomun köklerinin tersleri hata yerlerini verir. Hata yeri polinomu 3.27'de yerine konursa hata değerlendirme polinomu $\omega(X)$ bulunur. $\sigma'(X)$, hata yeri polinomu $\sigma(X)$'in X'e türevi olmak üzere, β hata yerine karşı düşen hata değeri e_{j_i}, 3.30 ifadesiyle hesaplanır.

$$e_{j_i} = \beta_i \frac{\omega(\beta_i^{-1})}{\sigma'(\beta_i^{-1})} \tag{3.30}$$

3.7 yardımıyla hata polinomu $e(X)$ bulunur ve 3.6 ile verilen $v(X)=r(X)+e(X)$ ifadesinde yerine konursa hata düzeltme işlemi gerçekleştirilir.

Örnek 3.3: Berlekamp yöntemi, Örnek 3.2'de Peterson yöntemiyle çözülen RS koduna uygulanacaktır. (15, 9) RS kodu için gönderilen kod vektörü $v=(000000000000000)$, alınan kod vektörü $r=(000\alpha^{7}00\alpha^{3}00000\alpha^{4}00)$ olsun. Alınan sözcüğün polinomsal ifadesi $r(X)=\alpha^{7}X^{3}+\alpha^{3}X^{6}+\alpha^{4}X^{12}$ olur. 3.9'dan hata belirteçleri,

$S_1=\alpha^{12}$ $\quad$ $S_4=\alpha^{10}$
$S_2=1$ $\quad$ $S_5=0$
$S_3=\alpha^{14}$ $\quad$ $S_6=\alpha^{12}$

olarak bulunur. Berlekamp yöntemini Tablo 7'yi doldurarak uygulanır.

Tablo 7. Berlekamp yönteminin Örnek 3.3'e uygulanışı

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$\mu-l_\mu$
-1	1	1	0	-1
0	1	$S_1=\alpha^{12}$	0	0
1	$1+\alpha^{12}X$	α^{7}	1	0
2	$1+\alpha^{3}X$	1	1	1
3	$1+\alpha^{3}X+\alpha^{3}X$	α^{7}	2	1
4	$1+\alpha^{4}X+\alpha X$	α^{10}	2	2
5	$1+\alpha^{7}X+\alpha^{4}X^{2}+\alpha^{6}X^{3}$	0	3	2
6	$1+\alpha^{7}X+\alpha^{4}X^{2}+\alpha^{6}X^{3}$	-	-	-

Yöntemin 5. adımında minimum uzaklığı $d_{\mu} = 0$ olarak hesaplanan polinom hata yeri polinomudur.

$$\sigma(X) = 1 + \alpha^{7} X + \alpha^{4} X^{2} + \alpha^{6} X^{3}$$

3.27 ile verilen anahtar eşitlikten hata değerlendirme polinomu hesaplanır.

$$\omega(X) = \sigma(X)[1 + S(X)] \bmod X^{2t+1}$$
$$\omega(X) = \left[\alpha^{3} X^{9} + \alpha X^{8} + X^{7} + \alpha^{6} X^{3} + X^{2} + \alpha^{2} X + 1\right] \bmod X^{7}$$
$$\omega(X) = \alpha^{6} X^{3} + X^{2} + \alpha^{2} X + 1$$

GF(2^4)'ün tüm elemanları $\sigma(X)$'de yerine konursa, $\alpha^{3}, \alpha^{9}, \alpha^{12}$ $\sigma(X)$'in kökleri olarak bulunur. Köklerin tersi $\alpha^{12}, \alpha^{6}, \alpha^{3}$ hata yerleridir. Yani 3. , 6. ve 12. semboller hatalı alınmıştır. Hata değerlerinin hesabı için 3.30 kullanılırsa;

$$e_{3} = \left(\alpha^{3}\right)^{-1} \frac{\omega\left(\left(\alpha^{3}\right)^{-1}\right)}{\sigma'\left(\left(\alpha^{3}\right)^{-1}\right)} = \alpha^{7}$$

$$e_{6} = \left(\alpha^{6}\right)^{-1} \frac{\omega\left(\left(\alpha^{6}\right)^{-1}\right)}{\sigma'\left(\left(\alpha^{6}\right)^{-1}\right)} = \alpha^{3}$$

$$e_{12} = \left(\alpha^{12}\right)^{-1} \frac{\omega\left(\left(\alpha^{12}\right)^{-1}\right)}{\sigma'\left(\left(\alpha^{12}\right)^{-1}\right)} = \alpha^{4}$$

bulunur.

Hata polinomu

$$e(X) = \alpha^{7} X^{3} + \alpha^{3} X^{6} + \alpha^{4} X^{12}$$

olur. Sonuç, Peterson yöntemiyle elde edilen ile aynıdır. 3.6 kullanılarak hata düzeltme işlemi yapılır.

$$v(X) = r(X) + e(X) = 0$$

2.3.3.4. Euclid Yöntemi

RS kodlarının çözümünde Euclid yöntemi de kullanılabilir. Euclid yöntemi, *A* ve *B* olarak verilen herhangi iki tamsayı veya polinomun ortak bölenlerinin en büyüğü (OBEB), *C*'yi verir. Ayrıca *C=SA+TB* eşitliğini sağlayan *S* ve *T* tamsayıları veya polinomlarını da bulur. Anahtar eşitliği 3.31 biçimde ifade edilirse Euclid yöntemiyle $\sigma(X)$ ve $\omega(X)$ polinomları bulunabilir.

$$\omega(X) = \sigma(X)[1 + S(X)] + \mu(X)X^{2t+1} \tag{3.31}$$

Yöntemi uygularken $A = X^{2t+1}$, $B = 1 + S(X)$ polinomlarını seçersek, $\sigma(X)$ ve $\omega(X)$ polinomlarını bulabiliriz. Öncelikle Euclid yöntemini tamsayılar ve polinomlar için tanımlanarak ve yöntemin uygulanışını bir örnek üzerinde incelenecektir.

A ve *B* tamsayı ise $A \geq B$, polinom ise *A*'nın derecesi ve *B*'nin derecesinden daha büyük eşit olsun $(der(A) \geq der(B))$ Başlangıç koşulu olarak $r_{-1} = A$ ve $r_0 = B$ seçersek, yöntemin n. *a* dımında r_{n-2}'nin r_{n-1}'e bölümünden kalan r_n'i elde ederiz. $(r_{n-2} = q_n r_{n-1} + r_n)$. Tamsayılar için $r_n \leq r_{n-1}$ ve polinomlar için $der(r_n) \leq der(r_{n-1})$ olur ve *n.* adımındaki r_n

$$r_n = r_{n-2} - q_n r_{n-1} \tag{3.32}$$

ile bulunur. Aynı zamanda $r_n = s_n A + t_n B$ eşitliğini sağlayan s_n ve t_n sayıları da bulunabilir. 3.32'ye benzer ifadeler s_n ve t_n içinde geçerlidir.

$$s_n = s_{n-2} - q_n s_{n-1} \tag{3.33}$$

$$t_n = t_{n-2} - q_n t_{n-1} \tag{3.34}$$

$r_{-1} = A = (1)A + (0)B$ ve $r_0 = B = (0)A + (1)B$ olduğundan başlangıç koşulu olarak $s_{-1} = 1, t_{-1} = 1, s_0 = 0, t_0 = 1$ alınır.

Örnek olarak 124 ve 46 sayılarının *OBEB*'ini bulmak için Euclid yöntemi kullanılsın. İlk adımda 124'ün 46'ya bölümünden kalan $r_1 = 32$, bölüm $q_1 = 2$ 'dir. İkinci adımda 46'nın 32'ye bölümünden kalan $r_2 = 14$, bölüm $q_2 = 1$ olur.

Bölümden kalan sıfır oluncaya kadar işleme devam edilirse *OBEB(124,46)* = 2 bulunur. Yöntemin her adımında kalan s_n ve t_n cinsinden aşağıdaki gibi ifade edilebilir.

124 = (1).124 + (0).46

46 = (0).124 + (1).46

32 = (1).124 + (-2).46

14 = (-1).124 + (3).46

4 = (3).124 + (-8).46

2 = 10).124 + (27).46

0 = (23).124 + (-62).46

Euclid yöntemini uygulamanın en kolay yolu r_n, q_n, s_n ve t_n için bir tablo oluşturmaktır. Örneğe ilişkin Tablo 8. aşağıda verilmiştir.

Tablo 8. Euclid yönteminin tamsayılara uygulanışı

n	r_n	q_n	$s_n = s_{n-} - q_n s_{n-}$	$t_n = t_{n-} - q_n t_{n-}$
-1	124	-	1	0
0	46	-	0	1
1	32	2	1	-2
2	14	1	-1	3
3	4	2	3	-8
4	2	3	-10	27
5	0	2	23	-62

Euclid yöntemini

$$\omega(X) = \sigma(X)[1 + S(X)] + \mu(X)X^{2t+1}$$

anahtar eşitliğinin çözümünde kullanmak üzere, X^{2t+1} ve $[1 + S(X)]$'e uygularsak, kodun hata kapasitesi aşılmadığı sürece belli bir adımda

$$r_n(X) = s_n(X)X^{2t+1} + t_n(X)[1 + S(X)] \tag{3.35}$$

elde ederiz. Burada n, $der(r_n) \le t$ özelliğini sağlayan ilk n değeridir. Bu sonuca bağlı olarak

$$\sigma(X) = t_n(X) \tag{3.36}$$

$$\omega(X) = r_n(X) \tag{3.37}$$

elde edilir. Artık hata yeri polinomu ve hata değerlendirme polinomu bulunduğundan 3.30 yardımıyla hata değerleri bulunup kod çözme işlemi tamamlanabilir.

Örnek 3.4: Euclid yöntemini, Örnek 3.2'de Peterson yöntemiyle çözülen RS koduna uygulansın. (15 , 9) RS kodu için gönderilen kod vektörü $v = (000000000000000)$, alınan kod vektörü $r = (000\alpha^7 00\alpha^3 00000\alpha^4 00)$ olsun. Alınan kod sözcüğün polinomsal ifadesi $r(X) = \alpha^7 X^3 + \alpha^3 X^6 + \alpha^4 X^{12}$ olur. 3.9'dan hata belirteçleri,

$S_1 = \alpha^{12}$ $\quad$ $S_4 = \alpha^{10}$
$S_2 = 1$ $\quad$ $S_5 = 0$
$S_3 = \alpha^{14}$ $\quad$ $S_6 = \alpha^{12}$

$$S(X) = \alpha^{12}X^6 + \alpha^{10}X^4 + \alpha^{14}X^3 + X^2 + \alpha^{12}X$$

olarak bulunur. Euclid yöntemini uygulamak üzere Tablo 9'u oluşturalım.

Tablo 9. Euclid yönteminin Örnek 3.4'e uygulanışı

n	r_n	q_n	$t_n = t_{n-2} - q_n t_{n-1}$
-1	X^7	-	0
0	$1+S(X)$	-	1
1	$\alpha^{13}X^5+\alpha^2X^4+\alpha^3X^3+X^2+\alpha^3X$	α^3X	α^3X
2	$\alpha^8X^4+\alpha^6X^3+\alpha^{13}X^2+\alpha^4X+1$	$\alpha^4X+\alpha^3$	$\alpha^2X^2+\alpha^6X+1$
3	$\alpha^7X^3+\alpha X^2+\alpha^3X+\alpha$	$\alpha^5X+\alpha$	$\alpha^7X^3+\alpha^5X^2+\alpha^8X+\alpha$

Yöntemin üçüncü adımında $der(r_3)=3\le t=3$ olduğundan

$$\omega(X)=\alpha^7X^3+\alpha X^2+\alpha^3X+\alpha$$
$$\sigma(X)=\alpha^7X^3+\alpha^5X^2+\alpha^8X+\alpha$$

elde edilir. Dikkat edilirse elde edilen polinomlar Berlekamp yöntemiyle elde edilen polinomların α katı kadardır. Bu durum hata yeri polinomunun köklerini değiştirmediğinden sonucu etkilemez. Hata değerlerinin bulunmasında 3.30 ifadesine dikkat edilirse $\omega(X)$ ve $\sigma'(X)$ bölüm halinde olduğundan α çarpanı sadeleşeceğinden hata değerleri de değişmez. Hata değerleri

$$e_3=\left(\alpha^3\right)^{-1}\frac{\omega\left(\left(\alpha^3\right)^{-1}\right)}{\sigma'\left(\left(\alpha^3\right)^{-1}\right)}=\alpha^7$$

$$e_6=\left(\alpha^6\right)^{-1}\frac{\omega\left(\left(\alpha^6\right)^{-1}\right)}{\sigma'\left(\left(\alpha^6\right)^{-1}\right)}=\alpha^3$$

$$e_{12}=\left(\alpha^{12}\right)^{-1}\frac{\omega\left(\left(\alpha^{12}\right)^{-1}\right)}{\sigma'\left(\left(\alpha^{12}\right)^{-1}\right)}=\alpha^4$$

olarak bulunur.

Hata polinomu

$$e(X) = \alpha^{7} X^{3} + \alpha^{3} X^{6} + \alpha^{4} X^{12}$$

olur ve 3.6. kullanılarak hata düzeltme işlemi yapılır.

$$v(X) = r(X) + e(X) = 0$$

3. BULGULAR VE TARTIŞMA

Bu kısımda, AWGN ve dar bantlı Rayleigh kanal baz alınarak, reel ve karmaşık datalardan oluşan veri paketlerinin Reed – Solomon kod çözücüdeki başarımını etkileyen faktörler (modülasyon ve kod hızı) değiştirilmek suretiyle elde edilen farklı benzetim sonuçları verilmiştir.

Benzetimlerde, modülasyon teknikleri olarak BPSK, QPSK ve 16-QAM kullanılmıştır. Ayrıca benzetimlerde, dizi uzunluğu 1000 sembol, kanal sayısı 200 olarak seçilmiştir. 3/7, 3/5 ve 15/16 kod hızlarına sahip Reed – Solomon kodları için kod çözme algoritması olarak Berlekamp – Massey ve Forney algoritmaları kullanılmıştır. Tüm benzetim sonuçları, bit hata oranı (BER) ve sembol hata oranı (SER) çizelgeleri kullanılarak ifade edilmiştir. İşaret gürültü oranı (SNR), dB cinsinden ifade edilmiş olup, BPSK modülasyonu için E_b / N_o değerini temsil eder. QPSK modülasyonu için E_b / N_o (dB) = SNR (dB) + 3 dB ve 16-QAM modülasyonu için ise E_b / N_o (dB) = SNR (dB) + 6 dB değeri ile belirlenir.

Benzetim programı, Visual Studio 2005 paket programı kullanılarak yazılmıştır. Reel dataların simülasyonunda AWGN kanal için üretilen Gauss gürültüsü n_1,

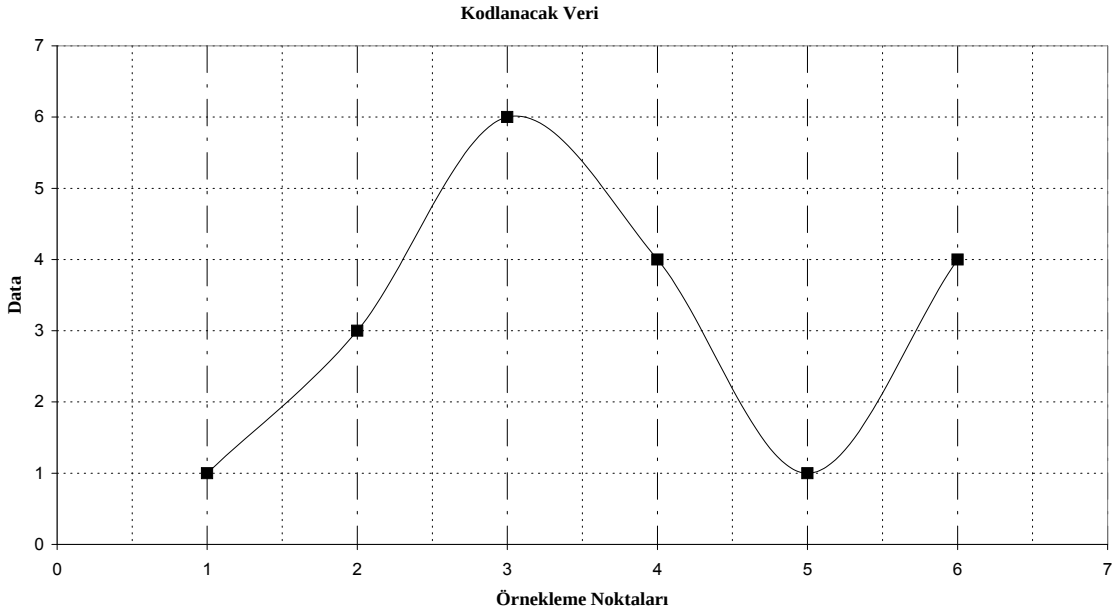
$$n_1 = \sqrt{-2\delta \log(U_1)}.\cos(U_2) \tag{3.1}$$

Karmaşık dataların simülasyonunda ise AWGN ve Rayleigh kanallar için Gauss gürültüsü n_2 ,

$$n_2 = \sqrt{-2\delta \log(U_1)}.\cos(U_2) + j\sqrt{-2\delta \log(U_1)}.\sin(U_2) \tag{3.2}$$

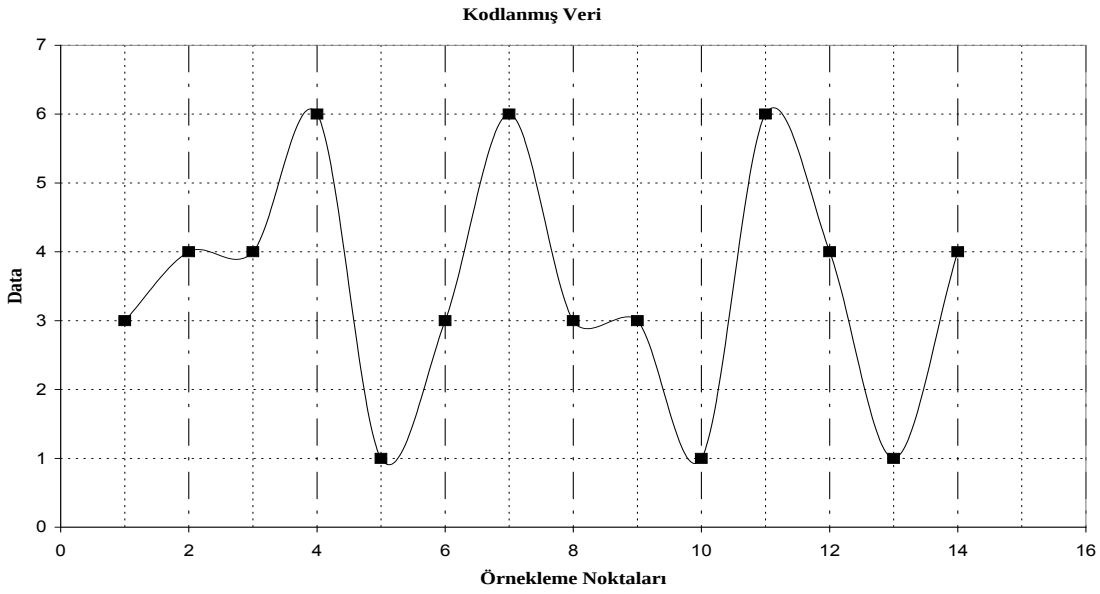
şeklindedir. Burada U_1 ve U_2, [0.0 - 1.0] aralığında düzenli dağılıma sahip, bağımsız rasgele değişkenleri temsil etmekte ve "rnd(&seed)" fonksiyonu kullanılarak elde edilmektedir. "rnd(&seed)" fonksiyonu, [0.0 - 1.0] aralığında düzenli dağılıma sahip reel sayılar üretmektedir. Gürültü varyansı $\delta = 0{,}001$ olarak alınmıştır.

Şekil 15’de kodlanacak veri paketi içinden alınan herhangi bir data paketi gösterilmektedir.



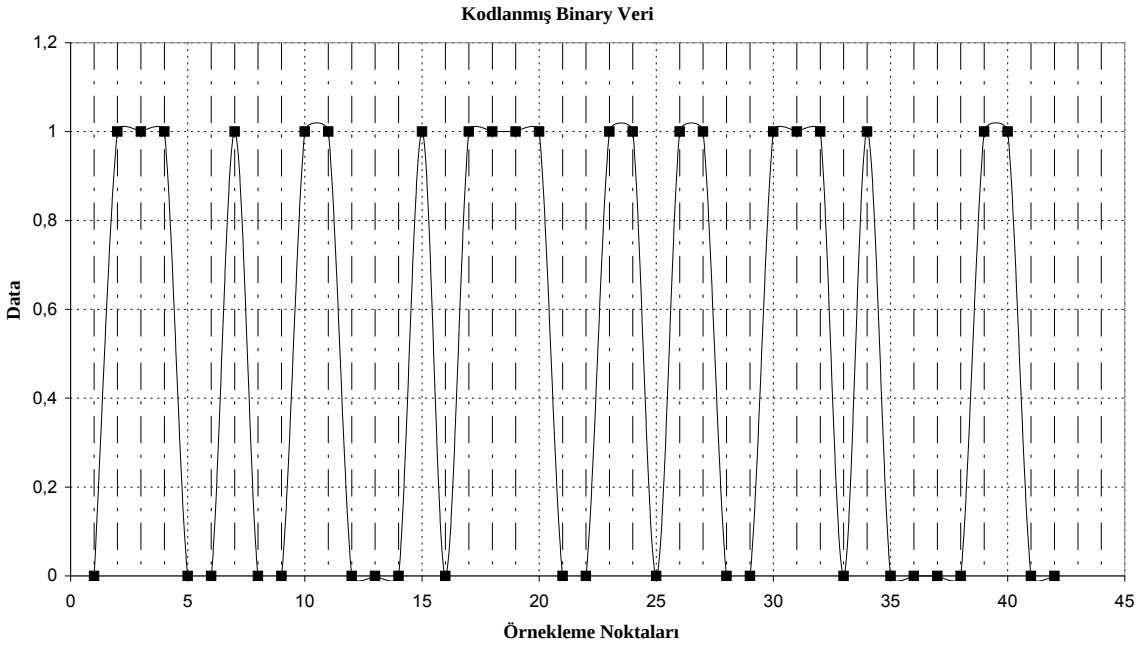
Şekil 15. Kodlanacak veri

Şekil 16’da kodlanmış veri paketi gösterilmektedir.



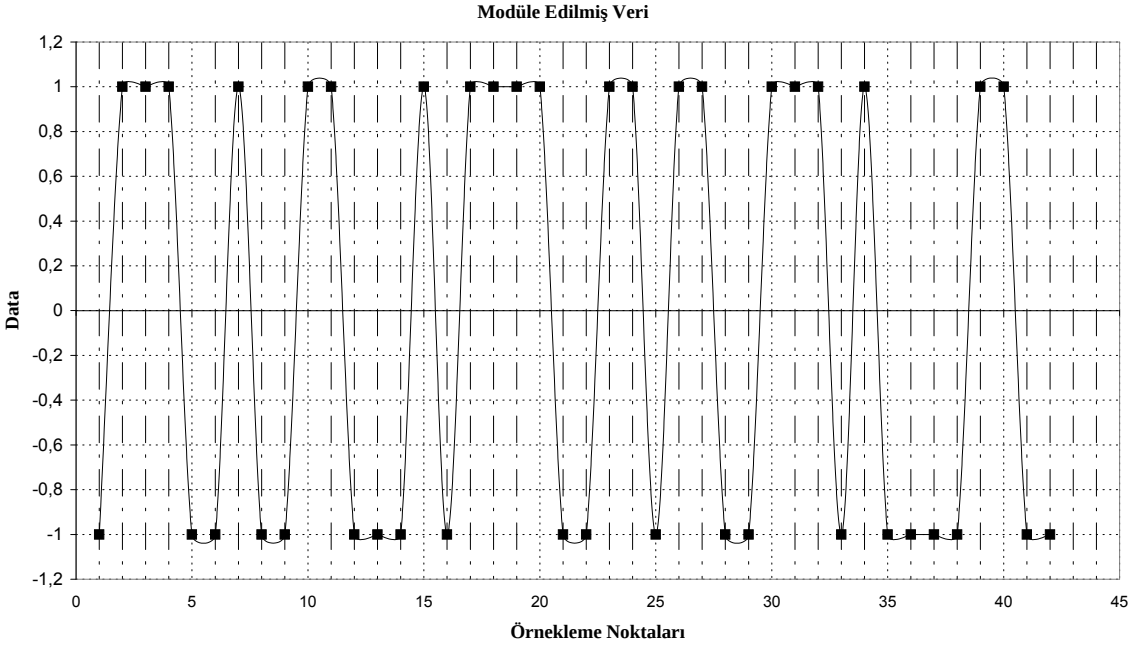
Şekil 16. Kodlanmış veri

Şekil 17’de kodlanmış binary veri paketi gösterilmektedir.



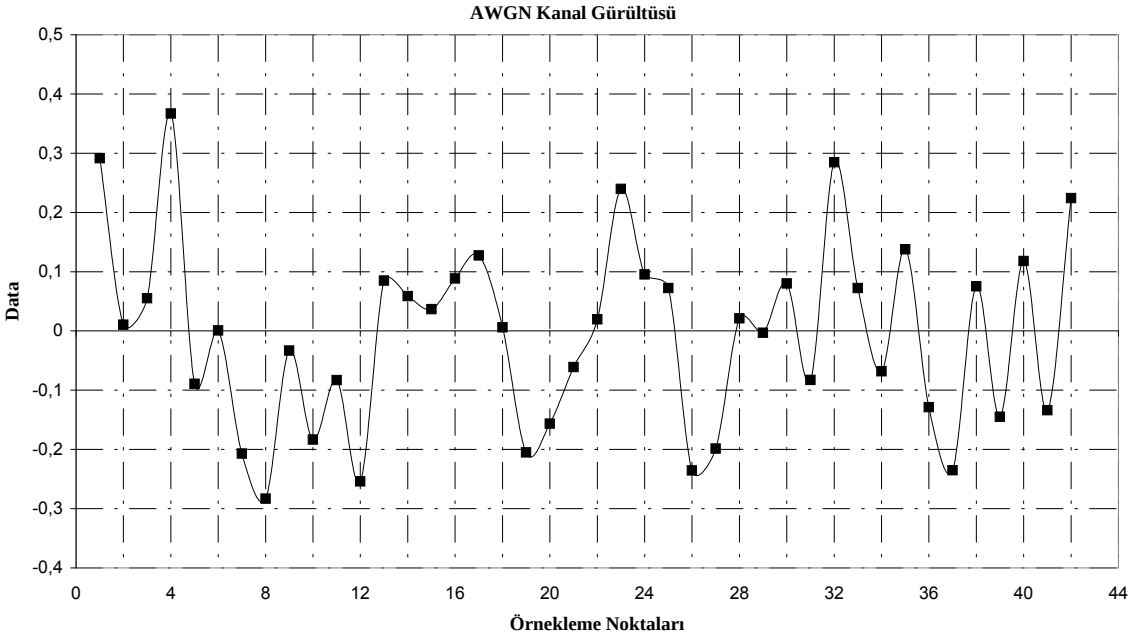
Şekil 17. Kodlanmış binary veri

Şekil 18’de kodlanmış ve module edilmiş veri paketi gösterilmektedir



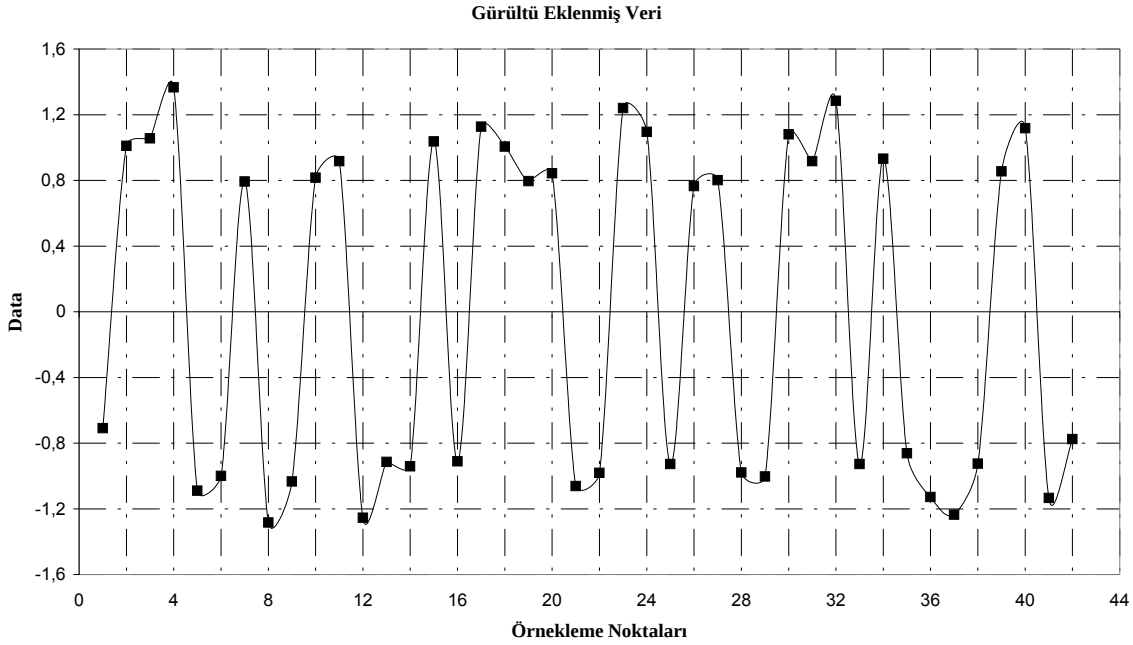
Şekil 18. Modüle edilmiş veri

Şekil 19’da gürültü paketi gösterilmektedir



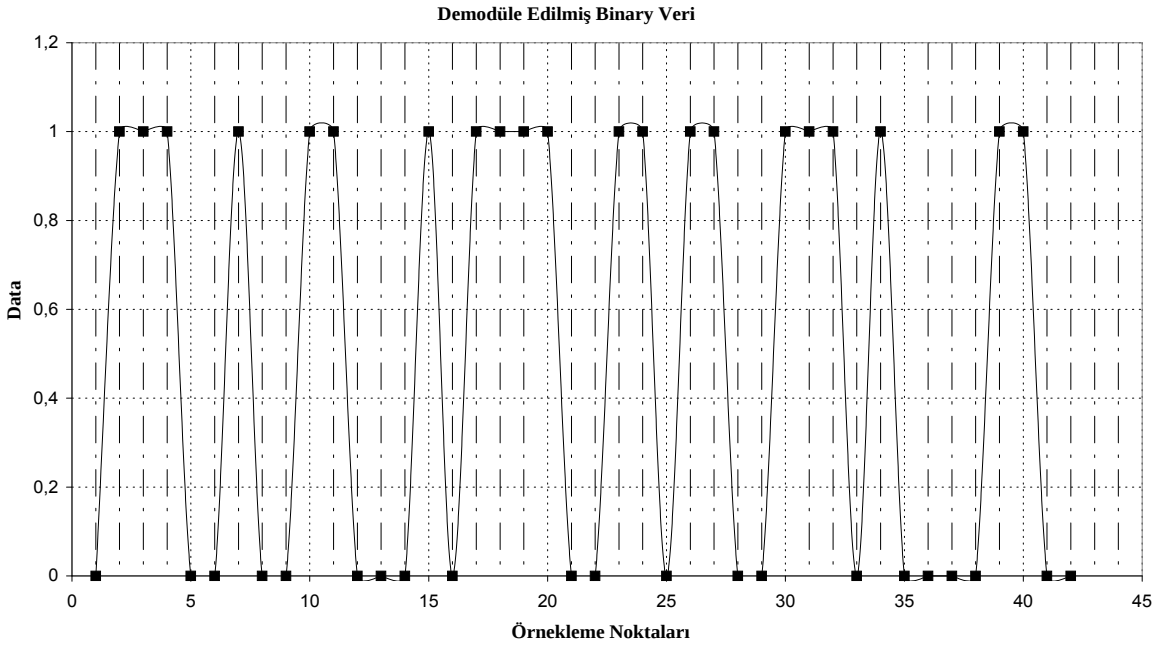
Şekil 19. Gürültü paketi

Şekil 20’de gürültü eklenmiş veri paketi gösterilmektedir



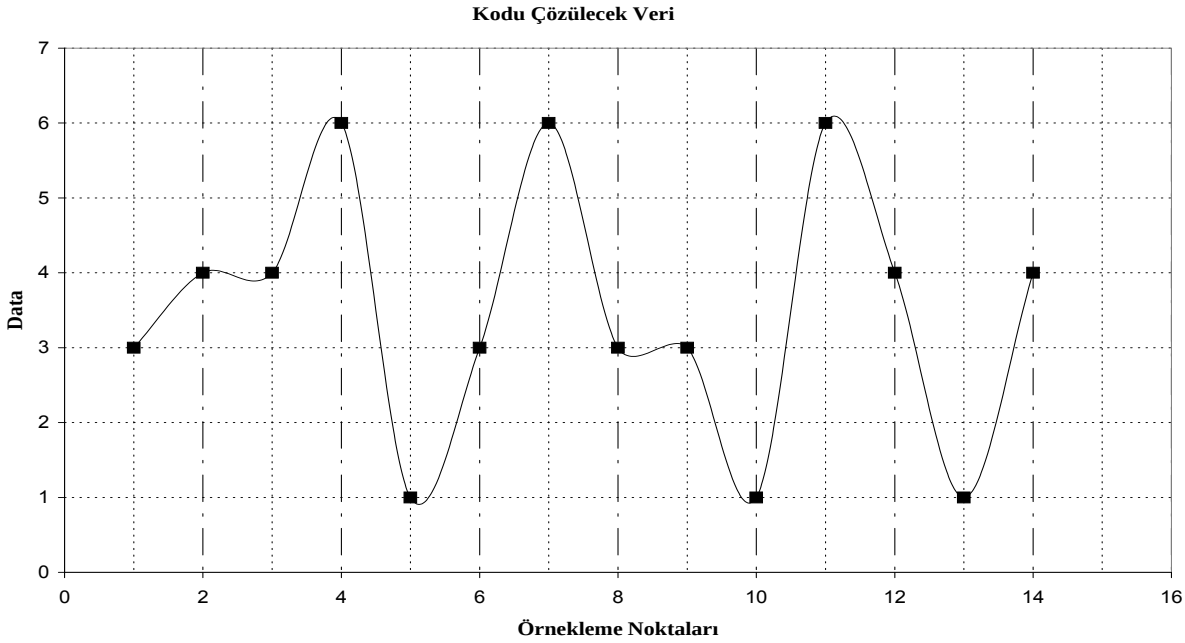
Şekil 20. Gürültü eklenmiş veri paketi

Şekil 21’de demodüle edilmiş veri paketi gösterilmektedir.



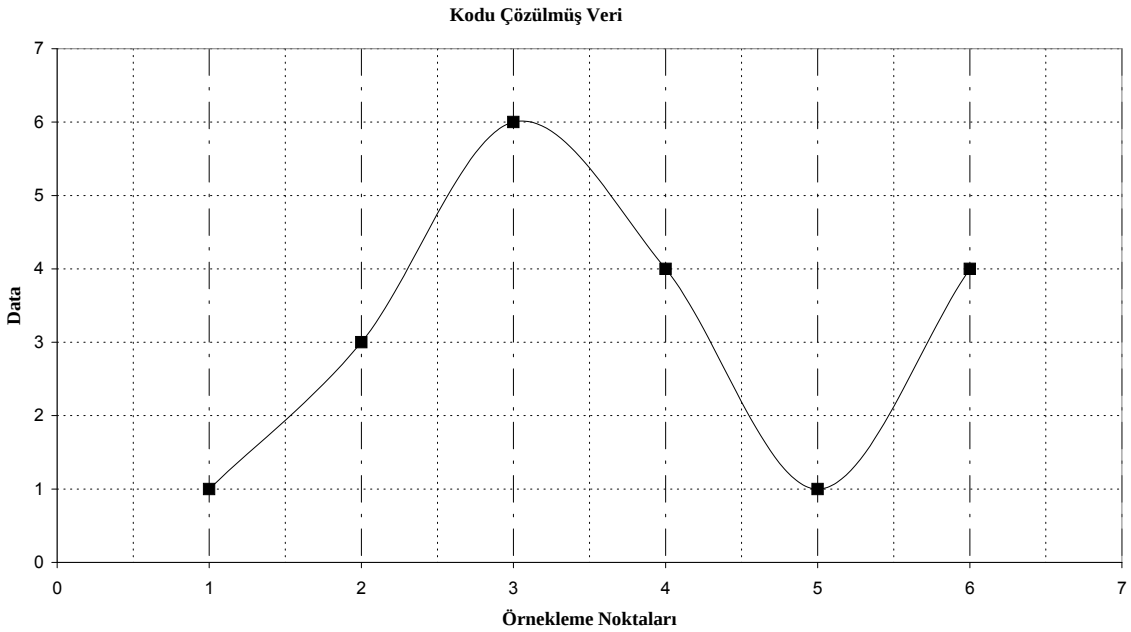
Şekil 21. Demodüle edilmiş veri paketi

Şekil 22’de kodu çözülecek veri paketi gösterilmektedir.



Şekil 22. Kodu çözülecek veri paketi

Şekil 23'de kodu çözülmüş veri paketi gösterilmektedir.



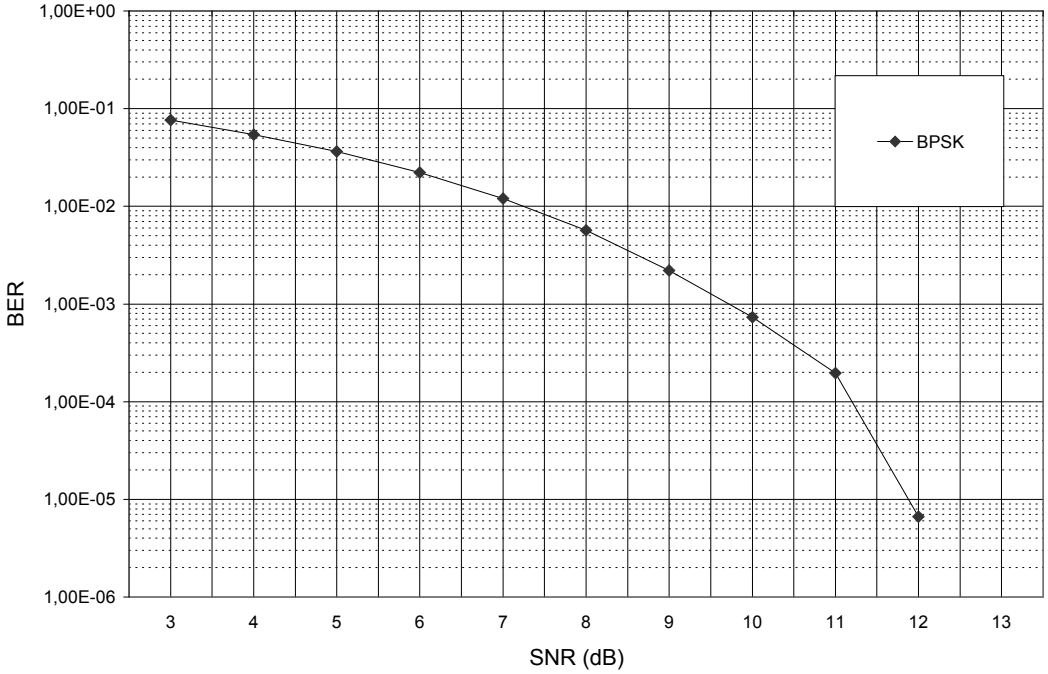
Şekil 23. Kodu çözülmüş veri paketi

3.1. Reed Solomon Kod Çözücüde Kod Hızının Etkisi (BER Analizi)

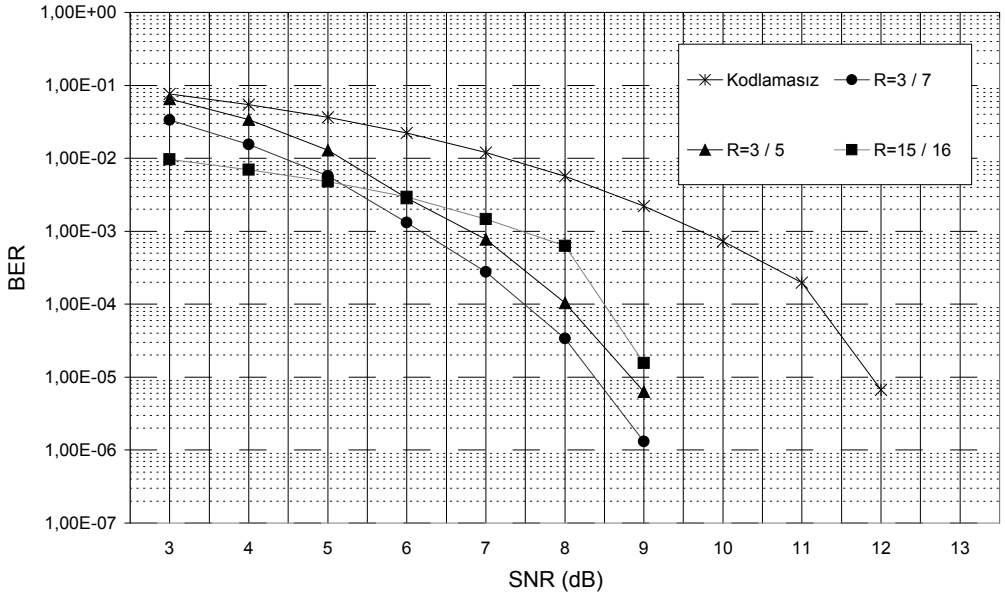
Benzetim sonuçlarında kod hızının etkisini gösterebilmek için, farklı modülasyon ve kanal modelleri kullanılarak, değişik kod hızları için benzetimler elde edilmiştir.

Kod hızındaki değişim, başarımdaki değişimle kısmen ters orantılıdır. Kod hızındaki azalma, yani bir biti kodlamak için kullanılan bit sayısındaki artış, doğal olarak başarımında bir yükselmeye neden olacaktır. Kod hızındaki azalma, Reed – Solomon kod çözücünün de işlem yükünü arttırmaktadır. Şekil 16'da, AWGN kanaldan gönderilen reel datalar için farklı kod hızlarında elde edilen benzetimler verilmiştir. Modülasyon olarak BPSK kullanılmıştır. Farklı kod hızları arasında yaklaşık 1 dB'lik başarım farkı elde edilmiştir. Kodlamasız duruma göre en iyi sonucu veren RS (7,3) kodlama sisteminin kullanılması durumunda bit hata olasılığının 10^{-5} olduğu mertebelerde yaklaşık 3,5 dB kazanç elde edilmiştir.

Şekil 24'te AWGN kanaldaki BPSK başarımı gösterilmektedir

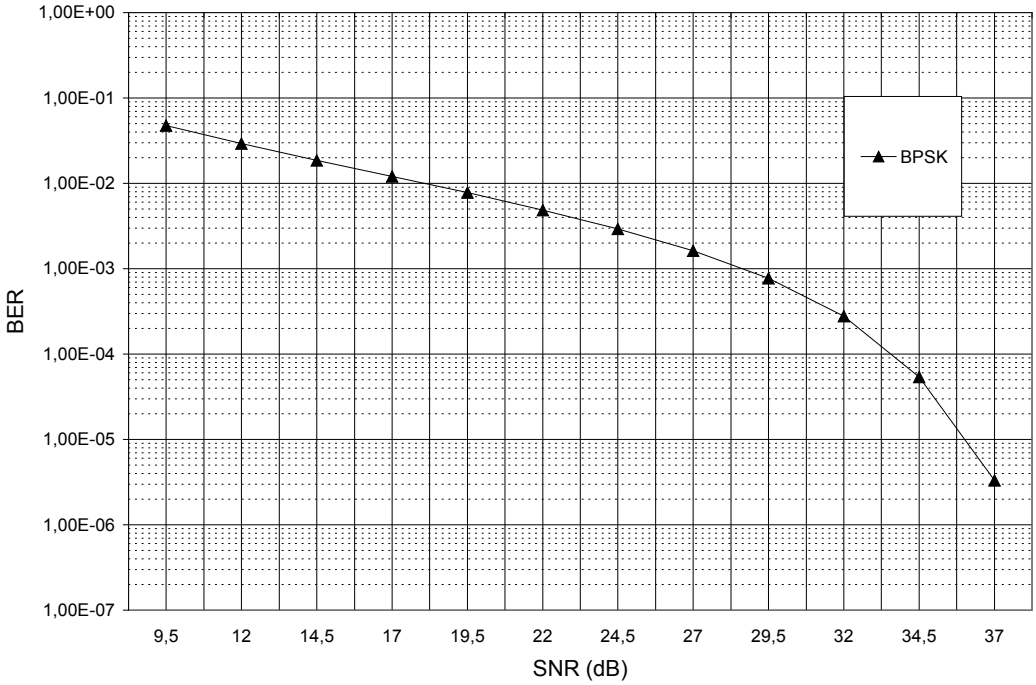


Şekil 24. AWGN kanalda BPSK başarımı, BER Analizi

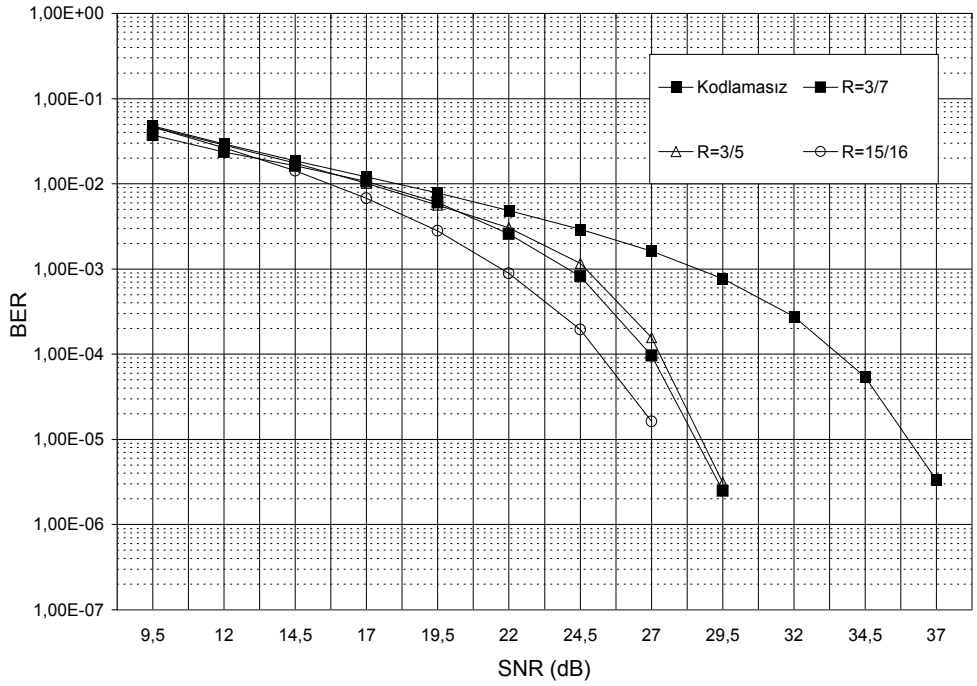


Şekil 25. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, BER Analizi
[1000 sembol, 200 kanal, AWGN kanal, BPSK modülasyonu]

Şekil 26'da Rayleigh kanaldaki BPSK başarımı gösterilmektedir.



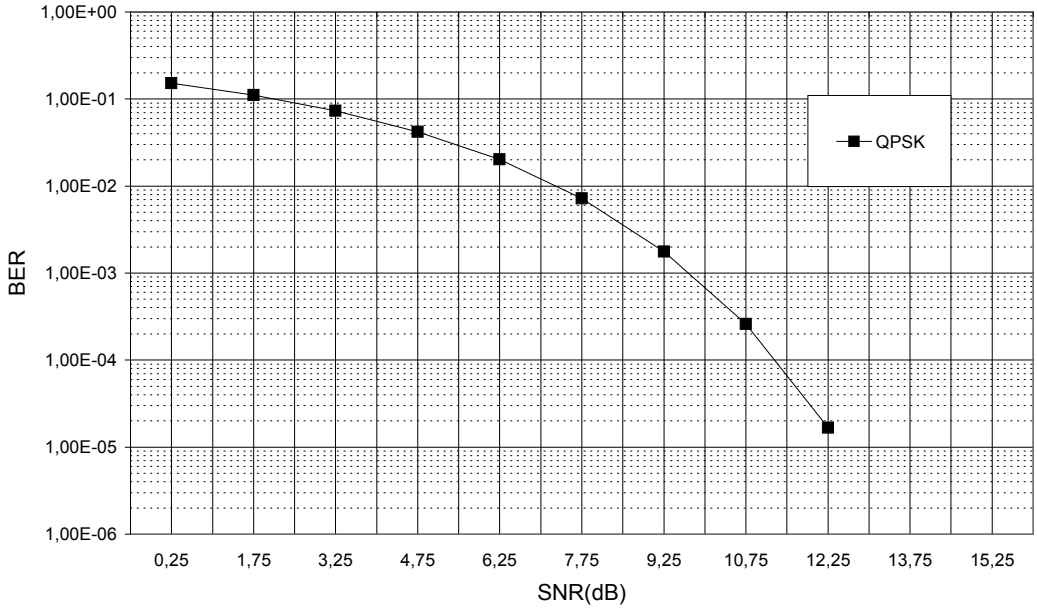
Şekil 26. Rayleigh kanalda BPSK başarımı, BER Analizi



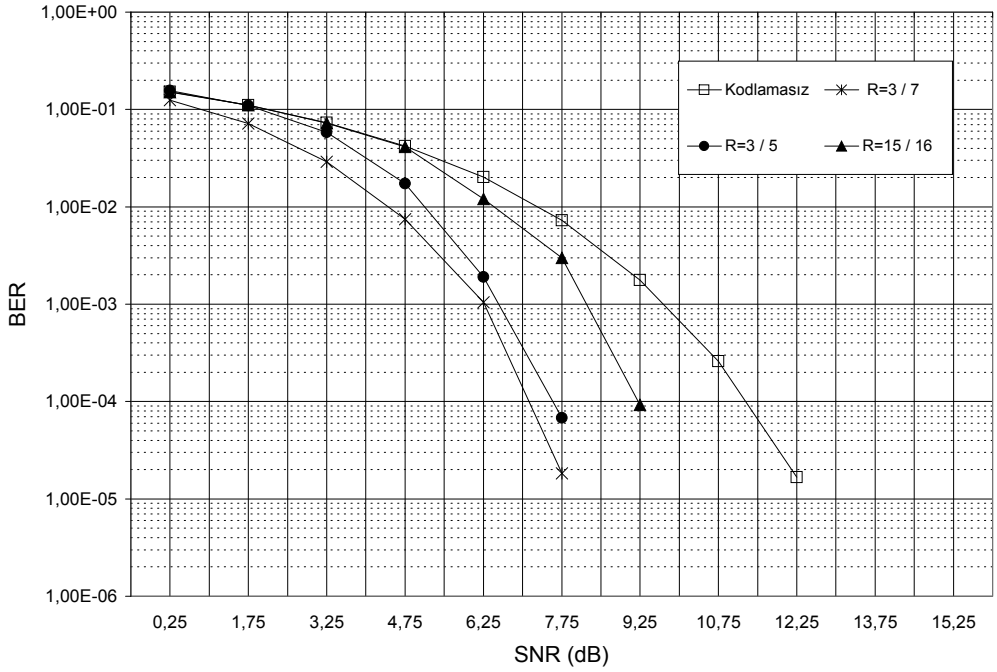
Şekil 27. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, BER Analizi [1000 sembol, 200 kanal, Rayleigh kanal, BPSK modülasyonu]

Şekil 27’de, Rayleigh kanaldan gönderilen reel datalar için farklı kod hızlarında elde edilen benzetimler verilmiştir. Modülasyon olarak BPSK kullanılmıştır. Farklı kod hızları arasında yaklaşık 1 dB’lik başarım farkı elde edilmiştir. Kodlamasız duruma göre en iyi sonucu veren RS (255,239) kodlama sisteminin kullanılması durumunda bit hata olasılığının 10^{-5} olduğu mertebelerde yaklaşık 8,6 dB kazanç elde edilmiştir.

Şekil 28’de AWGN kanaldaki QPSK başarımı gösterilmektedir.



Şekil 28. AWGN kanalda QPSK başarımı, BER Analizi

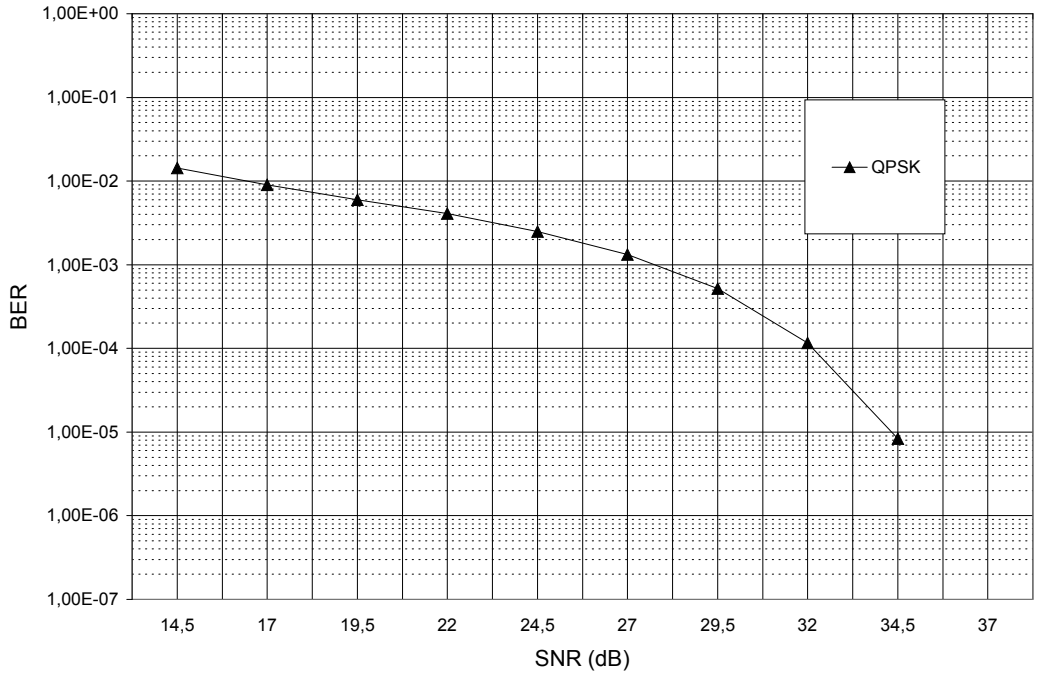


Şekil 29. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, BER Analizi [1000 sembol, 200 kanal, AWGN kanal, QPSK modülasyonu]

Şekil 29'da karmaşık veriler üzerinden AWGN kanalda farklı kod hızları için elde edilen benzetimler yer almaktadır. Modülasyon tipi QPSK olarak seçilmiştir. AWGN kanalda kod hızları arasında yapılan karşılaştırmada yaklaşık 2,15 dB'lik başarım farkı elde edildiği görülmektedir. Kodlamasız duruma göre en iyi sonucu veren RS (7,3) kodlaması yapılması durumunda ise bit hata olasılığı 10^{-5} mertebelerinde iken yaklaşık 4,5 dB kazanç elde edilmiştir.

Kanalın Rayleigh seçilmesi durumunda ise en iyi sonucu; diğer kanallarda elde edilenlerin tersine RS(255,239) kodlama sistemi vermiştir. Şekil 31'de bu durum gözlemlenebilmektedir.

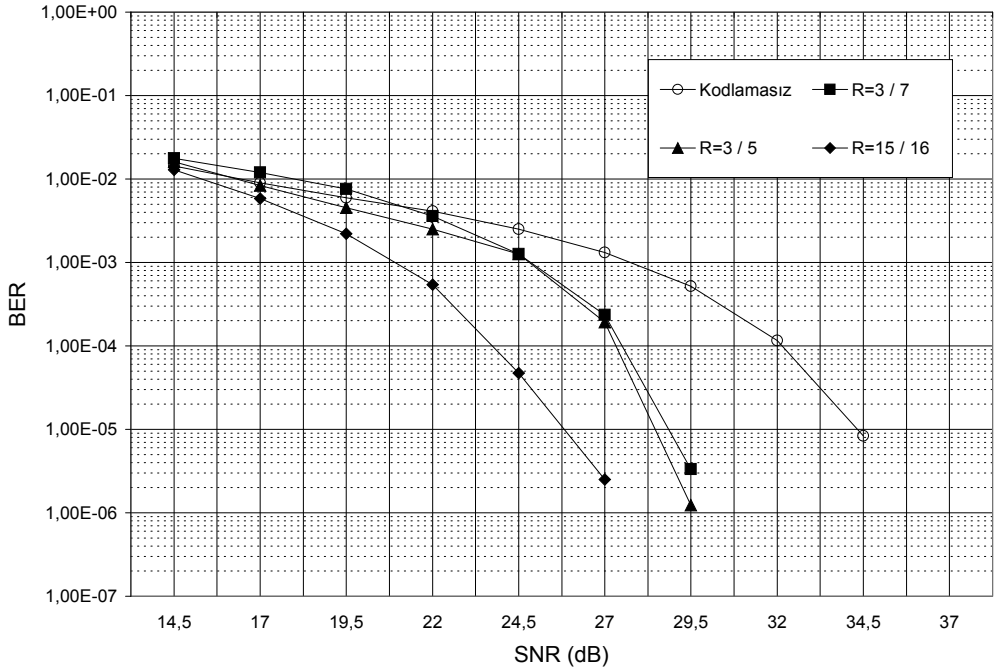
Şekil 30'da, Rayleigh kanaldaki QPSK başarımı gösterilmektedir.



Şekil 30. Rayleigh kanalda QPSK başarımı, BER Analizi

Şekil 31'de karmaşık veriler kullanılarak Rayleigh kanalda farklı kod hızları için elde edilen benzetimler yer almaktadır. Modülasyon tipi olarak QPSK seçilmiştir.

Bit hata olasılığının 10^{-5} mertebelerinde olduğu durumlarda kodlamasız duruma göre karşılaştırma yapılırsa yaklaşık 8,75 dB kazanç elde edildiği görülmektedir. Bununla birlikte farklı kod oranları arasında ise yaklaşık 3,25 dB'lik başarım farkı elde edilmiştir.

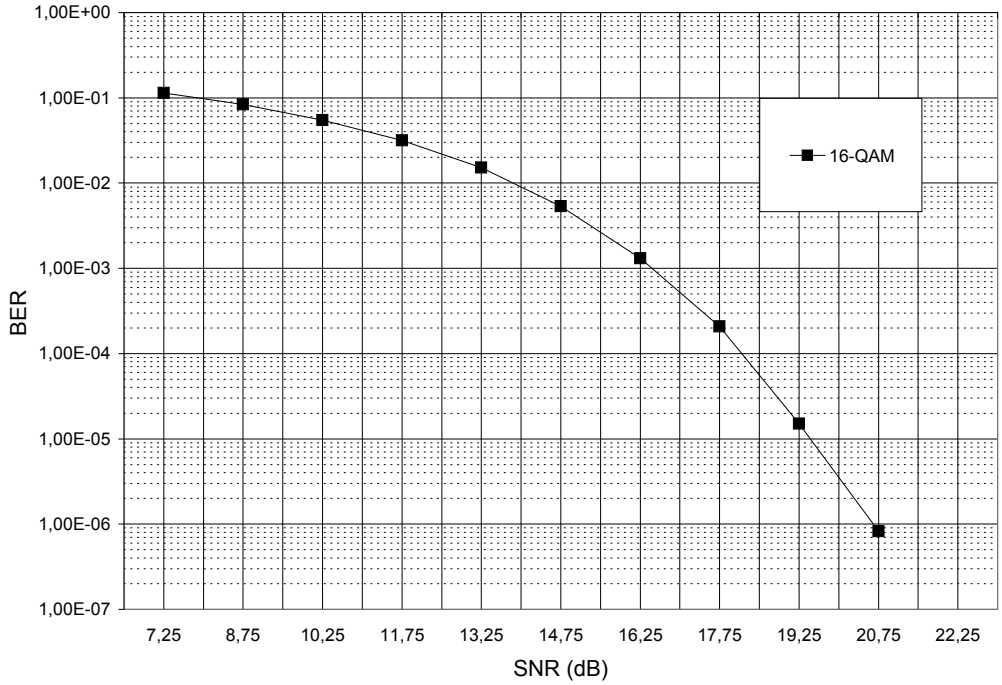


Şekil 31. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, BER Analizi [1000 sembol, 200 kanal, Rayleigh kanal, QPSK modülasyonu]

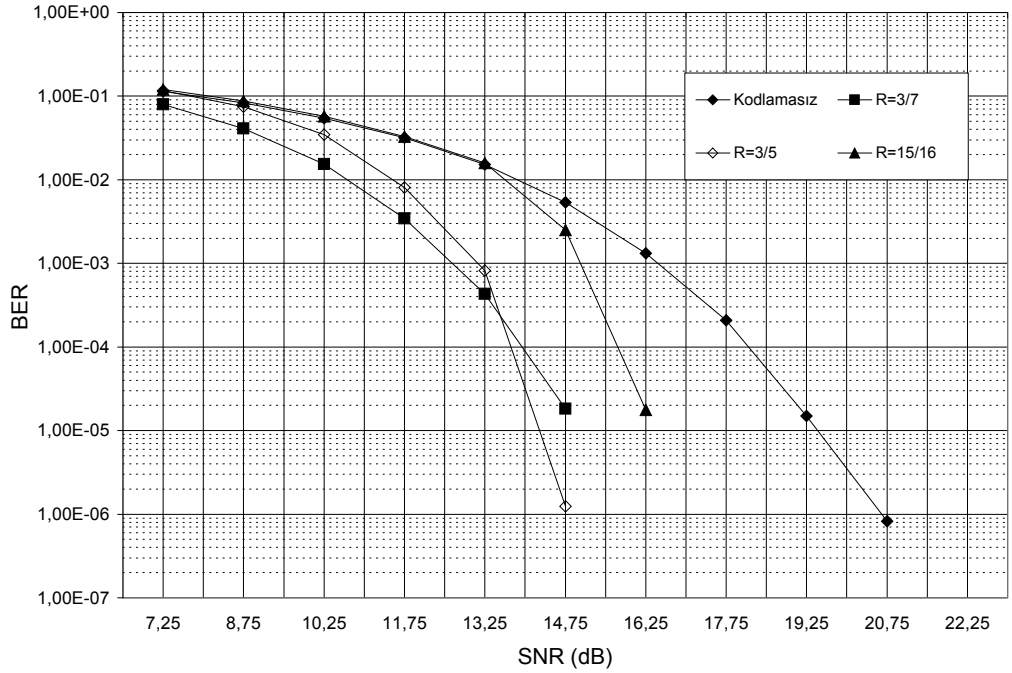
Şekil 32'de karmaşık veriler üzerinden AWGN kanaldaki 16-QAM başarımı verilmektedir.

Şekil 33'te ise karmaşık veriler kullanılarak AWGN kanalda farklı kod hızları için elde edilen benzetimler yer almaktadır. Modülasyon tipi olarak 16-QAM seçilmiştir.

Kodlamasız duruma göre en iyi sonucu veren RS (15,9) kodlama sisteminin kullanılması durumunda bit hata olasılığının 10^{-5} olduğu mertebelerde yaklaşık 5,25 dB kazanç elde edilmiştir. Bununla birlikte farklı kod oranları arasında ise yaklaşık 2,25 dB'lik başarım farkı elde edilmiştir



Şekil 32. AWGN kanalda 16-QAM başarımı, BER Analizi

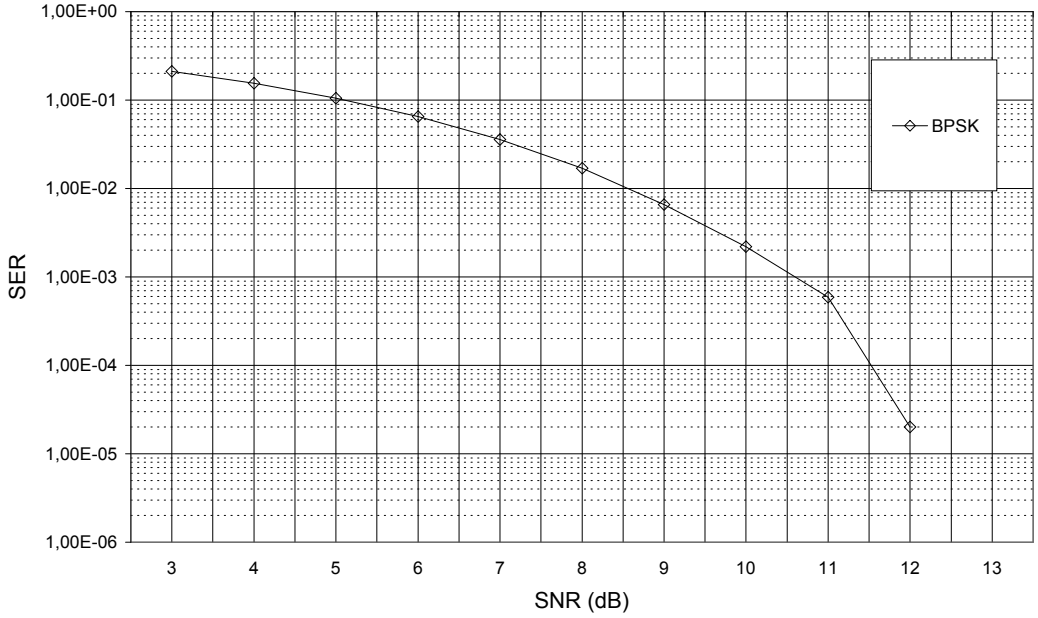


Şekil 33. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, BER Analizi [1000 sembol, 200 kanal, AWGN kanal, 16-QAM modülasyonu]

3.2. Reed Solomon Kod Çözücüde Kod Hızının Etkisi (SER Analizi)

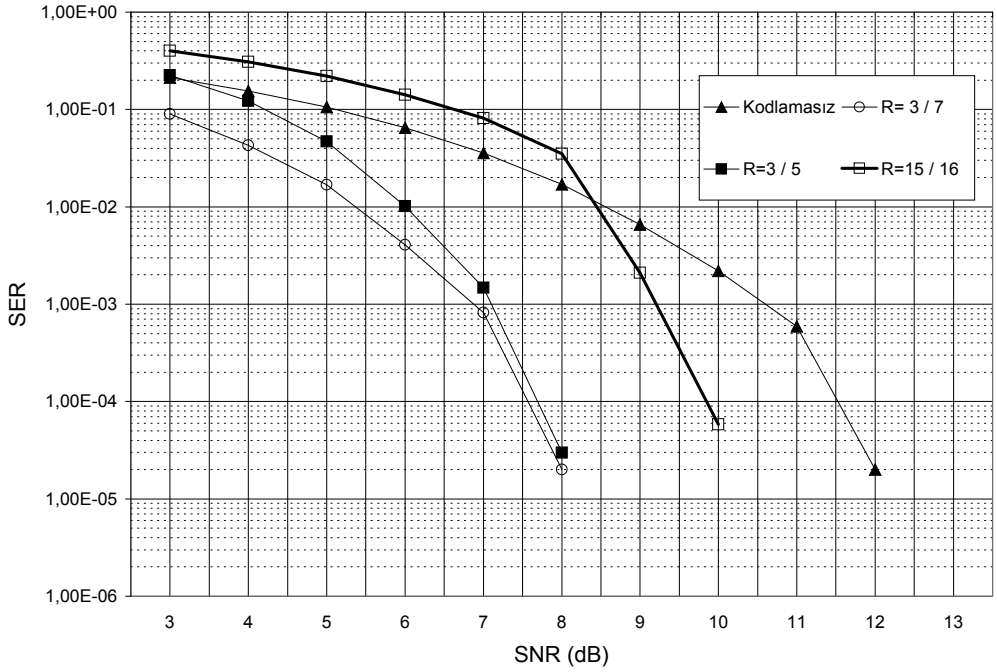
Reed – Solomon kodları diğer kodlardan farklı olarak semboller (ikili olmayan) üzerinden işlem yaptıklarından sembol hata oranı (SER) açısından da analiz edilmeleri gerekir. BER analizlerine benzer şekilde sırasıyla AWGN + BPSK, Rayleigh + BPSK, AWGN + QPSK, Rayleigh + QPSK ve AWGN + 16-QAM durumları için analizler yapılmıştır.

Şekil 34'de modülasyon türü BPSK olarak seçilen ve AWGN kanal üzerinden gönderilen reel veri sembollerinin kodlamasız durum için elde edilen benzetim sonucu verilmiştir.



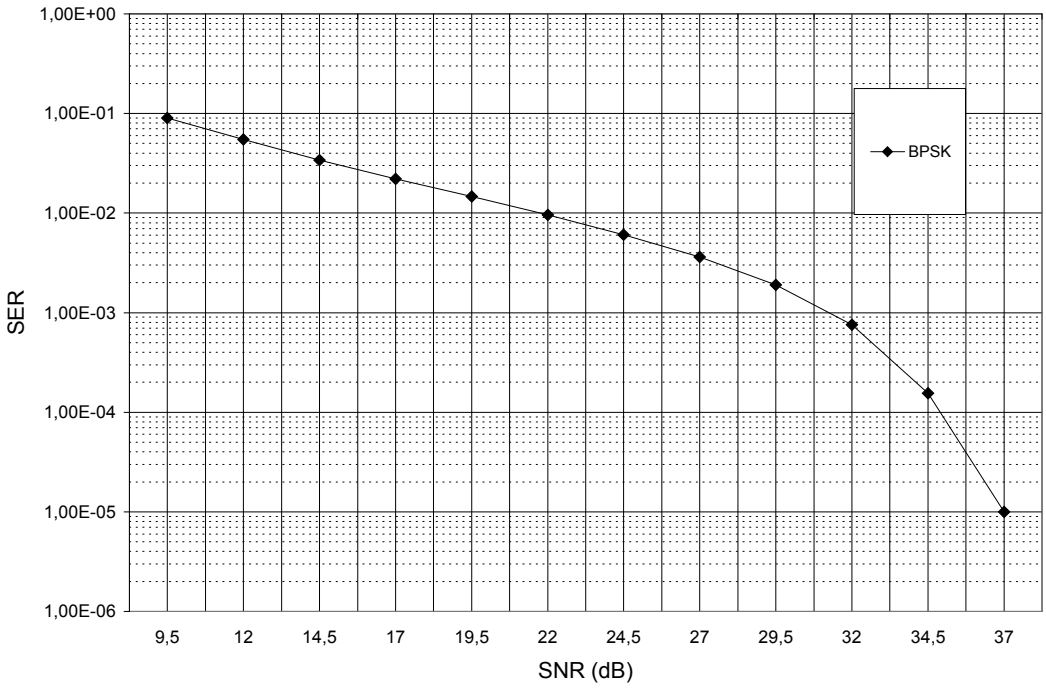
Şekil 34. AWGN kanalda BPSK başarımı, SER Analizi

Şekil 35'te ise farklı kod hızları için benzetim sonuçları elde edilmiştir. Yapılan analiz sonucunda farklı kod hızları kullanılması durumunda yaklaşık 2,2 dB başarım farkı elde edildiği gözlenmektedir. Kodlamasız durum ile karşılaştırıldığında ise; en iyi sonucu RS (7,3) kodlama düzeninin verdiği görülmektedir ve bu kodlama sistemi için 10^{-4} sembol hata olasılığı mertebelerinde başarım kazancı yaklaşık 3,5 dB'dir.



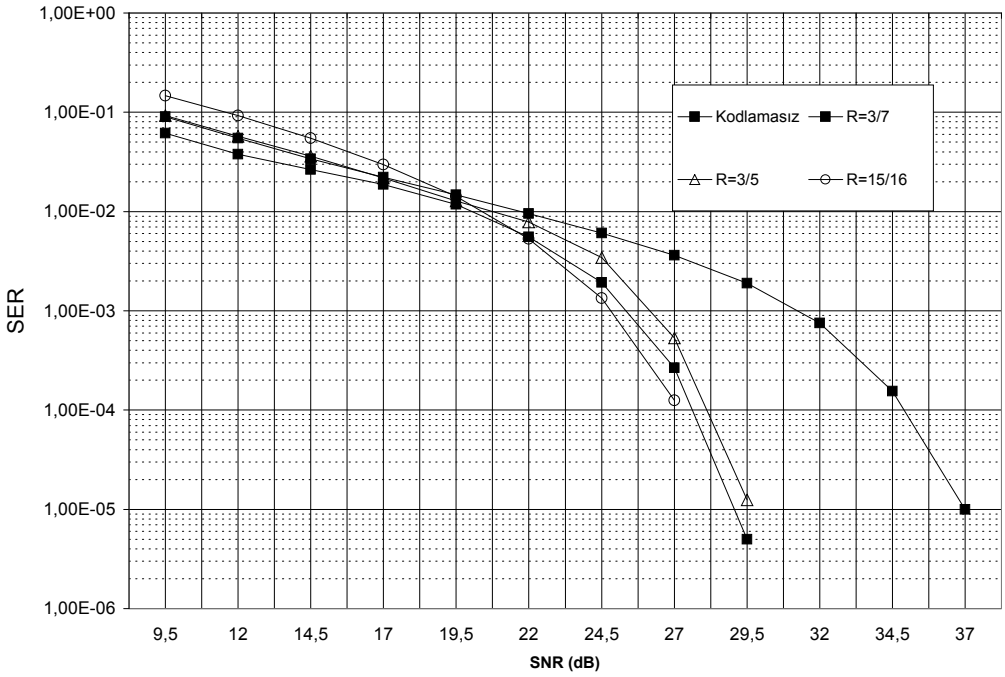
Şekil 35. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, SER Analizi [1000 sembol, 200 kanal, AWGN kanal, BPSK modülasyonu]

Şekil 36'da Rayleigh kanaldaki BPSK başarımı gösterilmektedir.



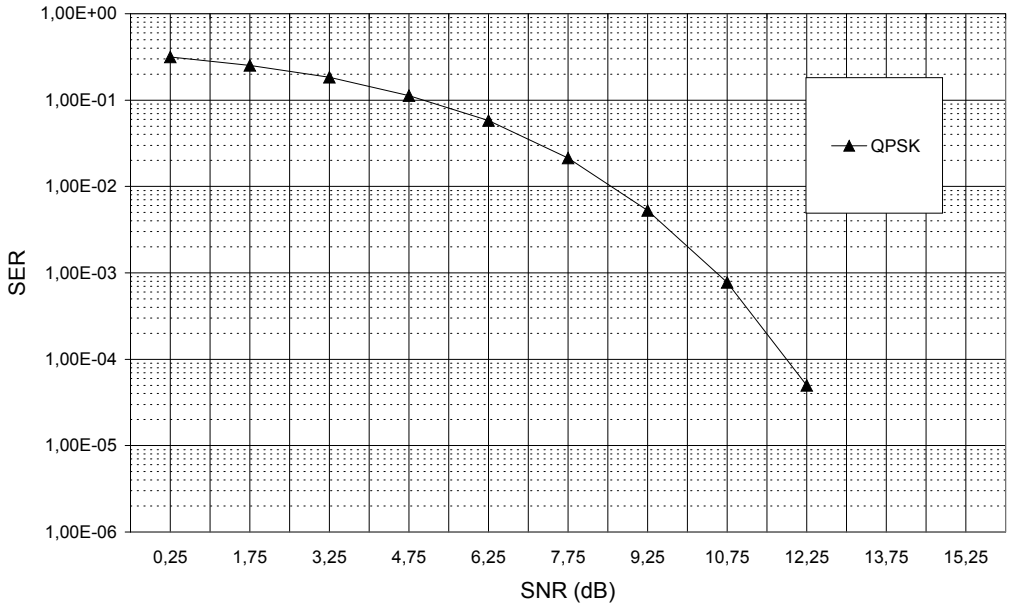
Şekil 36. Rayleigh kanalda BPSK başarımı, SER Analizi

Şekil 37’de karmaşık veriler kullanılarak Rayleigh kanalda farklı kod hızları için elde edilen benzetimler sonuçları yer almaktadır. Modülasyon tipi BPSK olarak seçilmiştir. Rayleigh kanalda kod hızları arasında yapılan karşılaştırmada yaklaşık 1 dB’lik başarım farkı elde edildiği görülmektedir. Kodlamasız duruma göre en iyi sonucu veren RS (255,239) kodlaması yapılması durumunda ise bit hata olasılığı 10^{-4} mertebelerinde iken yaklaşık 8 dB kazanç elde edilmiştir.



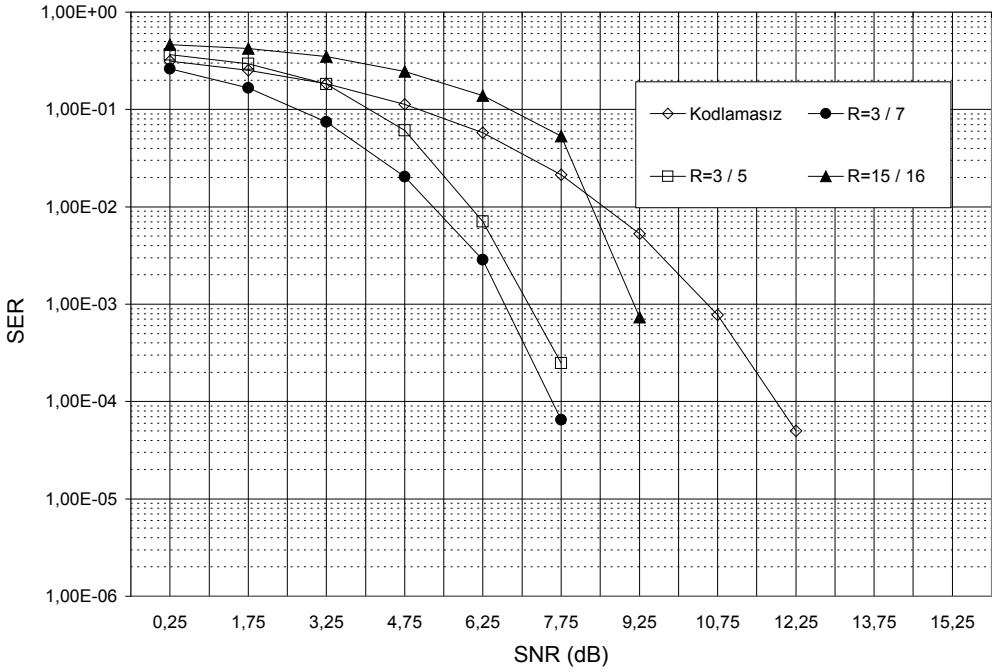
Şekil 37. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, SER Analizi [1000 sembol, 200 kanal, Rayleigh kanal, BPSK modülasyonu]

Şekil 38’de AWGN kanaldaki QPSK başarımı gösterilmektedir.



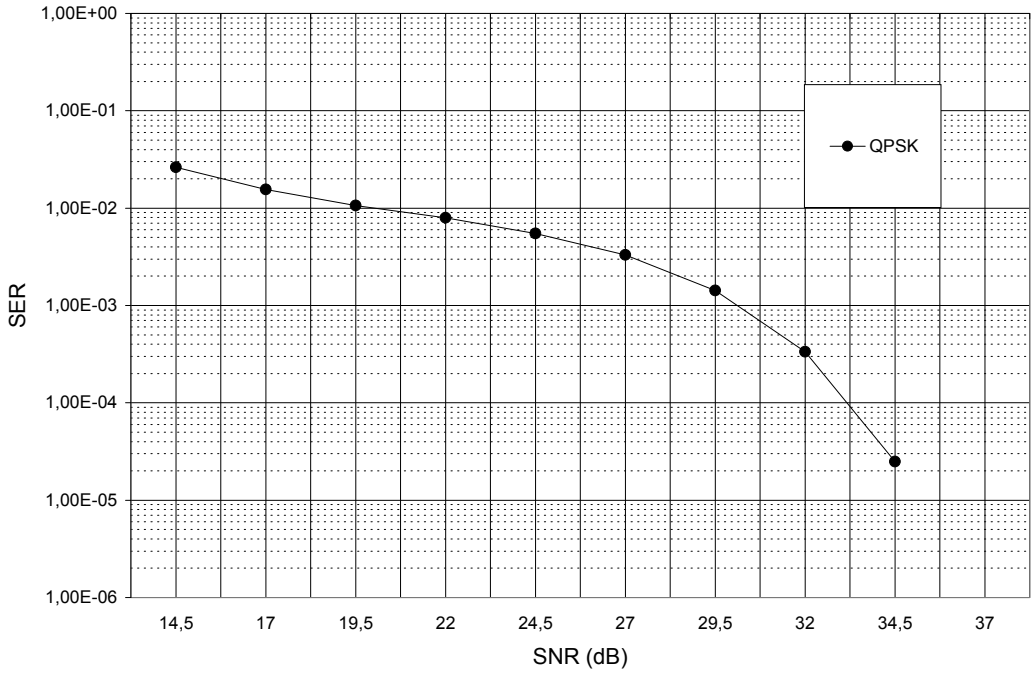
Şekil 38. AWGN kanalda QPSK başarımı, SER Analizi

Şekil 39'da karmaşık veriler kullanılarak AWGN kanalda farklı kod hızları için elde edilen benzetimler sonuçları yer almaktadır. Modülasyon tipi QPSK olarak seçilmiştir. AWGN kanalda kod hızları arasında yapılan karşılaştırmada yaklaşık 2 – 2,25 dB'lik başarım farkı elde edildiği görülmektedir. Kodlamasız duruma göre en iyi sonucu veren RS (7,3) kodlaması yapılması durumunda ise bit hata olasılığı 10^{-4} mertebelerinde iken yaklaşık 4,5 dB kazanç elde edilmiştir.



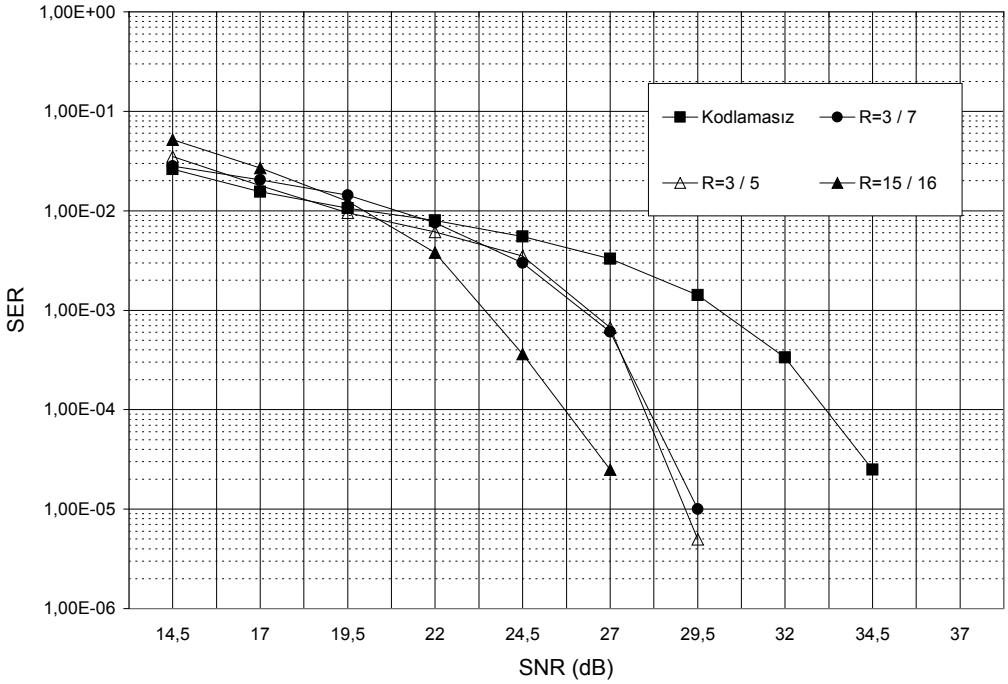
Şekil 39. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, SER Analizi [1000 sembol, 200 kanal, AWGN kanal, QPSK modülasyonu]

Şekil 40'ta Rayleigh kanaldaki QPSK başarımı gösterilmektedir.



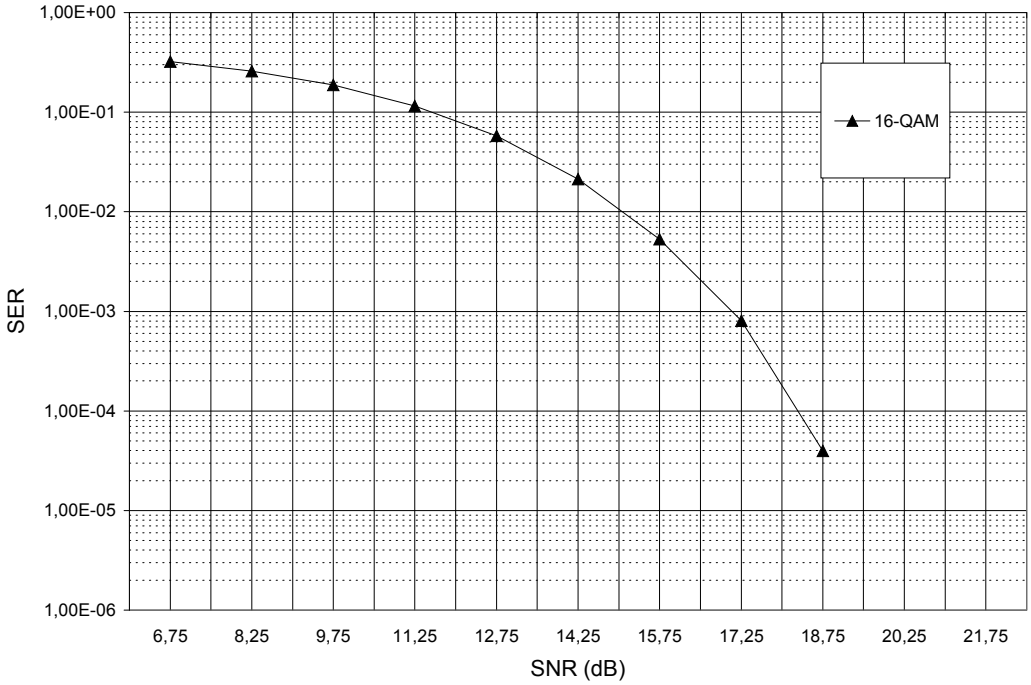
Şekil 40. Rayleigh kanalda QPSK başarımı, SER Analizi

Şekil 41’de Rayleigh kanalda modülasyon türü olarak QPSK seçilmesi durumunda farklı kod hızları için elde edilen benzetim sonuçları verilmiştir. Farklı kod hızları arasında başarım farkı 2 – 2,5 dB’dir. Kodlamasız durum ile karşılaştırıldığında en iyi sonucu veren RS(255,239) kodlama sistemidir. Sembol hata oranının 10^{-4} mertebelerinde başarım kazancı yaklaşık 7,5 dB’dir.



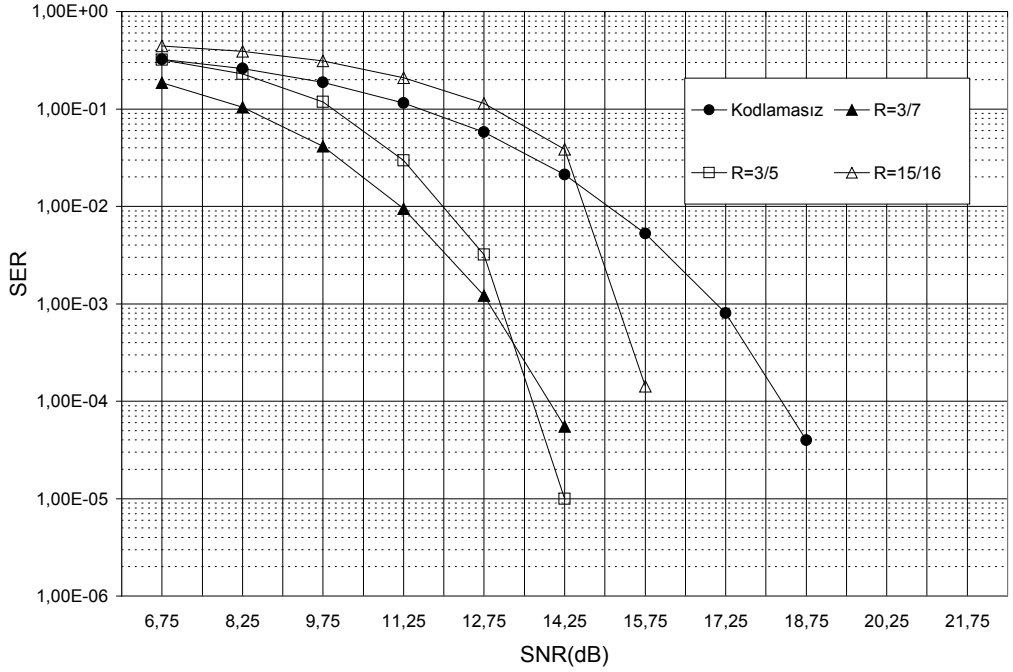
Şekil 41. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, SER Analizi [1000 sembol, 200 kanal, Rayleigh kanal, QPSK modülasyonu]

Şekil 42’de AWGN kanaldaki 16-QAM başarımı gösterilmektedir.



Şekil 42. AWGN kanalda 16-QAM başarımı, SER Analizi

Şekil 43'te AWGN kanalda modülasyon türü olarak 16-QAM seçilmesi durumunda farklı kod hızları için elde edilen benzetim sonuçları verilmiştir. Farklı kod hızları arasında başarım farkı yaklaşık 2,15 dB'dir. Kodlamasız duruma göre ise RS(15,9) kodlaması en iyi sonucu vermektedir. Sembol hata oranının 10^{-4} mertebelerinde başarım kazancı yaklaşık 4,6 dB'dir



Şekil 43. Kod hızının Reed – Solomon kod çözücü başarımına etkisi, SER Analizi [1000 sembol, 200 kanal, AWGN kanal, 16-QAM modülasyonu]

Reed – Solomon kod çözücünün toplanır beyaz Gauss gürültülü kanallardaki başarımı, benzetim sonuçlarından da anlaşılacağı üzere kod hızı azaldıkça gelişmektedir. Yani kod hızının en düşük olduğu hem reel hem de karmaşık data gönderilmesi durumlarında başarım kazancı daha yüksek olmaktadır. Modülasyon türünün kod çözücü başarımına etkisi ise benzetim eğrilerinden görüleceği gibi QPSK modülasyonu kullanılan AWGN kanaldaki başarım, BPSK kullanılan kanaldaki başarıma oranla daha iyi sonuç vermektedir. Ayrıca AWGN kanallarda daha iyi sonuç veren RS (7,3) kodlama sistemi kod çözücünün işlem yükünü artırmaktadır. Bu da dizi uzunluğunu sınırlamaktadır.

Rayleigh kanallarda ise kod çözücü başarımını etkileyen etmenlerden biri olan kod hızı arttıkça başarım giderek daha iyi sonuç vermektedir. Hem sembol bazında hem de bit bazında Rayleigh kanallarda kod hızının artması, elde edilen başarım kazancını artırmaktadır. Kod hızı en yüksek olan RS (255,239) kodlama sistemi, kod çözücünün işlem yükünü hafifletmekle birlikte daha çok dizinin gönderilmesine imkân tanır. Modülasyon türü seçiminin başarıma etkisi AWGN kanallardaki gibidir. En iyi sonucu QPSK modülasyonu kullanıldığında vermektedir. Yani modülasyon derinliğinin artması kodlama yapılması durumunda başarımı iyileştirmektedir. Ancak bu durum bant genişliğinin de azalmasına neden olur.

4. SONUÇLAR

Bu çalışma Reed – Solomon (RS) kodlarının AWGN ve Rayleigh kanallardaki başarımlarını sergilemiştir. Temel BPSK, QPSK ve 16-QAM modülasyonlarının AWGN ve Rayleigh kanallardaki BER ve SER başarımlarını vermektedir.

Çalışmanın önemli bir bölümünü standart RS kodlama, Galois alanı tanımlama ve kanal simülasyon programlarını geliştirmek almıştır. Gerekli literatür taraması ise giriş bölümünde verilmiştir.

Reed - Solomon kodlar güç sınırlı haberleşme sistemlerinde büyük bir kazanç sağlarken bir taraftan da serpiştirme ve iteratif kod çözme nedeniyle sistemde bir gecikmenin oluşmasına neden olur.

Reed - Solomon kodlar, işlem karmaşıklığı ve sistemde bir gecikmeye neden olması gibi iki olumsuz yönüne rağmen, kodlama alanında ikili olmayan semboller üzerinden işlem yaptıklarından diğer kodlama türleri ile birlikte kullanılmaları durumunda daha iyi sonuçlar vermektedirler. İşlemci hızlarının da artması ile uygulanabilirlikleri kolaylaşmıştır.

Tablo 10. BER = 10^{-5} için gerekli SNR değerleri (dB)

Kanal / Modülasyon	Simülasyon Durumları	AWGN	RAYLEIGH
BPSK	Kodlamasız	11,8	36
	R= 3/7	8,4	28,4
	R= 3/5	9	28,5
	R= 15/16	9,2	27,4
QPSK	Kodlamasız	12,25	34,5
	R= 3/7	7,85	29
	R= 3/5	8,5	28,35
	R= 15/16	10	25,75
16-QAM	Kodlamasız	19,5	
	R= 3/7	15	
	R= 3/5	14,25	
	R= 15/16	16,45	

Tablo 11. SER = 10^{-4} için gerekli SNR değerleri (dB)

Modülasyon \ Kanal	Simülasyon Durumları	AWGN	RAYLEIGH
BPSK	Kodlamasız	10,5	35
	R= 3/7	7,6	27,65
	R= 3/5	7,75	28,25
	R= 15/16	9,8	27
QPSK	Kodlamasız	11,85	33,25
	R= 3/7	7,5	28,25
	R= 3/5	8,15	28,15
	R= 15/16	9,75	25,75
16-QAM	Kodlamasız	18,35	
	R= 3/7	14	
	R= 3/5	13,75	
	R= 15/16	15,9	

5. ÖNERİLER

Kablosuz haberleşme sistemlerindeki en önemli problem, sınırlandırılmış verici gücü ile veri kaybı olmadan haberleşebilme uzaklığını artırmaktır. Kodlayıcıların önemi, bu noktada bariz olarak ortaya çıkmaktadır. HIPERLAN/2 (*High Performance Local Area Network*) standardında verici gücü 200mW (23dBm) olarak belirlenmiştir. Yapılan haberleşmenin türüne göre de (ses, veri, vs.) bit-hata oranı belirli bir değerin altında olmak zorundadır.

Kodlama olmadan, 23 dBm'lik verici gücü ve 10^{-3} bit-hata oranı ile 50 m mesafede haberleşme yapıldığı varsayılsın. Kodlama kazancının 6 dB olduğu durumda, aynı mesafede, aynı-bit hata oranında haberleşme yapabilmek için gerekecek verici gücü 6 dB azalmış olacaktır. Benzer şekilde verici gücü aynı değerde tutulacak olursa, aynı bit hata oranında haberleşme yapılabilecek mesafe artacaktır. Bir başka seçenek ise modülasyon derinliğini artırarak aynı mesafe, aynı verici gücü ve aynı bit-hata oranında daha yüksek veri hızına çıkmaktır.

Yapılan bu çalışmada, Reed – Solomon kod çözücünün AWGN ve dar bantlı Rayleigh kanaldaki başarımı incelenmiştir. HIPERLAN/2, BLUETOOTH veya WİMAX standartlarında Reed – Solomon kod çözücünün kullanılabilmesi için, bu kodun, geniş bantlı Rayleigh ve Rician kanallar için de başarımının incelenmesi gerekmektedir. Reed – Solomon çözücü kullanımı ile elde edilecek kod kazancı, HIPERLAN/2, BLUETOOTH veya WİMAX sistemlerinin başarımını artırmada etkili olabilecek düzeyde çıkarsa, ki muhtemelen çıkacaktır, sisteme eklenebilir. Böylece, bu standartları kullanan sistemlerin haberleşme mesafesi veya veri hızı artırılabilir. Ayrıca bu sistemler için gerekli olan verici gücü de azaltılmış olur.

6. KAYNAKLAR

1. Viterbi, A.J. ve Omura,A.J., Principles of Digital Communication and Coding, McGraw-Hill Co., New York, 1979.

2. McEliece, R.J., The Theory of Information and Coding, Addison-Weley Publishing Company, New York, 1977.

3. Rappaport, T.S., Wireless Communications: Principle and Practice, Prentice Hall Inc., Upper Saddle River, 1996.

4. Shannon, C.E., A Mathematical Theory of Communications, Bell Syst. Tech. J., 27 (1948) 379-423.

5. Hamming, R.W., Error Detecting and Error Correcting Codes, Bell Syst. Tech. J., 1950.

6. Sklar, B., Digital Communications: Fundamentals and Applications, Prentice-Hall Inc., Englewood Cliffs, 1988.

7. Proakis, J.G., Digital Communications, Third Edition, McGraw-Hill Co., 1995.

8. Elias, P., Coding for Noisy Channels, IRE Conv. Record, 4 (1955) 37-47.

9. Wozencraft J.M. ve Reiffen, B., Sequential Decoding, MIT Press, Cambridge, 1961.

10. Fano, R.M., A Heurstic Discussion of Probabilistic Decoding, IEEE Transactions on Information Theory, 9 (1963) 64-74.

11. Jelinek, F., An Upper Bound on Moments of Sequential Decoding Effort, IEEE Transactions Information Theory, 15 (1969) 464-468.

12. Viterbi A.J., Error Bounds for Convolutional Codes and An Asymptotically Optimum Decoding Algorithm, IEEE Transactions on Information Theory, 13 (1967) 260-269.

13. Odenwalder, J.P., Error Control Coding Handbook, Linkabit, 1976.

14. Wicker, S., Error Control Systems for Digital Communications and Storage, Prentice Hall Inc., Englewood Cliffs, 1995.

15. Costello, D.J., Hagenauer, J., Imai, H. ve Wicker, S.B., Applications of Error-Control Coding, IEEE Transactions on Information Theory, 44 (1998) 2531-2560.

16. Forney, G.D., Concatenated Codes, MIT Press, Cambridge, 1966.

17. Wilson, S.G., Digital Modulation and Coding, Prentice-Hall Inc., Upper Saddle River, 1996.

18. Massey, J.L., The How and Why of Channel Coding, Proceeding of 1984 Zurich Seminar on Digital Communications, 1984, Zürih, 67-73.

19. Wicker, S.B. ve Bharagava, V.K., Reed-Solomon Codes and Their Applications, IEEE Press, New York, 1994.

20. Berrou, C., Glavieux, A. ve Thitimajshima, P., Near Shannon Limit Error-Correction Coding and Decoding: Turbo Codes (1), Proc. 1993 IEEE Int. Conf. on Communication Geneva, 1993, İsviçre, 1064-1070.

21. Robertson, P., Illuminating The Structure of Parallel Concatenated Recursive Systematic (Turbo) Codes, GLOBECOM'94, Kasım 1994, San Francisco, Cilt III, 1298-1303.

22. Hagenauer, J., Robertson, P. ve Papke, L., Iterative (Turbo) Decoding of Systematic Convolutional Codes With MAP and SOVA Algorithms, Source and Channel Coding ITG Conference, Ekim 1994, Frankfurt, Almanya, 1-9.

23. Benedetto, S. ve Montorsi, G., Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes, IEEE Transactions on Information Theory, 42, 2 (1996) 409-428.

24. Benedetto, S. ve Montorsi, G., Design of Parallel Concatenated Convolutional Codes, IEEE Transactions on Information Theory, 44, 5 (1996) 591-600.

25. Valentini, M.C. ve Woerner, B.D., Turbo Codes and Iterative Processing, IEEE New Zealand Wireless Communications Symposium, Kasım 1998, Auckland, New Zealand, 16-24.

26. Papoulis A., Probability, Random Variables and Stochastic Processes, McGraw Hill Co, New York, 1991.

27. Collins, O.M., The Subleties and Intracies of Building A Constraint Length 15 Convolutioanl Decoder, IEEE Transactions on Communication, 40 (1992) 1810-1819.

28. Reed, I.S. ve Solomon, G., Polynomial Codes Over Certain Finite Fields, SIAM Journal on Applied Mathematics, 8 (1960) 300-304.

29. Perez, L.C., "Turbo Codes" In Trellis Coding, IEEE Press, New York, 1997.

30. Cover, T.M. ve Thomas, J.A., Elements of Information Theory, Wiley Series in Communications, 1998.

31. Summers, T.A. ve Wilson, S.G., SNR Mismatch and online estimation in turbo decoding, IEEE Transactions on Communication, 46 (1998) 421-423.

32. Tepe, K. ve Anderson, J., Turbo Codes for Binary Symmetric and Binary Erasure Channels, ISIT 1998, Ağustos 1998, Cambridge, USA, 59.

33. Hall, E.K. ve Wilson, S.G., Design and Analysis of Turbo Codes on Rayleigh Fading Channels, IEEE Selected Areas in Communications, 16, 2 (1998) 160-174.

34. Massey, J.L., Threshold Decoding, MIT Press, Cambridge, 1961.

35. Jelinek, F., A Fast Sequential Decoding Algorithm Using a Stack, IBM Journel of Research and Development, 13 (1969) 675-685.

36. Forney, G.D.JR., The Viterbi Algorithm, Proceedings of the IEEE, 1973, Cilt 61, 268-278

37. Lin, S. ve Costello, D.J., Error Control Coding: Fundamentals and Applications, Prentice-Hall Inc., Englewood Cliffs, 1983.

38. Forney, G.D.JR., Convolutional Codes I: Algebraic Structure, IEEE Transactions on Information Theory, 16 (1970) 720-738.

39. Sagan, C., Pale Blue Dot, Random Hause, New York, 1994.

40. Barbulescu, A.S. ve Pietrobon, S.S., Interleaver Design for Turbo Codes, IEEE Electronics Letters, 30, 25 (1994) 2107-2108.

41. Barbulescu, A.S. ve Pietrobon, S.S., Turbo Codes: A Tutorial on A New Class of Powerful Error Correcting Schemas, Part 1: Code Structures and Interleaver Design, J. Elec. and Electron. Eng., Avustralya, 1999, Cilt 19, 129-141.

42. Gorenstein, D. and Zierler, N., A Class of Error Correcting Codes in p^m Symbols, Journal of the Society for Industrial and Applied Mathematics, December 1960

43. Berlekamp, E.R., Algebraic Coding Theory, McGraw Hill, 1968

44. Chien, R.T., Cyclic Decoding Procedure for the Bose Chaudri Hocquenghem Codes, IEEE Transactions on Information Theory, pp.357-363, September 1964

45. Massey, J.L., Shift – Register Synthesis and BCH Decoding, IEEE Transactions on Information Theory, IT – 15, pp.122-127 , 1969

46. Hocquenghem, A.I., Codes Correcteurs d'erreurs, Chiffres, 2, pp. 147-156 ,1959

47. Reed, S.I. and Solomon, G., Polynomial Codes Over Certain Finite Fields, Journal of the Society for Industrial and Applied Mathematics, 8, pp. 300 -304, June 1960

48. Kayran, A.H., Panayırcı, E., Aygölü , Ü., Sayısal Haberleşme , pp. 131 – 151 Sistem Yayıncılık 1992

49. Seymour , S. and Jones , J.J., Modern Communications Principles, pp. 237 – 255 , Mc.Graw Hill, 1967

50. Charles , L., Error Control Block Codes for Communications Engineers, Artech House Telecommunications Library , February 2000

51. Michelson , A. K., and Levesque , A.H., Error Control Techniques for Digital Communication, Wiley – Interscıence Publication , 1985

52. Sweeney , P., Error Control Coding : An Introduction , Prentice Hall Int. , 1991

53. Peterson , W.W., Encoding and Error Correcting Procedures for the Bose Chaudri Codes , IRE Transactions on Information Theory , September 1960

ÖZGEÇMİŞ

Yusuf ZORLU, 22.07.1978 tarihinde Erzincan' ın Tercan ilçesinde dünyaya geldi. İlk öğrenimini 17 Şubat İlköğretim Okulu' nda, orta öğrenimini Tercan Lisesi' nde, lise öğrenimi ise Zonguldak Kozlu Lisesi' nde tamamladı. 1998–1999 eğitim-öğretim yılında, Karadeniz Teknik Üniversitesi Mühendislik Mimarlık Fakültesi Elektrik-Elektronik Mühendisliği Bölümü' nü kazandı. 2003 yılında bu bölümden ikincilikle mezun oldu. Aynı yıl Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı' nda yüksek lisans öğrenimine başladı. 2004 yılı Aralık ayından itibaren TEDAŞ Ankara Başkent Elektrik Dağıtım A.Ş de Elektrik-Elektronik Mühendisi olarak görev yapmaktadır. Yabancı dil olarak İngilizce bilmektedir.