T.C.
NEVŞEHİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SELF-DUAL KODLAR VE İNŞA YÖNTEMLERİ

Tezi Hazırlayan
Hatice TOPCU

Tezi Yöneten
Doç. Dr. Hacı AKTAŞ

Matematik Anabilim Dalı
Yüksek Lisans Tezi

Ocak 2012
NEVŞEHİR

T.C.
NEVŞEHİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SELF-DUAL KODLAR VE İNŞA YÖNTEMLERİ

Tezi Hazırlayan
Hatice TOPCU

Tezi Yöneten
Doç. Dr. Hacı AKTAŞ

Matematik Anabilim Dalı
Yüksek Lisans Tezi

Ocak 2012
NEVŞEHİR

Doç. Dr. Hacı AKTAŞ danışmanlığında **Hatice TOPCU** tarafından hazırlanan **"Self-Dual Kodlar ve İnşa Yöntemleri"** adlı bu çalışma, jürimiz tarafından Nevşehir Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalında **Yüksek Lisans Tezi** olarak kabul edilmiştir.

30.01.2012

JÜRİ:

Başkan : Doç. Dr. Necdet BATIR

Üye : Doç. Dr. Naim ÇAĞMAN

Üye : Doç. Dr. Hacı AKTAŞ

ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulunun 02.03.2012 tarih ve 2012-20/1 sayılı kararı ile onaylanmıştır.

02/03/2012

Prof. Dr. Selçuk KERVAN
Enstitü Müdürü

TEŞEKKÜR

Tez çalışmamın her aşamasında bilgi ve tecrübeleriyle beni yönlendiren sayın danışmanım Doç. Dr. Hacı AKTAŞ'a, tezimle ilgili kendisiyle çalışma fırsatı bulduğum Doç. Dr. Jon-Lark KIM'e, eğitim hayatım boyunca benden maddi manevi desteğini esirgemeyen sevgili aileme ve bu süreçte en büyük manevi desteğim olan sevgili eşime en içten teşekkürlerimi sunarım.

SELF-DUAL KODLAR VE İNŞA YÖNTEMLERİ

Hatice TOPCU
Nevşehir Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Ocak 2012
Tez Danışman: Doç. Dr. Hacı AKTAŞ

ÖZET

Hata düzeltilmesinde etkili sonuçlar vermesi, oldukça geniş yapılar üzerinde inşa edilebilmesi, kodlama teorisinin başlangıcından bu yana kullanılmış ve halen kullanılmakta olan uzatılmış Hamming kod, uzatılmış Golay kod gibi bilinen önemli kodları barındırıyor olması gibi çeşitli nedenlerden dolayı self-dual kodlar, kodlama teorisyenlerinin oldukça ilgisini çeken ve günümüz bilgi ve iletişim teknolojilerinin birçok alanında kendine yer edinmiş olan bir kod sınıfıdır. Literatürde, bu sınıfa ait çok sayıda çalışmaya rastlanmasına rağmen halen daha belirli parametrelere ya da özelliklere sahip birçok self-dual kod bulunamamış ve yine belirli parametrelere sahip self-dual kodların sınıflandırılması tamamlanamamıştır.

Bu tez çalışmasında, yukarıda kısaca bahsedilen self-dual kodlar özellikle de inşa yöntemleri üzerinde durulmuştur. Günümüzde üzerinde yoğun bir şekilde çalışılan self-dual kodlar ve inşa yöntemleri geniş bir literatür taramasıyla araştırılmış ve derlenerek elde edilen bulgular bu çalışma boyunca verilmiştir.

Bu çalışma genel olarak aşağıdaki gibi düzenlenmiştir. Öncelikle kodlama teorisi ile ilgili genel bilgi verilmiş ve lineer kodlar tanıtılmıştır. Daha sonra lineer kodların önemli bir sınıfı olan self-dual kodlar ailesine ve özelliklerine yer verilmiştir. Genel kod inşa metodları ile özel olarak self-dual kod inşa eden çeşitli metodlar incelenmiştir. Bu metodlar kullanılarak self-dual kodlar elde edilmeye çalışılmış ve elde edilen kodlar ile literatürde var olan bazı kodlara ait tablo ve matrislere yer verilmiştir.

Son olarak, self-dual kod inşa eden iki özel metod olan yinelemeli algoritma ve üst-yapı inşa metodu karşılaştırılmış ve bu metodlardan birinin aslında diğerinin özel bir hali olduğu ispatlanmıştır.

Anahtar Kelimeler: Hata düzelten kodlar, self-dual kodlar, kod inşa yöntemleri.

SELF-DUAL CODES AND THE CONSTRUCTION METHODS

Hatice TOPCU
Nevşehir University, Graduate School of Natural and Applied Sciences
M.Sc. Thesis, January 2012
Thesis Supervisor: Assoc. Prof. Dr. Hacı AKTAŞ

ABSTRACT

Self-dual codes is an important class of linear codes which has been received attention by researchers since the beginning of the coding theory because of various reasons that are effective results in error-correction, construction on the large structures and containing well-known good codes such as extended Hamming code and extended Golay code. These codes have an important part in today's communication and information technology. Although many papers of self-dual codes have seen in the coding theory literature, stil many self-dual codes which has determinate parameters are unknown and also classifications of this type codes with determinate parameters are incomplete.

In this thesis, self-dual codes which is briefly explained above and the construction methods of this type codes are examined. Self-dual codes and the construction methods of this type codes are investigated widely in the literature and findings obtained from this investigation are compiled in this study.

Arrangement of this thesis is as follows. Firstly, basic concepts of coding theory and linear codes are introduced. Then, family of self-dual codes which is an important class of linear codes and properties are given. Also, some general construction methods and some special construction methods generated self-dual codes are examined. Self-dual codes are constructed by using these methods and codes obtained and some known codes from the literature are mentioned with their generator matrices and tables.

Finally, two of these special methods, building-up construction and recursive algorithm are compared in this thesis and it has proved that recursive algorithm is actually a special case of building-up construction for the binary self-dual codes .

Keywords: Error-correcting codes, self-dual codes, construction methods.

İÇİNDEKİLER

1. BÖLÜM

GİRİŞ

Kodlama Teorisinin ya da diğer bir adıyla Hata Düzelten Kodlar Teorisinin konusunu iletişimde dijital olarak kodlanmış verilerin kullanımı esnasında meydana gelen problemler ve çözümleri oluşturmaktadır. 1948 yılında Claude E. Shannon tarafından yayınlanan "Haberleşmenin Matematiksel Teorisi" [15] isimli makale Bilgi Teorisinin ve Kodlama Teorisinin başlangıç noktası olarak kabul edilir. Shannon bu makalesinde transfer edilen veride bozulmanın meydana gelebileceği bir iletişim kanalı için kanal kapasitesini tanımlamış ve bu kapasitenin altında herhangi bir güvenilirlik düzeyinde iletişim sağlanmasının mümkün olduğunu olasılık kuramına ait metodlar kullanarak ispatlamıştır. Telefon, telgraf, uzaydan veri transferi sağlayan cihazlar, manyetik kayıt cihazları v.b. bu tür iletişim kanallarına örnek olarak verilebilir. Bu kanalların ortak özelliği, verinin belirli bir kaynaktan çıkıp kanal yolu ile diğer taraftaki alıcıya iletilmesidir. Eğer alınan veri, gönderilen veriden farklı ise veride meydana gelen bu bozulmanın sebebi "gürültü" olarak adlandırılır ve bu kanal "gürültülü"dür denir.

Yukarıda örnekleri verilen iletişim kanallarının gerçek hayatta kullanımına bakılırsa, kanalların gürültüye maruz kalabilecekleri görülmektedir. Dolayısıyla burada temel problem alınan veriyi gönderilen veriye mümkün olduğunca benzer bir şekilde belirleyebilmektir. Bunun gerçekleştirilebilmesi için temel düşünce gönderilen mesaj üzerinde çeşitli değişiklikler yapılarak alınan mesajın gönderilen mesaja oldukça yakın (hatta aynı) biçimde çözülebilmesinin sağlanmasıdır. Bu amaca ulaşabilmek için yapılan en yaygın değişiklik mesajın doğruluğunu kontrol edebilecek biçimde mesaja ekleme yapılmasıdır. Burada ilk akla gelebilecek örneklerden biri mesajın çok defa tekrar edilerek gönderilmesi ve gelen veride çoğunluğa bakılarak gönderilen mesajın çözülmeye çalışılmasıdır. Elbette bazen hatalı sonuçlar olabilir fakat buna rağmen bu yolla elde edilen sonuçlar daha güvenilir olacaktır, olasılık kuramına ait metodlarla kodun doğru çözülme ihtimalinin arttığı da ayrıca görülebilir [1]. Alınan mesajların güvenilirliğindeki bu artış ise mesajın transferi için harcanan uzun zaman, tek bir veri transferi için kanalın çok defa kullanımı v.b. gibi bedeller

gerektirmektedir. Dolayısıyla mesaj üzerinde yapılacak değişikliklerde bu bedelleri olabildiğince düşük tutmaya çalışarak mesajdaki güvenilirliği arttırmak gerekmektedir.

Kodlama Teorisinin geçmişine baktığımızda, ilk etapta genel olarak binary(ikilik) kodların üzerinde yoğunlaşıldığı görülebilir. Binary self-dual kodlar için, ilk olarak Pless [17] de $n \leq 20$ olmak üzere n uzunluklu kodların bir sınıflandırmasını yapmıştır. $n = 22, 24$ uzunluklu kodlar için Pless ve Sloane tarafından [21] de bir sınıflandırma yapılmış, daha sonra Pless ve Conway [19] da uzunluğu 26 ile 30 arasında olan kodların sınıflandırmasını yapmış ve ayrıca burada 32 uzunluklu Tip 2 kodlar elde edilmiştir. Pless [18] de 32 uzunluklu kodlar ile ilgili [19] da yapılan çalışmayı tamamlamış ve [20] de Conway, Pless ve Sloane 32 uzunluğuna kadar binary self-dual kodların sınıflandırılması ile ilgili daha önce yapılan bütün çalışmaları revize eden yeni bir çalışma ortaya koymuşlardır. Günümüzde ise binary kodlar artan bir öneme sahip olmasına rağmen çeşitli diğer yapılar üzerindeki kodlar hem Matematik hem de Mühendislik literatüründe görülmektedir. Pless ve arkadaşlarının [20] de yaptıkları çalışmadan uzun bir zaman sonra Huffman [10] da F_2, F_3, F_4 cisimleri ve $Z_4, F_2 + uF_2, F_2 + vF_2$ halkaları üzerinde tanımlı bütün self-dual kodların sınıflandırılmalarını içeren kapsamlı bir çalışma yapmıştır. Hata Düzelten Kodlar Teorisi oldukça geniş bir alandır ve bu çalışmada bu disiplinin oldukça küçük ama etkili bir kısmı incelenmiştir.

Bir kodda genel olarak istenen özellikler kısa yani hızlı transfer edilebilecek olması, çok sayıda mesaj göndermeye elverişli yani çok sayıda kod kelimesine sahip olması ve aynı zamanda çok sayıda hata düzeltebilmesi yani kod kelimelerinin mümkün olduğunca birbirinden farklı olmasıdır. Buradan açıkça görülebilir ki istenen bu özellikler birbirleriyle çelişecek biçimdedir. Dolayısıyla bu parametrelerin bir ya da daha fazlasının üzerinde kısıtlamalar koyarak bir ya da tüm "en iyi" kodlar diğer parametrelere bağlı biçimde bulunmaya çalışılabilir. Kodlama teorisinde araştırmacılar genellikle özel işlevlere sahip en iyi kodları bulmakla ilgilenmişler, bazen sadece bir "en iyi" kodu bazen de çeşitli ortak özelliklere sahip bütün "en iyi" kodları bulmak istemişlerdir. Örneğin, bu çalışmanın temel başlığını oluşturan self-dual kodlar için, Harada[42] de 36 uzunluklu bütün self-dual kodları sınıflandırmış, [41] de 40 uzunluklu katlı-çift self-dual kodların sınıflandırmasını tamamlamıştır. Pless ve Lam [43] de [24,12,10] tipinde bir quaternary self-dual kodun var olamayacağını ispatlamış, [72,36,16] tipinde bir binary kodun var olup olmadığı problemi ise birçok kodlama teorisyeni tarafından yoğun şekilde araştırılmasına rağmen henüz cevaplanamamıştır. Ayrıca kodlama teorisyenleri tarafından özel olarak self-dual

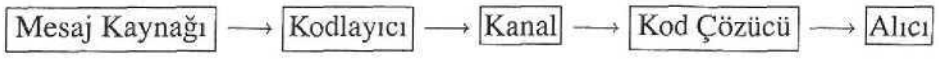
kod üreten metodlar elde edilmeye çalışılmıştır. Harada [44] de ortogonal dizaynlar kullanarak, [12] de ise döngüsel yapılar kullanarak self-dual kod inşa etmeye çalışmış, Kim [8] de var olan bir self-dual koddan daha uzun yeni bir self-dual kod üreten üst-yapı inşa metodunu tanıtmış, [7,9,14,25] de bu metodu farklı yapılara taşımıştır. [11] de ise Melchor, binary self-dual kodlar için belirli uzunlukluktaki self-dual kodları kullanarak daha uzun bütün self-dual kodları üreten bir algoritma geliştirmiştir. Bunların dışında da literatürde özel olarak self-dual kodlar üreten çeşitli metodlar bulunmaktadır.

Bu tez çalışmasında ilk bölüm giriş kısmına ayrılmış ve kodlama teorisi ve self-dual kodlar hakkında genel bilgi verilmiştir. İkinci bölümde, çalışmanın devamında sıklıkla kullanılacak olan kodlama teorisinin temel kavramlarından bahsedilmiştir. Bu çalışmada incelenen self-dual kodlar ailesi lineer kodların bir sınıfı olduğundan üçüncü bölüm lineer kodlara ayrılmıştır. Bu bölümde, ilk önce lineer kodlar ve özelliklerine yer verilmiştir. Daha sonra lineer kodlarda ağırlık ve uzunluk üzerinde durulmuş ve bu kavramlarla ilgili bazı tanım ve teoremlere yer verilmiştir. Ayrıca lineer kodlar için önemli bir örnek teşkil eden Hamming kod sınıfına değinilmiştir. Self-dual kodların bulunabilmesinin yanısıra sınıflandırılması da oldukça önemlidir. Bu yüzden dördüncü bölüm self-dual kod sınıflandırmasında önemli rol oynayan kodların denkliğine ayrılmıştır. Beşinci bölümde, self-dual kodlar incelenmiştir. Bu bölümde önce, self-dual kodlar için GPW teoremi ve Gleason polinomları verilmiştir. Sonra, self-dual kodların sayılması ve sınıflandırılması için metod ve formüller verilmiştir. Altıncı bölümde, ilk olarak genel kod inşa metodlarından bazıları verilmiştir. Sonra da, self-dual kod üreten bazı özel inşa metodlarına yer verilmiş ve yinelemeli algoritma ile üst-yapı inşa metodu kıyaslanarak bu metodlardan birinin aslında diğerinin özel bir hali olduğu ispatlanmıştır. Yedinci bölüm, bir önceki bölümde bahsedilen metodlarla üretilmiş kodlara ait tablo ve üreteç matrislerinden oluşmaktadır. Sekizinci ve son bölüm sonuç ve öneriler kısmına ayrılmıştır. Bu çalışmada yapılan kıyaslama ile elde edilebilecek bulgulardan bahsedilmiştir. Çalışma boyunca soyut cebir ve lineer cebir ile ilgili birçok kavram sıklıkla kullanılmıştır ve bu çalışmada yer alan bu kavramlarla ilgili tanım ve teoremler [45,46] kaynaklarında mevcuttur.

2. BÖLÜM

KODLAMA TEORİSİNDE TEMEL KAVRAMLAR

Bir mesajın, üzerinde hiçbir değişiklik yapılmadan bir iletişim kanalı yoluyla gönderilmesi halinde, transfer esnasında mesajda bozulma meydana gelebilir ve bunu düzeltmek mümkün olmayabilir. Dolayısıyla, bu bozulmaya karşı mesajın korunabilmesi amacıyla mesaj üzerinde çeşitli değişiklikler ya da eklemeler yapılması gerekmektedir. İletişimde kalitenin arttırılması için mesaj üzerinde yapılan bu işleme *kodlama* denir. Gönderilmek istenen her bir verinin kodlandıktan sonraki haline ise *kod kelimesi* denir. Mesajların kodlanması için gerekli olan sembollerin tümünün kümesine *kod alfabesi* denir. Kod alfabesi kullanılarak oluşturulan bütün kod kelimelerinin oluşturduğu kümeye **kod** denir. Örneğin, $C = \{00, 11\}$ kodunun kod kelimeleri sırasıyla 00 ve 11 dir; kod alfabesi ise $\{0, 1\}$ kümesidir. Şekil 2.1 de bir iletişim kanalı, kodlama ve kod çözme safhaları ile birlikte şemalaştırılmıştır.



Şekil 2.1

q adet sembol içeren bir kod alfabesi kullanılarak yazılan kod kelimelerinin kümesi bir *q-luk kod* oluşturur. Örneğin $F_2 = \{0, 1\}$ alfabesi ile yazılan kod kelimelerinin kümesi 2-lik ya da diğer bir deyişle *binary* kod olarak isimlendirilir. Benzer şekilde $F_3 = \{0, 1, 2\}$ alfabesi ile yazılan kod kelimelerinin kümesi 3-lük ya da diğer bir deyişle *ternary* kod; $F_4 = \{0, 1, \omega, \overline{\omega}\}$ alfabesi ile yazılan kod kelimelerinin kümesi 4-lük ya da diğer bir deyişle *quaternary* kod olarak isimlendirilir.

Bir kodda bütün kod kelimeleri aynı sayıda bileşene sahiptir ve bu sayıya *kod uzunluğu* denir. Örneğin 3 uzunluklu binary tekrarlama kodu $C = \{000, 111\}$ şeklindedir. Bir kod kelimesindeki sıfırdan farklı bileşen sayısına *Hamming ağırlık* denir ve x bir kod kelimesini temsil etmek üzere, $wt(x)$ ile gösterilir. Örneğin $wt(111) = 3$ ve $wt(000) = 0$ dır. Kod-

lama teorisi literatüründe *Lee ağırlık*, *Euclidean ağırlık* gibi çeşitli ağırlık tanımları yer almaktadır. Fakat bu çalışmada Hamming ağırlık esas alınmış ve çalışmanın devamında kısaca *ağırlık* olarak nitelendirilmiştir.

İki kod kelimesinde birbirinden farklı sembollerin bulunduğu bileşenlerin sayısı kelimelerin arasındaki *Hamming uzaklık* olarak isimlendirilir ve x ile y birer kod kelimesini temsil etmek üzere bu iki kelime arasındaki Hamming uzaklık $d(x,y)$ ile gösterilmektedir. Örneğin, $d(000,111) = 3$ tür. F_q sonlu cisim olmak üzere, Hamming uzaklık fonksiyonu F_q^n vektör uzayı üzerinde aşağıdaki dört özelliği sağlar;

(i) $\forall\, x, y \in F_q^n$ için $d(x,y) \geq 0$,

(ii) $d(x,y) = 0 \Leftrightarrow x = y$,

(iii) $\forall\, x, y \in F_q^n$ için $d(x,y) = d(y,x)$,

(iv) $\forall\, x, y, z \in F_q^n$ için $d(x,z) \leq d(x,y) + d(y,z)$

Böylece Hamming uzaklık F_q^n uzayı üzerinde bir metrik olur. Ağırlık tanımına benzer şekilde, kodlama teorisi literatüründe uzaklık için de Hamming uzaklıktan farklı çeşitli uzaklık tanımları yer almaktadır. Fakat bu çalışmada Hamming uzaklık esas alınmış ve çalışmanın devamında kısaca *uzaklık* olarak nitelendirilmiştir.

Kod kümesinde sıfırdan farklı en küçük kelime ağırlığı o kodun *minimum ağırlığı*, birbirinden farklı kelimeler arasındaki en küçük uzaklık ise o kodun *minimum uzaklığı*'dır. Örneğin, $C = \{000, 111, 110, 001\}$ binary kodu için $wt(001) = 1$ olduğundan minimum ağırlık 1; $d(110,111) = d(001,000) = 1$ olduğundan minimum uzaklık 1 dir. Genel olarak q adet sembole sahip bir alfabe üzerinde, n uzunluklu, M adet kelimeye sahip ve minimum uzaklığı d olan bir C kodu, **q-luk (n,M,d)-kod** biçiminde gösterilir. Buradan da anlaşılabileceği gibi C kodu, q adet sembolle yazılan n-lilerin kümesinin bir alt kümesidir.

Bu bölümde yer alan bütün kavramlar [1,2,3,4,5] kaynaklarından alınmıştır.

3. BÖLÜM

LİNEER KODLAR

Cebirsel kodlama teorisinde, lineer kodlar üzerinde oldukça yoğun bir şekilde çalışılmaktadır. Bunun nedeni ise sahip oldukları cebirsel yapıdan dolayı kodlama ve kod çözme açısından lineer olmayan kodlara göre çok daha avantajlı olmalarıdır. Örneğin, bir lineer kod bu bölümde tanımları verilecek olan üreteç matrisi veya eşlik-kontrol matrisi ile belirlidir. Bu durum, bütün kod kümesinin sadece tek bir matris ile ifade edilebilmesi kolaylığını sağlar. Lineer bir kodun, kod alfabesi sonlu bir cisimdir ve sonlu bir cisim, bir asalın kuvveti olan q eleman sayısını temsil etmek üzere, $GF(q)$ ya da F_q ile gösterilir.

Tanım 3.0.1. F_q Galois cismi üzerindeki bütün n-lilerin oluşturduğu vektör uzay F_q^n ile gösterilmek üzere; F_q^n uzayının k boyutlu bir alt vektör uzayına n uzunluklu, k boyutlu bir *Lineer Kod* denir ve $[n,k]_q$-*kod* ya da q-*luk* $[n,k]$ *kod* ile gösterilir. Eğer bu kodun minimum uzaklığı d biliniyorsa, $[n,k,d]_q$-*kod* ya da q-*luk* $[n,k,d]$ *kod* biçiminde gösterilir [2].

Kodlama Teorisi literatürüne bakıldığında F_q^n vektör uzayından alınan herhangi bir $(a_1, a_2, a_3, ..., a_n)$ vektörü genellikle $a_1...a_n$ formunda yazılmaktadır. Dolayısıyla lineer bir koddaki kod kelimeleri de bu formdadır. F_q cismi üzerindeki k boyutlu bir lineer kodda q^k adet kod kelimesi vardır.

Tanım 3.0.2. C bir $[n,k]_q$ lineer kod olmak üzere, satırları C nin taban vektörleri olan kxn tipinde bir G matrisine C kodunun *üreteç matrisi* denir [1].

C kodundaki bütün kod kelimeleri bu matrisin satırlarının birer lineer kombinasyonu olacağından C kodu kısaca bu matris yoluyla temsil edilebilir. Ayrıca bir vektör uzayın alt uzayının birden fazla tabanı olabileceğinden, bir lineer kodun birden fazla üreteç matrisinin bulunabileceğini söylemek mümkündür.

Tanım 3.0.3. kxn tipinde bir G üreteç matrisinin herhangi k adet lineer bağımsız sütunu C kodunun *bilgi pozisyonlarının kümesini* oluşturur. Kalan $r = n - k$ adet sütun ise *yineleme pozisyonlarının kümesini* oluşturur ve r, C kodunun *yinelemesi* olarak adlandırılır [4].

Tanım 3.0.4. I_k, kxk tipinde birim matrisi temsil etmek üzere, eğer C kodunun bir üreteç matrisinde ilk k adet sütun bilgi kümesini oluşturuyorsa bu kodun $[I_k \mid A]$ biçiminde bir üreteç matrisi vardır. Böyle bir üreteç matris için *standart formda*'dır denir [2].

Tanım 3.0.5. C bir $[n,k]_q$ lineer kod ise, C kodunun $C = \{x \in F_q^n \mid Hx^T = 0\}$ ile tanımlanan $(n-k)$xn tipinde bir H *eşlik-kontrol matrisi (parity-check matrix)* vardır [2].

Üreteç matriste olduğu gibi, bir kodun birden fazla eşlik-kontrol matrisi bulunabilir. Ayrıca eşlik-kontrol matrisinin satırları lineer bağımsızdır.

3.1 Lineer Kodlarda Ağırlık ve Uzunluk

Teorem 3.1.1. *C bir lineer kod ve minimum ağırlığı d olsun.*

(i) Eğer $d \geq s+1$ ise C kodu s adet hataya kadar tespit eder.

(ii) Eğer $d \geq 2t+1$ ise C kodu t adet hataya kadar düzeltir [2].

İspat . (i) $d \geq s+1$ olsun. Bir x kod kelimesi gönderilsin ve alınan vektör s ya da daha az hata içersin. Bu durumda bu vektör C koduna ait herhangi bir kod kelimesi olamaz. Dolayısıyla alınan vektörün hatalı olduğu tespit edilir.

(ii) $d \geq 2t+1$ olsun. Bir x kod kelimesi gönderilsin ve alınan vektör t ya da daha az hata içeren y vektörü olsun. Buradan $d(x,y) \leq t$ olur. Eğer $x' \neq x$ ve $x' \in C$ ise $d(x',y) \geq t+1$ dir. Aksi takdirde $d(x',y) \leq t$ olduğundan ve üçgen eşitsizliğinden
$d(x,x') \leq d(x,y) + d(x',y) \leq 2t$ olur. Bu da $d \geq 2t+1$ kabulü ile çelişir. Böylece y vektörüne en yakın kod kelimesi x olur ve alınan vektör x olarak çözülür. Dolayısıyla hata düzeltilmiş olur.

Sonuç 3.1.2. *Eğer bir C kodunun minimum uzaklığı d ise, bu kod*

(i) $d-1$ adet hataya kadar tespit eder.

(ii) $\lfloor \frac{d-1}{2} \rfloor$ adet hataya kadar düzeltir [2].

Lemma 3.1.3. *Eğer $x, y \in F_q^n$ ise $d(x,y) = wt(x-y)$ dir [2].*

İspat . $x = x_1 \dots x_n$, $y = y_1 \dots y_n$ ve $d(x,y) = n - i$ olsun. Bu durumda x vektörünün i adet koordinatı ile y vektörünün i adet koordinatı ortaktır. Dolayısıyla $x - y$ vektöründe ortak olan koordinatlar sıfır ve diğer bütün koordinatlar sıfırdan farklı olacaktır. Bu da $wt(x-y) = n-i$ olduğunu gösterir. Sonuç olarak $d(x,y) = wt(x-y)$ eşitliği elde edilir.

Teorem 3.1.4. *Lineer bir kodda minimum ağırlık ve minimum uzaklık birbirine eşittir [2].*

İspat . Bir C lineer kodunun minimum ağırlığı $wt(C)$ ve minimum uzaklığı $d(C)$ ile gösterilsin. Buna göre $\exists x, y \in C$ için $d(C) = d(x,y)$ dir. Lemma 3.1.3 den $d(C) = wt(x-y)$ ve aynı zamanda $x-y \in C$ olduğundan $wt(x-y) \geq wt(C)$ elde edilir. Yani $d(C) \geq wt(C)$ olur. Diğer yandan $\exists x \in C$ için $wt(C) = wt(x) = d(x,0) \geq d(C)$ elde edilir. Dolayısıyla $d(C) \geq wt(C)$ ve $wt(C) \geq d(C)$ olduğundan $d(C) = wt(C)$ dir.

Tanım 3.1.5. F_q^n uzayına ait $x = x_1 \dots x_n$ ve $y = y_1 \dots y_n$ vektörlerinin *standart iç çarpımı* $x \cdot y = \sum_{i=1}^{n} x_i y_i$ ile tanımlıdır. *Hermitian iç çarpımı* ise q, p gibi bir asal sayının çift kuvveti olmak üzere, $x \cdot \overline{y} = \sum_{i=1}^{n} x_i y_i^{\sqrt{p}}$ ile tanımlıdır [4].

Teorem 3.1.6. *$x, y \in F_2^n$ ise aşağıdaki eşitlikler sağlanır.*

(i) $wt(x+y) = wt(x) + wt(y) - 2wt(x \cap y)$

(ii) $wt(x \cap y) \equiv x \cdot y (mod 2)$

(iii) $wt(x) \equiv x \cdot x (mod 2)$ *[4].*

İspat . (i) $wt(x \cap y)$, x ve y vektörlerinin karşılıklı 1 sembolünü bulunduran ortak koordinatlarının sayısıdır. $x + y$ toplamında bu ortak koordinatlar sıfırı verecektir. Diğer yandan hem x hem de y vektörlerinin ağırlıkları bulunurken karşılıklı ortak koordinatlardaki 1 sembolleri ayrı ayrı hesaba katılmıştır. Dolayısıyla buradan $wt(x+y) = wt(x) + wt(y) - 2wt(x \cap y)$ sonucu elde edilir.

(ii) $wt(x \cap y) = k$ olsun. Bu durumda $x \cdot y = x_1y_1 + x_2y_2 + \dots + x_ny_n$ toplamında k adet terim 1, geriye kalan $n-k$ adet terim ise sıfır olacaktır. Dolayısıyla $x \cdot y = \sum_{i=1}^{n} x_i y_i \equiv k (mod 2)$ olur. Bu da $wt(x \cap y) \equiv x \cdot y (mod 2)$ demektir.

(iii) Bir önceki şıkta $y = x$ seçildiği takdirde istenen sonuç elde edilir.

Tanım 3.1.7. C, bir $[n, k]_q$ tipinde lineer kod olsun. $C^{\perp} = \{x \in F_q^n | \, x \cdot c = 0, \forall c \in C\}$ kümesine C kodunun *duali* denir [1].

Teorem 3.1.8. *Eğer G ve H, bir C kodunun sırasıyla üreteç ve eşlik kontrol matrisleri ise, $C^{\perp}$ kodunun üreteç ve eşlik-kontrol matrisleri sırasıyla H ve G olur [4].*

İspat. Eşlik-kontrol matrisinin ve $C^{\perp}$ kodunun tanımlarından, H matrisinin $C^{\perp}$ kodu için bir üreteç matrisi olduğu görülür. G, C kodunun bir üreteç matrisi olduğundan, $\forall x \in C$ için $G \cdot x^T = 0$ elde edilir. Bu da G matrisinin $C^{\perp}$ kodu için bir eşlik-kontrol matrisi olduğunu gösterir.

Tanım 3.1.9. C bir lineer kod olsun. Eğer $C \subseteq C^{\perp}$ ise C koduna üzerinde tanımlı olan iç çarpımın türüne göre *Öklid Self-Ortogonal* ya da *Hermitian Self-Ortogonal* kod denir. Eğer $C = C^{\perp}$ ise C koduna üzerinde tanımlı olan iç çarpımın türüne göre *Öklid Self-Dual* ya da *Hermitian Self-Dual* kod denir [4].

Bu çalışmanın genelinde Öklid self-dual kodlar esas alınmış ve kısaca *Self-Dual* kod olarak isimlendirilmiştir.

Tanım 3.1.10. Eğer bir C lineer kodunda bütün kod kelimelerinin ağırlıkları $\Delta > 1$ sayısına bölünüyorsa bu koda *bölünebilen kod* denir. Δ bu koşulu sağlayan en büyük tamsayı ise C kodunun *böleni* olarak adlandırılır [4].

Tanım 3.1.11. 4 ile bölünebilen binary kodlar *katlı-çift (doubly-even)* kod olarak isimlendirilir. Aksi takdirde, bir binary koda *tekli-çift(singly-even)* kod denir [4].

Tanım 3.1.12. Bir C lineer kodunda ağırlığı i olan kod kelimelerinin sayısı $A_i(C)$ ya da kısaca A_i ile gösterilsin. $0 \leq i \leq n$ için A_i lerin listesine C kodunun *ağırlık dağılımı* ya da *ağırlık spektrumu* denir [4].

Tanım 3.1.13. n uzunluklu bir C lineer kodunda i ağırlıklı kod kelimelerinin sayısı A_i ile gösterilsin. Bu durumda,

$$W_c(x, y) = A_0 x^n + A_1 x^{n-1} y + A_2 x^{n-2} y^2 + \ldots + A_n y^n$$

polinomuna C kodunun *ağırlık sayaç polinomu* denir [4].

Örnek 3.1.14. Bir üreteç matrisi $G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$ olan binary C kodunun

ağırlık dağılımı $A_0 = A_6 = 1$ ve $A_2 = A_4 = 3$ olur.

Teorem 3.1.15. *C, F_q üzerinde tanımlı bir $[n, k, d]$ kod olsun. Bu durumda,*

(i) $A_0(C) + A_1(C) + ... + A_n(C) = q^k$

(ii) $A_0(C) = 1$ *ve* $A_1(C) = A_2(C) = ... = A_{d-1}(C) = 0$

(iii) Eğer bir C binary kodu $1 = 11...1$ *vektörünü içeriyorsa* $0 \leq i \leq n$ *için* $A_i(C) = A_{n-i}(C)$ *dir.*

(iv) Eğer C bir binary self-ortogonal kod ise bütün kod kelimeleri çift ağırlıklı olur. Ayrıca $C^{\perp}$ kodu $1 = 11...1$ *vektörünü içerir [4].*

İspat . (i) $|C| = q^k$ olduğundan, $A_0(C) + A_1(C) + ... + A_n(C) = q^k$ yani C deki bütün kod kelimelerinin sayısıdır.

(ii) C kodu lineer bir kod olduğundan 0 vektörünü içermelidir. Dolayısıyla $A_0(C) = 1$ dir. Diğer yandan, C kodunun minimum ağırlığı d olduğundan C de ağırlığı d den daha küçük olan kod kelimesi yoktur. Böylece $A_1(C) = A_2(C) = ... = A_{d-1}(C) = 0$ olur.

(iii) C, $1 = 1...1$ vektörünü içeren bir binary kod ve C de i ağırlıklı bütün kod kelimelerinin sayısı $A_i(C)$ olsun. i ağırlıklı bütün bu kod kelimeleri ile 1 vektörü toplandığı takdirde C deki $n - i$ ağırlıklı bütün kod kelimeleri elde edilir. Dolayısıyla $A_i(C) = A_{n-i}(C)$ olur.

(iv) C binary bir self-ortogonal kod ise $\forall x \in C$ için $x \cdot x \equiv 0(mod 2)$ dir. Teorem 3.1.6 ya göre $wt(x) \equiv x \cdot x(mod 2)$ olduğundan C kodunun bütün kod kelimeleri çift ağırlıklı olur. Dolayısıyla $\forall x \in C$ için $x \cdot 1 = 0$ olur. Bu da $1 \in C^{\perp}$ demektir.

Teorem 3.1.16. *C self-ortogonal bir binary kod olsun. C kodunda ağırlığı 4 ile bölünebilen kod kelimelerinin kümesi C_0 olsun. Buna göre,*

(i) $C = C_0$ *dır, ya da*

(ii) C_0, C *nin* $[n, k-1]$ *tipinde bir alt kodudur ve ağırlığı 4 ile bölünemeyen fakat çift sayı olan herhangi bir* x *vektörü için* $C_1 = x + C_0$ *olmak üzere* $C = C_0 \cup C_1$ *dir. Ayrıca* C_1 , C *kodunda ağırlığı 4 ile bölünemeyen bütün kod kelimelerini içerir [4].*

İspat . Teorem 3.1.15 in (iv) şıkkından C kodundaki bütün kod kelimelerinin çift ağırlıklı olduğu söylenir. Buna göre, bu teoremin ya (i) şıkkı doğrudur ya da ağırlığı çift fakat dördün katı olmayan en az bir kod kelimesi mevcuttur. İkinci durumun doğru olduğunu varsayalım. Ağırlığı çift fakat dördün katı olmayan diğer bir kod kelimesi y olsun. Bu durumda Teorem 3.1.6 nın (i) şıkkından, $wt(x+y) = wt(x)+wt(y)-2wt(x\cap y) \equiv 2+2-2wt(x\cap y)(mod4)$ elde edilir. Aynı teoremin (ii) şıkkına göre $wt(x\cap y) \equiv x\cdot y(mod2)$ idi. C kodu self-ortogonal bir kod olduğundan $wt(x\cap y) \equiv 0(mod2)$ olur. Bu da $wt(x+y)$ nin 4 ile bölünebildiğini gösterir ve $x+y \in C_0$ olur. Dolayısıyla $y \in x+C_0$ ve $C = C_0\cup(x+C_0)$ elde edilir. $C_1 = x + C_0$ biçiminde yazılırsa $C = C_0 \cup C_1$ olur. Burada C_1, ağırlığı çift fakat dördün katı olmayan bütün kod kelimelerinin kümesi ve C nin bir alt kodudur.

Teorem 3.1.17. *C bir binary $[n, k]$ kod olsun. C kodunda ağırlığı çift olan kod kelimelerinin kümesi C_e ile gösterilsin. Buna göre,*

(i) $C = C_e$ *dir, ya da*

(ii) x ağırlığı tek olan herhangi bir kod kelimesi olmak üzere $C_o = x + C_e$ olsun. Buna göre $C = C_e \cup C_o$ dır ve C_e, C kodunun bir $[n, k-1]$ alt kodudur. Bunun yanısıra C kodundaki ağırlığı tek olan bütün kod kelimelerinin kümesi C_o olur [4].

İspat . $[n, k]$ tipinde binary bir C kodunda bütün kod kelimelerinin ağırlığının çift olduğu ya da yarısının çift yarısının tek ağırlıklı olduğu gösterildiği takdirde bu teorem ispatlanmış olur. Öncelikle, C kodundaki çift ağırlıklı kelimelerin kümesi C_e ve tek ağırlıklı kelimelerin kümesi C_o ile gösterilsin. $f : C \to \{0, 1\}$, $f(x) \equiv wt(x)(mod2)$ fonksiyonunun iyi-tanımlı olduğu açıktır. Ayrıca $f(0) = 0$ dır. Böylece,

$$f(x+y) \equiv wt(x+y)(mod2)$$

$$\equiv wt(x) + wt(y) - 2wt(x\cap y)(mod2)$$

$$\equiv wt(x) + wt(y)(mod2)$$

$$= f(x) + f(y)$$

olduğundan f bir grup homomorfizmidir ve *çekf*$= C_e$ dir. Eğer C kodundaki bütün kod kelimeleri çift ağırlıklı ise $C = C_e$ olur. Aksi takdirde, f örten olur ve bu durumda homomorfizma teoreminden $C/C_e \cong \{0,1\}$ elde edilir. Yani $[C : C_e] = 2$ olur. Böylece $\frac{|C|}{|C_e|} = 2$ eşitliğinden $|C_e| = 2^{k-1}$ olduğu görülür. Bu da $dim(C_e) = k-1$ olduğunu gösterir. Sonuç olarak C_e, C nin $k-1$ boyutlu bir alt kodu olur. Diğer yandan, $x \in C - C_e$ ise $x + C_e = C_o$ elde edilir. Dolayısıyla $C = C_e \cup C_o$ ve C_e, C nin $[n, k-1]$ tipinde bir alt kodu olur.

Teorem 3.1.18. *C bir lineer binary kod olsun. Buna göre,*

(i) Self-ortogonal bir C kodunun bir üreteç matrisinin bütün satırlarının ağırlığı 4 ile bölünebiliyorsa, C kodundaki bütün kod kelimelerinin ağırlıkları 4 ile bölünebilir.

(ii) Bir C kodunda bütün kod kelimelerinin ağırlıkları 4 ile bölünebiliyorsa, C self-ortogonal bir kod olur [4].

İspat . (i) Self-ortogonal bir binary C kodu bütün satırlarının ağırlıkları dördün katı olan bir üreteç matrise sahip olsun. Böylece bu matrisin herhangi iki satırı x ve y ile gösterilirse, Teorem 3.1.6(i) den $wt(x+y) = wt(x) + wt(y) - 2wt(x \cap y) \equiv 0 + 0 - 2wt(x \cap y) \equiv 0(mod4)$ elde edilir. Böylece C kodundaki bütün kod kelimeleri bu matrisin satırlarının bir lineer kombinasyonu olduğundan bütün bu kod kelimelerinin ağırlıklarının 4 ile bölünebildiği görülür.

(ii) $x, y \in C$ olsun. Teorem 3.1.6 (ii) den, $2wt(x \cdot y) \equiv 2wt(x \cap y) \equiv 2wt(x \cap y) - wt(x) - wt(y) \equiv -wt(x+y) \equiv 0(mod4)$ elde edilir. Bu da $x \cdot y \equiv 0(mod2)$ demektir. Yani C self-ortogonal olur.

Teorem 3.1.19. *Bir C binary kodunun bütün satırları çift ağırlıklı olan bir üreteç matrisi varsa bu kodun bütün kod kelimeleri çift ağırlıklıdır [4].*

İspat . C binary kodu bütün satırları çift ağırlıklı olan bir üreteç matrise sahip olsun. C kodunun bütün kod kelimeleri bu matrisin satırlarının lineer bir kombinasyonudur. x ve y bu matrisin herhangi iki satırı olmak üzere, Teorem 3.1.6(i) den $wt(x+y) = wt(x) + wt(y) - 2wt(x \cap y) \equiv 0(mod2)$ elde edilir. Dolayısıyla C kodundaki bütün kod kelimeleri çift ağırlıklı olur.

Tanım 3.1.20. F_q^n vektör uzayındaki herhangi bir $x = x_1x_2 \dots x_n$ vektörü $\sum_{i=1}^{n} x_i = 0$ eşitliğini sağlıyorsa bu vektör *çiftimsi (even-like)* olarak isimlendirilir. Çiftimsi olmayan bir vektör *tekimsi(odd-like)* olarak isimlendirilir. Eğer bir kodda bütün kod kelimeleri çiftimsi ise o koda *çiftimsi (even-like) kod* denir, aksi takdirde *tekimsi (odd-like) kod* denir [4].

Teorem 3.1.21. *C, F_q üzerinde tanımlı bir $[n, k]$ kod olsun. C kodundaki çiftimsi kod kelimelerinin kümesi C_e olsun. Buna göre,*

(i) $C = C_e$ dir, ya da

(ii) C_e, C nin bir $[n, k-1]_q$ alt kodudur [4].

İspat . Bu teoremin ispatı Teorem 3.1.17 nin ispatına benzer şekilde yapılır.

3.2 Hamming Kodlar

Hamming kodlar, ilk olarak Claude Shannon ile Bell laboratuarlarında birlikte çalışmış olan Richard Hamming tarafından ortaya konulmuştur [35]. Hamming, kendi bilgisayarı ile yaptığı çalışmalar esnasında meydana gelen hataların düzeltilmesi için bu kodlara ihtiyaç duymuş ve bilinen ilk Hamming kod olan $[7, 4, 3]_2$ Hamming kodunu üretmiştir [35]. Daha sonra bu kodlar Golay tarafından daha uzun binary kodlara ve ayrıca binary olmayan kodlara genelleştirilmiştir [36]. Hamming [7,4,3] kod, kodlama ve kod çözme safhalarının daha pratik bir şekilde yapılabilmesini sağlayan bir kod türüdür. Tek hata düzeltebilen ve iki hata tespit edebilen bu kodlar, geçmişte olduğu gibi günümüzde de önemini korumaktadır. Özellikle veri sıkıştırma amacıyla kullanılan kodlardır [1].

3.2.1 Hamming(7,4) Kod

Tanım 3.2.1. 7 uzunluklu binary Hamming kodu ; $H_{2,3} = \{x \in F_2^n | H_3x^T \equiv 0(mod2)\}$ ile tanımlıdır.

$[7,4,3]_2$ Hamming kodunun üreteç matrisi G_3 ve eşlik-kontrol matrisi H_3 aşağıda verilmiştir.

$$G_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}, H_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Bu kod tek hata düzeltir ve iki hata belirler. H_3 matrisi ile kanaldan alınan y vektörünün transpozunun çarpımıyla (eğer bu matris tek hata içeriyorsa) yine H_3 matrisinin sütunlarından biri elde edilir. Bu sütunlar ikilik sistemde sırasıyla 1,2,...,7 sayılarını temsil eder. Elde ettiğimiz sütun hangi pozisyonda hata olduğunu gösterir ve böylece hata düzeltilebilir.

Örnek 3.2.2. Uygun bir iletişim kanalı yoluyla $x = 1000011 \in H_3$ kod kelimesinin transfer edildiğini ve bu esnada kod kelimesinde bozulma meydana geldiğini varsayalım. Tek hata oluştuğunda alınan vektör $y_1 = 1100011$, iki hata oluştuğunda alınan vektör $y_2 = 0100011$, üç hata oluştuğunda alınan vektör $y_3 = 0110011$ olsun.

Buna göre;

$$H_3 \cdot y_1^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Bu sonuç H_3 matrisinin ikinci sütununu verir. Aynı zamanda ikilik tabanda $(010)_2 = 2$ dir. Her iki şekilde de elde edilen sonuç kod kelimesinin transferi esnasında ikinci pozisyonda hata yapıldığını gösterir. Böylece $y_1 = 1100011$ vektörünün ikinci bileşeni 0 olarak değiştirilir ve oluşan tek hata düzeltilerek orjinal kod kelimesi $x = 1000011$ bulunur.

$$H_3 \cdot y_2^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

Bu sonuç H_3 matrisinin 3.sütununu verir. Aynı zamanda ikilik tabanda $(011)_2 = 3$ tür. Her iki şekilde de elde edilen sonuç kod kelimesinin transferi esnasında üçüncü pozisyonda hata olduğunu gösterir. Dolayısıyla alınan vektörün hatalı olduğu tespit edilmesine rağmen ilk iki pozisyonda yapılmış olan hatalar düzeltilemez.

$$H_3 \cdot y_3^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Bu sonuç $y_3 \in H_3$ olduğunu gösterir. Bu durumda y_3 vektöründeki 3 hata düzeltilemez ve vektörün hatalı olduğu da saptanamaz. Dolayısıyla Hamming $[7, 4, 3]_2$ kodunun kod çözme prosedürü ile tek hata düzeltilebilir, iki hata saptanabilir fakat 3 ya da daha fazla hata oluştuğunda bu durum saptanamaz.

3.2.2 Genelleştirilmiş Hamming Kodlar

$r \geq 2$ için $n = 2^r - 1$ olsun. Bu durumda sütunları $1, 2, ..., 2^r - 1$ olarak sıralanmış r x $(2^r - 1)$ tipindeki H_r matrisi $[n = 2^r - 1, k = n - r]$ binary kodunun eşlik-kontrol matrisini oluşturur. H_r matrisinin sütunlarının yer değiştirmesiyle elde edilen kodlar yine

aynı koda denk olur. Birbirine denk olan bütün bu kodlar $n = 2^r - 1$ uzunluklu binary Hamming kod olarak adlandırılır ve $H_{2,r}$ ile gösterilir. $H_{2,r}$ binary kodunun minimum uzaklığı 3 tür. Dolayısıyla $H_{2,r}$, $[2^r - 1, 2^r - 1 - r, 3]$ binary kod olur [36].

$H_{2,r}$ binary Hamming koduna benzer şekilde herhangi bir F_q cisimi üzerinde $H_{q,r}$ Hamming kodu tanımlanır [36]. $r \geq 2$ için $H_{q,r}$ Hamming kodunun eşlik-kontrol matrisi, F_q^r nin tek boyutlu her bir alt uzayından sıfırdan farklı vektörler alınarak bu vektörlerin sütunlar halinde yazılmasıyla oluşturulur. F_q^rnin $(q^r - 1)/(q - 1)$ adet tek boyutlu alt uzayı vardır. Dolayısıyla $H_{q,r}$ kodu $n = (q^r - 1)/(q - 1)$ uzunluğa sahiptir ve boyutu $n - r$ olur. Burada r yinelemeyi verir. $H_{q,r}$ kodunun da minimum uzaklığı benzer şekilde 3 olur. $q = 2$ durumunda $H_{q,r}$, $H_{2,r}$ yi ifade eder.

4. BÖLÜM

KODLARIN DENKLİĞİ

4.1 Kodların Permütasyon Denkliği

Tanım 4.1.1. C_1 ve C_2 iki lineer kod olsun. C_1 ve C_2 kodları arasında, C_1 in koordinatlarını C_2 ye götüren bir permütasyon varsa bu iki kod *permütasyon denk*'tir denir. Bu permütasyon, her satır ve sütununda yalnızca bir adet 1 bulunan ve diğer bütün bileşenleri sıfır olan kare *permütasyon matrisi* ile temsil edilir [4].

Yukarıdaki tanıma göre, P bir permütasyon matrisi olmak üzere , C_1 ve C_2 kodları permütasyon denk ise, C_1 kodunun bir üreteç matrisinin G_1 olması için gerek ve yeter koşul C_2 kodunun bir üreteç matrisinin G_1P olmasıdır [4]. Bir üreteç matrise P matrisinin uygulanması, üreteç matrisin sütunlarının yer değiştirmesini sağlar. Eğer P, C_1 kodunu C_2 koduna gönderen bir permütasyon ise $C_1P = C_2$ yazılabilir ki burada C_1P aşağıdaki gibidir.

$$C_1P = \{y | y = xP, x \in C_1\}$$

Teorem 4.1.2. *C_1 ve C_2 kodları permütasyon denk olacak biçimde $C_1P = C_2$ eşitliğini sağlayan bir P permütasyon matrisi var olsun. Buna göre,*

(i) $C_1^{\perp}P = C_2^{\perp}$

(ii) C_1 self-dual ise C_2 self-dualdir [3].

İspat . (i) C_1 kodunun üreteç ve eşlik-kontrol matrisleri sırasıyla G ve H olsun. Bu durumda C_2 kodunun üreteç ve eşlik-kontrol matrisleri sırasıyla GP ve HP olur. Diğer yandan, $C_1^{\perp}$ kodunun üreteç ve eşlik-kontrol matrisleri sırasıyla H ve G dir. Dolayısıyla $C_1^{\perp}P$ kodunun üreteç ve eşlik-kontrol matrisleri sırasıyla HP ve GP olur ve böylece istenen sonuç elde edilir.

(ii) $C_1 = C_1^{\perp}$ ise $C_2^{\perp} = C_1^{\perp}P = C_1P = C_2$ elde edilir. Bu da C_2 kodunun self-dual olduğunu gösterir.

Örnek 4.1.3. C_1, C_2, C_3 üreteç matrisleri sırasıyla G_1, G_2, G_3 olan binary kodlar olsun.

$$G_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, G_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}, G_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Bu üç kodun da ağırlık dağılımları $A_0 = A_6 = 1$ ve $A_2 = A_4 = 3$ tür. G_1 matrisinde 2. ve 6. sütunların yeri değiştirilirse G_2 matrisi elde edilir. Bu permütasyon G_1 matrisini G_2 matrisine gönderdiğinden C_1 ve C_2 kodları permütasyon denk olur. C_1 ve C_2 kodları self-dualdir fakat C_3 kodu self-dual değildir. Dolayısıyla C_3 kodu C_1 ya da C_2 ile permütasyon denk değildir.

Teorem 4.1.4. *C bir lineer kod olsun.*

(i) C kodu, üreteç matrisi standart formda olan bir koda denktir.

(ii) Eğer C kodunun bilgi ve yineleme pozisyonlarının kümeleri sırasıyla I ve R ise $C^{\perp}$ kodunun bilgi ve yineleme pozisyonlarının kümeleri sırasıyla R ve I olur [4].

İspat . (i) C kodunun bir üreteç matrisine elementer satır işlemleri uygulandığında bu matris I_k birim matrisinin sütunlarını içeren bir matrise indirgenmiş olur. İndirgeme sonucunda elde edilen matrisin sütunlarına bir permütasyon uygulanarak $[I_k|A]$ formunda yeni bir üreteç matris elde edilir. Dolayısıyla, bu matris tarafından üretilen kod C koduna denk olur.

(ii) C kodunun bir bilgi kümesi I ile gösterilsin. Buna göre, C kodunun bir üreteç matrisine satır indirgemesi yapıldığında bilgi pozisyonlarının bulunduğu sütunlarda, I_k matrisinin sütunları elde edilir. Sonuçta elde edilen matrise bir P permütasyon matrisi uygulanarak $[I_k|A]$ formunda yeni bir üreteç matris bulunur. Burada I kümesi ilk k adet pozisyona kaydırılmış olur. Böylece, $(CP)^{\perp}$ kodunun son $n-k$ adet koordinatının, bu kodun bilgi pozisyonlarını oluşturduğu söylenir. Teorem 4.1.2 den $(CP)^{\perp} = C^{\perp}P$ dir ve böylece R, $C^{\perp}$ kodunun bilgi pozisyonlarının kümesi olur. □

Denklik kavramı açıklanırken döngüsel formdaki permütasyonlar, permütasyon matrislerinden daha sık kullanılır. n adet koordinat içeren bir küme üzerinde tanımlı bütün permütasyonların kümesi Sym_n ile gösterilsin. Eğer $\sigma \in Sym_n$ ve
$x = x_1x_2 \dots x_n$ ise,

$$x\sigma = y_1y_2 \dots y_n, \text{ burada } y_j = x_{j\sigma^{-1}} \text{ ve } 1 \leq j \leq n$$

biçiminde tanımlanır. Böylece $P = [p_{i,j}]$ permütasyon matrisi

$$p_{i,j} = \begin{cases} 1 \text{ , eğer j=i}\sigma \\ 0 \text{ , diğer durumlarda} \end{cases}$$

ile tanımlandığında $x\sigma = xP$ olur.

Örnek 4.1.5. $n = 3, x = x_1x_2x_3 = (x_1, x_2, x_3)$ ve $\sigma = (1, 2, 3)$ olsun. O zaman $1\sigma^{-1} = 3$, $2\sigma^{-1} = 1$, ve $3\sigma^{-1} = 2$ olur. Böylece $x\sigma = x_3x_1x_2$ elde edilir. Ayrıca $P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ matrisi için $xP = x_3x_1x_2$ olur.

Bir C kodunu kendisine götüren bütün koordinat permütasyonları bir grup oluşturur. Bu gruba C kodunun *permütasyon otomorfizm grubu* denir ve $PAut(C)$ ile gösterilir. Eğer C kodu n uzunluklu ise $Paut(C)$, Sym_n simetri grubunun bir alt grubu olur.

Teorem 4.1.6. *C, C_1 ve C_2, F_q üzerinde tanımlı birer kod olsunlar.*

(i) $PAut(C) = PAut(C^{\perp})$

(ii) P bir permütasyon matrisi olmak üzere, eğer $C_1P = C_2$ *ise* $P^{-1}[PAut(C_1)]P = PAut(C_2)$ *dir [4].*

İspat . (i) $\forall P \in PAut(C)$ için $CP = C$ dir. Teorem 4.1.2 den $C^{\perp}P = C^{\perp}$ olur ve böylece $P \in PAut(C^{\perp})$ elde edilir. Bu da $PAut(C) \subseteq PAut(C^{\perp})$ olduğunu gösterir. Benzer şekilde $\forall P' \in PAut(C^{\perp})$ için $C^{\perp}P' = C^{\perp}$ dir. Teorem 4.1.2 den $CP' = C$ olur. Buradan $P' \in PAut(C)$ elde edilir. Bu da $PAut(C^{\perp}) \subseteq PAut(C)$ olduğunu gösterir. Sonuç olarak $PAut(C) = PAut(C^{\perp})$ dir.

(ii) Bir P permütasyon matrisi için $C_1P = C_2$ olsun. $P' \in PAut(C_1)$ ise $C_1P' = C_1$ dir. $P^{-1}(P')P = P''$ ile gösterilirse, $C_2P'' = C_2(P^{-1}P'P) = C_1(P'P) = C_1P = C_2$ elde edilir. Bu da $P'' = P^{-1}P'P \in PAut(C_2)$ olduğunu gösterir. Yani $P^{-1}[PAut(C_1)]P \subseteq PAut(C_2)$ olur. Ters kapsama da benzer yolla gösterilir. Sonuç olarak, $P^{-1}[PAut(C_1)]P = PAut(C_2)$ dir.

4.2 Kodların Monomial Denkliği

F_2 dışındaki cisimler düşünüldüğünde, denklik daha genel bir form alır. Bu cisimler üzerinde kod kelimelerinin ağırlıklarını koruyan farklı dönüşümler vardır. Bunun için öncelikle monomial matris tanımına bakmak gerekir.

Tanım 4.2.1. Her satır ve sütununda sıfırdan farklı yalnız bir eleman bulunan kare matrise *monomial matris* denir. D ve D_1 birer diagonal matris ve P bir permütasyon matrisi olmak üzere, bir M monomial matrisi DP ya da PD_1 formunda yazılabilir [1].

Örnek 4.2.2. $a \neq 0, b \neq 0, c \neq 0$ olmak üzere, $M = \begin{bmatrix} 0 & a & 0 \\ 0 & 0 & b \\ c & 0 & 0 \end{bmatrix}$ monomial matrisi;

$$DP = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} = PD_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} c & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & b \end{bmatrix}$$

matrislerine denktir.

Monomial bir matris genellikle $M = DP$ ile gösterilir, burada D diagonal kısmı ve P permütasyon kısmı oluşturur. Örnek 4.2.2 da, M monomial matrisi $M = diag(a, b, c)(1, 2, 3)$ şeklinde yazılabilir. Buna göre, $diag(a, b, c)$ diagonal kısım ve $(1, 2, 3)$ permütasyon kısımdır.

Örnek 4.2.3. $M = diag(a, b, c)(1, 2, 3)$ monomial matrisi $x = x_1x_2x_3 = (x_1, x_2, x_3)$ vektörüne uygulanırsa $xM = xDP = (ax_1, bx_2, cx_3)P = (cx_3, ax_1, bx_2)$ olur.

Tanım 4.2.4. C_1 ve C_2, F_q üzerinde tanımlı aynı uzunluğa sahip iki kod olsun. C_1 kodunun bir üreteç matrisi G_1 ve M bir monomial matris olmak üzere, C_1 ve C_2 kodlarının

monomial denk olması için G_1M matrisinin C_2 kodunun bir üreteç matrisi olması gerekir. Daha kısa bir ifadeyle, $C_2 = C_1M$ olacak biçimde bir M monomial dönüşümü varsa C_1 ve C_2 kodları monomial denk olur [4].

Binary kodlar için permütasyon denk olma durumu ve monomial denk olma durumu birbirine denk olur [4].

4.3 Kodların Genel Anlamda Denkliği

Kodların genel anlamda denkliğinin tanımlanabilmesi için öncelikle bir vektöre, bir monomial dönüşüm ve bir cisim otomorfizmasının aynı anda nasıl uygulandığının bilinmesi gerekir. Eğer γ, F_q üzerinde bir cisim otomorfizması ve $M = DP$ bileşenleri F_q cisimine ait bir monomial dönüşüm ise herhangi bir x vektörüne $M\gamma$ dönüşümü aşağıdaki adımlarla uygulanır;

(i) x vektörünün i. bileşeni, D diagonal matrisinin i. diagonal bileşeni ile çarpılır.

(ii) Bu çarpım $i\sigma$ koordinatına kaydırılır.

(iii) Elde edilen bu bileşene γ otomorfizmi uygulanır.

Örnek 4.3.1. F_4 cisimi üzerinde γ otomorfizmi $x\gamma = x^2$ ile tanımlansın. Eğer bir M monomial dönüşümü $M = DP = diag(a,b,c)(1,2,3)$ ile tanımlı ve $x = x_1x_2x_3 = (x_1,x_2,x_3) \in F_4^3$ ise,

$$xM\gamma = (ax_1, bx_2, cx_3)P\gamma = (cx_3, ax_1, bx_2)\gamma = ((cx_3)^2, (ax_1)^2, (bx_2)^2)$$

bulunur. $(1,\omega,0)$ vektörü için; $(1,\omega,0)diag(\omega,\overline{\omega},1)(1,2,3)\gamma = (0,\overline{\omega},1)$ elde edilir.

Tanım 4.3.2. C_1 ve C_2, F_q üzerinde tanımlı aynı uzunlukta iki kod olsun. M bir monomial matris ve γ, F_q üzerinde bir cisim otomorfizmi olmak üzere $C_2 = C_1M\gamma$ ise C_1 ve C_2'ye *genel anlamda denk* kodlar denir [4] .

İki kodun aynı olduğunu söylemek için permütasyon denklik, monomial denklik ve genel anlamda denklik olmak üzere üç çeşit denklik vardır. Karşılaştırılan kodlar binary kodlar

olduğunda bu üç durum birbirine denk olur. Kod alfabesi olarak kullanılan cisimin eleman sayısının asal olması durumunda ise monomial denklik ve genel anlamda denklik aynı olur [1].

Birbirine denk iki kod aynı ağırlık dağılımına sahiptir. Fakat aynı ağırlık dağılımına sahip iki kod her zaman birbirine denk olmayabilir. Teorem 4.1.2 ye göre eğer C_1 ve C_2 birbirine permütasyon denk ise dualleri de aynı dönüşümle birbirine permütasyon denktir. Monomial denklikte bu durum her zaman doğru olmayabilir. Yani $C_1M = C_2$ ise $C_1^{\perp}M = C_2^{\perp}$ eşitliği her zaman sağlanmayabilir.

Örnek 4.3.3. C_1 ve C_2, F_4 üzerinde tanımlı [2,1,2] kodlar olsun. [1 1] ve [1 ω] sırasıyla C_1 ve C_2 kodlarının birer üreteç matrisi ise standart iç çarpıma göre $C_1^{\perp}$ ve $C_2^{\perp}$ kodlarının üreteç matrisleri sırasıyla [1 1] ve [1 $\overline{\omega}$] olur. Fakat $C_1 diag(1,\omega) = C_2$ olmasına rağmen $C_1^{\perp} diag(1,\omega) \neq C_2^{\perp}$ dir.

Bu örnekte C_1 self-dual olmasına rağmen C_2 self-dual değildir. Dolayısıyla monomial denklikte self-dual olma özelliği korunamayabilir.

Teorem 4.3.4. *C, F_q üzerinde tanımlı bir kod ve M, bileşenleri $\{0,-1,1\}$ kümesine ait olan monomial bir matris olsun. Bu durumda, C nin self-dual olması için gerek ve yeter koşul CM nin self-dual olmasıdır [4].*

İspat . $a_1, \ldots a_n \in \{1,-1\}$ için $M = diag(a_1, a_2, \ldots, a_n)$ olsun. Öncelikle $(CM)^{\perp} = C^{\perp}M$ olduğunu gösterelim. $(CM)^{\perp}$ ve $C^{\perp}M$ kodlarının tanımından,

$x \in (CM)^{\perp} \Rightarrow \forall c_1 \in CM$ için $x \cdot c_1 = 0$ dır. $c_1 \in CM$ ise $c \in C$ için, $c_1 = cM$ dir.

$$\Rightarrow x \cdot c_1 = x \cdot (cM) = 0$$

$$\Rightarrow x \cdot (c \cdot diag(a_1, a_2, \ldots, a_n)) = 0$$

$$\Rightarrow x \cdot (a_1c_1, a_2c_2, \ldots, a_nc_n) = 0$$

$$\Rightarrow a_1x_1c_1 + a_2x_2c_2 + \ldots + a_nx_nc_n = 0$$

$$\Rightarrow (a_1x_1, a_2x_2, \ldots, a_nx_n) \cdot c = 0$$

$$\Rightarrow (x \cdot diag(a_1, \ldots, a_n)) \cdot c = 0$$

$$\Rightarrow (xM) \cdot c = 0$$

$$\Rightarrow xM \in C^{\perp}$$

$$\Rightarrow (xM)M \in C^{\perp}M$$

Buradan, $xM^2 = aM$ olacak şekilde bir $a \in C^{\perp}$ olduğu söylenir. $M^3 = M$ olduğundan, $xM^3 = aM^2$ yani $xM = aM^2$ dir ve $x = aM$ elde edilir. Bu da, $x \in C^{\perp}M$ olduğunu gösterir. Böylece $(CM)^{\perp} \subseteq C^{\perp}M$ olur. $x \in C^{\perp}M$ ise $x = bM$ olacak şekilde $b \in C^{\perp}$ vardır. $\forall c \in C$ için, $x \cdot cM = (bM) \cdot (cM) = (a_1b_1, \ldots, a_nb_n) \cdot (a_1c_1, \ldots, a_nc_n) = a_1^2b_1c_1 + \ldots + a_n^2b_nc_n = (b \cdot c) \cdot M^2 = 0$ dır. Dolayısıyla $x \in (CM)^{\perp}$ yani $C^{\perp}M \subseteq (CM)^{\perp}$ olur . Buradan, $C^{\perp}M = (CM)^{\perp}$ olduğu söylenir.

C self-dual bir kod olsun. Bu durumda $C = C^{\perp}$ dir. Buradan $CM = C^{\perp}M = (CM)^{\perp}$ olur ve bu CM kodunun self-dual olduğunu gösterir. Tersine, CM kodunun self-dual bir kod olduğunu varsayalım. Bu durumda $(CM)^{\perp} = CM$ yani $C^{\perp}M = CM$ dir. Buna göre, $x \in C^{\perp} \Leftrightarrow xM \in C^{\perp}M \Leftrightarrow xM \in CM \Leftrightarrow x \in C$ elde edilir. Yani C kodu self-dual bir kod olur. Sonuç olarak, C kodunun self-dual olması için gerek ve yeter koşul CM kodunun self-dual olmasıdır. □

Lineer kodlarda denkliğin üç türü olduğundan, kodların otomorfizm grubu da yine üç tiptir. C, F_q üzerinde tanımlı bir kod olsun. C kodunun permütasyon otomorfizm grubu, kodu kendisine resmeden permütasyonların oluşturduğu Sym_n nin bir alt grubu idi ve $PAut(C)$ ile gösterilmişti. Benzer şekilde C kodunun *monomial otomorfizm grubu*, kodu kendisine resmeden monomial matrislerin oluşturduğu gruptur ve $MAut(C)$ ile gösterilir. M bir monomial matris ve γ bir cisim otomorfizmi olmak üzere, C kodunun *otomorfizm grubu*, kodu kendisine resmeden Mγ dönüşümlerinin oluşturduğu gruptur ve $\Gamma Aut(C)$ ile gösterilir. Binary durumda bu üç grup birbirine eşit olur. q asal sayı ise $MAut(C) = \Gamma Aut(C)$ dir. Genel durumda ise $PAut(C) \subseteq MAut(C) \subseteq \Gamma Aut(C)$ dir.

5. BÖLÜM

SELF-DUAL KODLAR

Bilinen en iyi cebirsel kodların birçoğunun self-dual olması bu kod sınıfının önemini büyük ölçüde arttırmaktadır [4]. 8 uzunluklu uzatılmış Hamming kod ve 24 uzunluklu uzatılmış Golay kod bu kod sınıfına ait başlıca örneklerdir [35,36]. Self-dual kodlar, iyi kodlar elde edebilmek için üzerinde çalışmaya değer bir kod sınıfıdır.

Örnek 5.0.1. En kısa binary self-dual kod C_1={00, 11} dir. Üreteç matrisi $G = [11]$ olan [2,1,2] koddur.

Bir sonraki en kısa binary self-dual kod $C_2 = C_1 \oplus C_1$ kodudur. Üreteç matrisi $G = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ olan [4,2,2] koddur.

F_q üzerinde tanımlı n uzunluklu bir C kodu için $dimC + dimC^{\perp} = n$ olduğundan, C self-dual kod ise $dimC = dimC^{\perp} = \frac{n}{2}$ olacaktır. Dolayısıyla bir self-dual kod çift uzunluğa sahiptir ve boyutu uzunluğunun yarısı kadardır.

Tanım 5.0.2. C, F_q üzerinde tanımlı bir lineer kod olsun. $W_C(x, y) = W_{C^{\perp}}(x, y)$ ise C koduna *biçimsel (formally) self-dual kod* denir [4] .

Teorem 5.0.3. *(Gleason-Pierce-Ward Teoremi) C, F_q üzerinde tanımlı bir $[n, \frac{n}{2}]$ kod ve $\Delta > 1$ bu kodun böleni olsun. Bu takdirde aşağıdaki durumlardan biri sağlanır.*

(i) $q = 2$ ve $\Delta = 2$,

(ii) $q = 2$, $\Delta = 4$ ve C self-dual,

(iii) $q = 3$, $\Delta = 3$ ve C self-dual,

(iv) $q = 4$, $\Delta = 2$ ve C Hermitian self-dual,

(v) $\Delta = 2$ *ve* C, F_q *üzerinde tanımlı üreteç matrisi* $[I_{\frac{n}{2}}\ I_{\frac{n}{2}}]$ *olan koda denktir [37,38].*

Gleason-Pierce-Ward teoremi, Gleason ve Pierce tarafından 1967 de yayınlanan Gleason-Pierce teoreminin [39] bir genelleştirilmesidir. Gleason-Pierce teoreminde C nin biçimsel self-dual olması ve (v) şıkkında C nin ağırlık sayacının $(y^2 + (q-1)x^2)^{\frac{n}{2}}$ olması koşulları yer almaktadır. Ward tarafından [37,38] de Gleason-Pierce teoremi genelleştirilmiş ve bu iki koşul kaldırılmıştır. Kısa adıyla GPW teoremi, bir kodu böleni ve self-dual olma özelliği ile ilişkilendirir. GPW teoreminin (i) şıkkında kod sadece böleni ile ilişkilendirilmiştir. Dolayısıyla GPW teoreminin (i) şıkkındaki koşulu sağlayan bir kod self-dual olmayabilir. Bu durum aşağıda örneklendirilmiştir.

Örnek 5.0.4. $G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$ olmak üzere, bir üreteç matrisi G olan [6,3,2] binary kodu, GPW teoreminin (i) şıkkındaki koşulları sağlamasına rağmen self-dual kod değildir.

Tanım 5.0.5. GPW teoremindeki durumları sağlayan self-dual kodlara özel isimler verilmiştir. (ii) şıkkındaki kodlara yani böleni 4 olan katlı-çift self-dual binary kodlara özel olarak *Tip 2* kod; (iii) şıkkındaki böleni 3 olan self-dual ternary kodlara özel olarak *Tip 3* kod; (iv) şıkkındaki F_4 üzerinde tanımlı böleni 2 olan Hermitian self-dual kodlara özel olarak *Tip 4* kod denir. (ii) şıkkındaki kodlar aynı zamanda (i) şıkkındaki koşulları da sağlamaktadır. Dolayısıyla Tip 2 olmayan binary self-dual kodlara özel olarak *Tip 1* kod denir. Diğer bir deyişle Tip 1 kodlar tekli-çift self-dual kodlar; Tip 2 kodlar katlı-çift self-dual kodlardır [4].

5.1 Gleason Polinomları

Bir kod kullanılırken kodun doğru çözülme olasılığı büyük önem taşır. Uygun bir durumda bu olasılığın hesaplanmasında kodun ağırlık dağılımının bilinmesi oldukça faydalıdır. Fakat bunu her kod için hatta bazı çok uzun olmayan kodlar için bile hesaplayabilmek oldukça zor bir iştir.

Gleason tarafından 1970 yılında Paris'te uluslararası bir konferansta F_2, F_3 ve F_4 cisimleri üzerinde tanımlı self-dual bir kodun ağırlık sayacının bulunabilmesini sağlayan özel

polinomlar tanıtılmış ve Gleason bu polinomları kendi adını vererek [40] da yayınlamıştır. Gleason polinomları teorik olarak önemlidir çünkü bu polinomlar sayesinde self-dual kodların ağırlık dağılımlarının hesaplanması önemli ölçüde kolaylaşmaktadır. Bu polinomlar F_2, F_3 ve F_4 cisimleri üzerindeki self-dual kodların ağırlık dağılımlarını üreten ağırlık sayaç polinomlarıdır.

Teorem 5.1.1. *(Gleason Teoremi)* $q = 2, 3$ *veya 4 için* C, F_q *üzerinde tanımlı bir* $[n, \frac{n}{2}]$ *kod olsun.*

$g_1(x, y) = y^2 + x^2$,

$g_2(x, y) = y^8 + 14x^4y^4 + x^8$,

$g_3(x, y) = y^{24} + 759x^8y^{16} + 2576x^{12}y^{12} + 759x^{16}y^8 + x^{24}$,

$g_4(x, y) = y^4 + 8x^3y$,

$g_5(x, y) = y^{12} + 264x^6y^6 + 440x^9y^3 + 24x^{12}$,

$g_6(x, y) = y^2 + 3x^2$ *ve*

$g_7(x, y) = y^6 + 45x^4y^2 + 18x^6$ *olmak üzere*;

(i) Eğer $q = 2$, C *formally self-dual ve çift ise,* C *kodunun ağırlık sayaç polinomu aşağıdaki gibidir.*

$$W_c(x, y) = \sum_{i=0}^{\lfloor \frac{n}{8} \rfloor} a_i g_1(x, y)^{\frac{n}{2}-4i} g_2(x, y)^i$$

(ii) Eğer $q = 2$, C *self-dual ve doubly-even ise,* C *kodunun ağırlık sayaç polinomu aşağıdaki gibidir.*

$$W_c(x, y) = \sum_{i=0}^{\lfloor \frac{n}{24} \rfloor} a_i g_2(x, y)^{\frac{n}{8}-3i} g_3(x, y)^i$$

(iii) Eğer $q = 3$ ve C self-dual ise, C kodunun ağırlık sayaç polinomu aşağıdaki gibidir.

$$W_c(x,y) = \sum_{i=0}^{\lfloor \frac{n}{12} \rfloor} a_i g_4(x,y)^{\frac{n}{4}-3i} g_5(x,y)^i$$

(iv) Eğer $q = 4$ ve C Hermitian self-dual ise, C kodunun ağırlık sayaç polinomu aşağıdaki gibidir.

$$W_c(x,y) = \sum_{i=0}^{\lfloor \frac{n}{6} \rfloor} a_i g_6(x,y)^{\frac{n}{2}-3i} g_7(x,y)^i$$

Burada bütün a_i katsayıları rasyoneldir ve $\sum_i a_i = 1$ dir [4].

Örnek 5.1.2. Gleason teoremine göre çift ağırlıklı bir binary self-dual kodun ağırlık sayaç polinomu $g_1(x,y)$ ve $g_2(x,y)$ polinomlarının bir kombinasyonudur. Bazı küçük n değerleri için bu durumu inceleyelim;

$n = 2$ için, C_1={00, 11} self-dual kodunun ağırlık sayaç polinomu,

$$W_{C_1}(x,y) = g_1(x,y) = x^2 + y^2$$

$n = 4$ için, C_1^2 self-dual [4,2] kodun ağırlık sayaç polinomu,

$$W_{C_1^2}(x,y) = g_1(x,y)^2 = (x^2+y^2)^2$$

$n = 6$ için, C_1^3 self-dual [6,3] kodun ağırlık sayaç polinomu,

$$W_{C_1^3}(x,y) = g_1(x,y)^3 = (x^2+y^2)^3$$

$n = 8$ için, C_1^4 self-dual [8,4] kodun ağırlık sayaç polinomu,

$$W_{C_1^4}(x,y) = a_0 g_1(x,y)^4 + a_1 g_2(x,y) = a_0(x^2+y^2)^4 + a_1(x^8 + 14x^4y^4 + y^8)$$

elde edilir.

Bu polinomun tam olarak belirlenebilmesi için a_0 ve a_1 katsayılarının bulunması gerekir. Bu polinom daha açık bir şekilde yazıldığında aşağıdaki gibi olur.

$$W_{C_1^4}(x,y) = (a_0 + a_1)x^8 + 4a_0x^6y^2 + (6a_0 + 14a_1)x^4y^4 + 4a_0x^2y^6 + (a_0 + a_1)y^8$$

Öncelikle A_0=1 olmalıdır. Gleason teoreminden $a_0 + a_1$=1 olduğu görülür. a_0=1, a_1=0 ve a_0=0, a_1=1 durumları için aşağıdaki tablo elde edilir.

	A_0	A_2	A_4	A_6	A_8
a_0	1	4	6	4	1
a_1	1	0	14	0	1

A_2=4a_0 olduğundan $a_0 \geq 0$ ve 4a_0 bir tamsayı olmalıdır. Böylece A_4=14a_1+6a_0=14-8a_0 elde edilir ve buradan a_0<2 olduğu söylenir. Buna göre a_0 için olası değerler 0, $\frac{1}{4}$, $\frac{2}{4}$, $\frac{3}{4}$, $\frac{4}{4}$, $\frac{5}{4}$, $\frac{6}{4}$ ve $\frac{7}{4}$ 'tür. Bu sekiz olası değere göre oluşturulan tablo aşağıdaki gibidir;

	1	2	3	4	5	6	7	8
A_0	1	1	1	1	1	1	1	1
A_2	0	1	2	3	4	5	6	7
A_4	14	12	10	8	6	4	2	0
A_6	0	1	2	3	4	5	6	7
A_8	1	1	1	1	1	1	1	1

a_0=0 durumunda birinci çözüm elde edilir ve bu ağırlık dağılımı aslında self-dual uzatılmış [8,4,4] Hamming koduna aittir. Bu durumun dışındaki çözümlerde $A_2 \leq 4$, $A_6 \leq 4$ ve $A_4 \leq 6$ olmalıdır. Aksi takdirde kod self-dual olma özelliğini kaybeder. Dolayısıyla [8,4] bir binary self-dual kod için geriye kalan olası tek çözüm beşinci çözümdür. Bu da C_1^4 kodunun ağırlık dağılımını verir. A_0=A_8=1, A_2=A_6=4 ve A_4=6 olur.

Sonuç 5.1.3. *(i) n uzunluklu bir self-dual binary kod vardır* $\Leftrightarrow n|8$ *dir.*

(ii) n uzunluklu bir self-dual ternary kod vardır $\Leftrightarrow n|4$ *tür.*

(iii) F_4 *üzerinde tanımlı* n *uzunluklu Hermitian self-dual bir kod vardır* $\Leftrightarrow$ n *bir çift sayıdır [4].*

Teorem 5.1.4. *(Gleason Teoreminin Minimum Ağırlık Üzerinde Etkisi)* $q = 2, 3$ *veya 4 için* C, F_q *üzerinde tanımlı* $[n, \frac{n}{2}, d]$ *kod olsun.*

(i) $q = 2$ *ve* C *çift ağırlıklı bir biçimsel self-dual kod ise,*

$$d \leq 2[n/8] + 2$$

(ii) $q = 2$ *ve* C *katlı-çift self-dual kod ise;*

$$d \leq 4[n/24] + 4$$

(iii) $q = 3$ *ve* C *self-dual kod ise,*

$$d \leq 3[n/12] + 3$$

(iv) $q = 4$ *ve* C, *Hermitian self-dual kod ise,*

$$d \leq 2[n/6] + 2$$

Bütün bu durumlarda eşitsizlik sınırda sağlanıyorsa C *kodunun ağırlık sayacı tektir [34].*

Tanım 5.1.5. Olabilecek en büyük minimum ağırlığa sahip bir self-dual kod *extremal kod* olarak isimlendirilir. Dolayısıyla bu teoremdeki sınırları gören Tip 2, Tip 3, Tip 4 kodlar extremal olur [4].

Teorem 5.1.6. C *bir* $[n, \frac{n}{2}, d]$ *binary self-dual kod olsun.*

(i) $d \leq 4\lfloor \frac{n}{24} \rfloor + 4$, $n \not\equiv 22 (mod 24)$

(ii) $d \leq 4\lfloor \frac{n}{24} \rfloor + 6$, $n \equiv 22 (mod 24)$ *dir.*

Ayrıca, eğer $24|n$ *ve* $d = 4\lfloor \frac{n}{24} \rfloor + 4$ *ise* C*, Tip 2 kod olur [33] .*

Tanım 5.1.7. Tip 1 kodlar için Teorem 5.1.6 daki sınırları gören kodlar *extremal* koddur. Extremal bir kodun ağırlık sayacı Gleason polinomlarının kombinasyonu biçimindedir ve *extremal ağırlık sayacı* olarak adlandırılır.

5.2 Self-Dual Kodların Sayılması ve Sınıflandırılması

Teorem 5.2.1. *n uzunluklu binary self-dual kodların toplam sayısı;*

$$\prod_{i=1}^{\frac{n}{2}-1} (2^i + 1)$$

olarak hesaplanır [4].

İspat . Binary bir self-dual kod 1=(11...1) vektörünü içermelidir. $\sigma_{n,k}$, 1 vektörünü içeren self-ortogonal $[n, k]$ kodların sayısını temsil etsin. Öncelikle $k = 1$ için $\sigma_{n,1}$=1 olur. Burada $\sigma_{n,k}$ için bir yineleme bağıntısının bulunması gerekliliği ortaya çıkar. Bu bağıntının bulunabilmesi için önce $\sigma_{n,2}$ yi hesaplanarak durum incelensin. Buna göre, burada 0 ya da 1 vektöründen farklı 2^{n-1}-2 adet çift ağırlıklı vektör vardır. Bütün bu vektörlerin her biri 1 vektörünü içeren tek bir $[n, 2]$ self-ortogonal kodu tarafından kapsanır ve buradaki her bir kod bu vektörlerin 2 tanesini içerir. Böylece $\sigma_{n,2}$=2^{n-2}-1 olur.

1 vektörünü içeren bütün $[n, k+1]$ tipindeki self-ortogonal kodlar, yine 1 vektörünü içeren $[n, k]$ tipinde bir self-ortogonal kodu içerir. 1 vektörünü içeren $[n, k]$ tipinde bir self-ortogonal kodu kullanarak bu kodu içeren $[n, k + 1]$ tipinde bir C' self-ortogonal kod elde edebilmek için C nin $C^{\perp}$ deki kendisinden farklı kosetlerinin herhangi birinden alınan bir c' vektörünün C kodunun üreteç matrisine bir satır olarak eklenmesi gerekir. c' vektörünün seçilebileceği kosetlerin sayısı 2^{n-2k}-1 dir. Böylece C, 2^{n-2k}-1 adet farklı $[n, k + 1]$ C' self-ortogonal koduna uzatılabilir. Ayrıca buradaki herbir C' kodu 2^k-1 adet 1 vektörünü içeren k boyutlu alt koda sahiptir. Böylece $\sigma_{n,k+1} = \frac{2^{n-2k}}{2^k-1} \cdot \sigma_{n,k}$ yineleme bağıntısı elde edilir.

$k = 2$ için yukarıda elde edilen bağıntı ile daha önce elde ettiğimiz gibi $\sigma_{n,2}$=2^{n-2}-1 olur. Yineleme bağıntısı self-dual kodlar için yeniden yazıldığında n uzunluklu bir self-dual kod için,

$$\sigma_{n,\frac{n}{2}} = \frac{2^2-1}{2^{\frac{n}{2}-1}-1} \cdot \frac{2^4-1}{2^{\frac{n}{2}-2}-1} \cdot \frac{2^6-1}{2^{\frac{n}{2}-3}-1} \cdots \frac{2^{n-2}-1}{2-1} \cdot \sigma_{n,1}$$

$$= \frac{2^2-1}{2-1} \cdot \frac{2^4-1}{2^2-1} \cdot \frac{2^6-1}{2^3-1} \cdots \frac{2^{n-2}-1}{2^{\frac{n}{2}-1}-1}$$

$= (2^1+1) \cdot (2^2+1) \cdot (2^3+1) \cdots (2^{\frac{n}{2}-1}+1)$ elde edilir. Bu da istenen sonuçtur.

Örnek 5.2.2. Teorem 5.2.1 e göre [4,2] self-dual binary kodların sayısı 2^1+1=3 tür. Bu kodların üreteç matrisleri sırasıyla aşağıdaki gibidir.

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Bu matrisler birbirine denk olduğundan [4,2] tipinde sadece bir tane binary self-dual kod vardır.

Teorem 5.2.3. *$n \equiv 0(mod8)$ olsun. Bu durumda*

(i) n uzunluklu Tip 2 self-dual kodların sayısı $\prod_{i=0}^{\frac{n}{2}-1}(2^i+1)$ dir.

(ii) C, 1 vektörünü içeren $[n,k]$ doubly-even bir kod olsun. Buna göre, C yi içeren n uzunluklu Tip 2 kodların sayısı $\prod_{i=1}^{\frac{n}{2}-k-1}(2^i+1)$ dir.

Teorem 5.2.4. *(i) $n \equiv 0(mod4)$ olsun. Bu durumda F_3 üzerinde tanımlı n uzunluklu Tip 3 kodların sayısı $2\prod_{i=1}^{\frac{n}{2}-1}(3^i+1)$ dir.*

(ii) $n \equiv 0(mod2)$ olsun. Bu durumda F_4 üzerinde tanımlı n uzunluklu Tip 4 kodların sayısı $\prod_{i=0}^{\frac{n}{2}-1}(2^{2i}+1)$ dir.

İspat . Teorem 5.2.3 ve 5.2.4 ün ispatları Teorem 5.2.1 in ispatına benzer şekilde yapılır.

5.2.1 Mass Formülleri

Teorem 5.2.1 yardımıyla n uzunluklu binary self-dual kodların toplam sayısı bilinmektedir. Bu kodların bir kısmı birbirine denk olduğundan, denklik sınıflarından her biri için birer temsilci bulunması gerekmektedir. Bu durum Sınıflandırma Problemi olarak isimlendirilir. Bu problemin çözülebilmesi için birbirine denk olmayan kodların sayısının bilinmesine ihtiyaç vardır. Dolayısıyla bu sayıyı veren direkt bir formül olmasa da birbirine denk olmayan self-dual kod sınıflarından her biri için bir temsilci elde etmemizi sağlayan bir formül yardımıyla problem çözülebilir. Bu formül Mass Formülü olarak adlandırılır ve aşağıdaki gibi tanımlanır [5].

n uzunluklu binary self-dual kodlardan oluşan her bir denklik sınıfının temsilcileri sırasıyla $C_1, \dots C_s$ olsun. Buna göre $1 \leq j \leq s$ olmak üzere bir C_j koduna denk olan kodların sayısı $\frac{|Sym_n|}{|PAut(C_j)|} = \frac{n!}{|PAut(C_j)|}$ olur. Burada j üzerinde toplam alınarak self-dual kodların toplam sayısını veren aşağıdaki formül binary self-dual kodlar için Mass Formülü elde edilir.

$$\sum_{j=1}^{s} \frac{n!}{|PAut(C_j)|} = \prod_{i=1}^{\frac{n}{2}-1} (2^i + 1)$$

Tip 2, Tip 3 ve Tip 4 kodlar için de Mass formülü benzer yolla hesaplandığı takdirde aşağıdaki teorem elde edilir.

Teorem 5.2.5. *(i) n uzunluklu self-dual binary kodlar için Mass Formülü,*

$$\sum_{j} \frac{n!}{|PAut(C_j)|} = \prod_{i=1}^{\frac{n}{2}-1} (2^i + 1)$$

şeklindedir.

(ii) n uzunluklu Tip 2 kodlar için Mass formülü,

$$\sum_{j} \frac{n!}{|PAut(C_j)|} = \prod_{i=0}^{\frac{n}{2}-2} (2^i + 1)$$

şeklindedir.

(iii) n uzunluklu Tip 3 kodlar için Mass formülü,

$$\sum_j \frac{2^n.n!}{|MAut(C_j)|} = 2\prod_{i=1}^{\frac{n}{2}-1}(3^i+1)$$

şeklindedir.

(iv) n uzunluklu Tip 4 kodlar için Mass formülü,

$$\sum_j \frac{2.3^n.n!}{|\Gamma Aut(C_j)|} = \prod_{i=0}^{\frac{n}{2}-1}(2^{2i+1}+1)$$

şeklindedir.

Uyarı 5.2.6. Her durumda bütün j ler için toplam hesaplanır, burada $\{C_j\}$ verilen tipte birbirine denk olmayan kodların oluşturduğu bütün denklik sınıflarının temsilcilerinin kümesidir.

5.2.2 Sınıflandırma

Kod sınıfları için Mass formüllerinin var olması, kodların sınıflandırılabilmesine olanak sağlar. n uzunluklu binary self-dual kodların sınıflandırılması için aşağıdaki algoritma kullanılır.

I. n uzunluklu bir C_1 binary self-dual kodu bulunur.

II. $PAut(C_1)$ otomorfizm grubunun eleman sayısı bulunur ve $\frac{n!}{|PAut(C_j)|}$ hesaplanır.

III. Daha önce elde edilen self-dual kodlara denk olmayan n uzunluklu bir self-dual kodu bulunur. Bu kod için de $PAut(C_j)$ otomorfizm grubunun eleman sayısı belirlenerek $\frac{n!}{|PAut(C_j)|}$ hesaplanır. Bulunan sonuç Teorem 5.2.5 nin (i) şıkkındaki eşitliğin sol tarafını sağlatmak üzere daha önce elde edilen değerlere eklenir.

IV. Üçüncü adım, Teorem 5.2.5 nin (i) şıkkındaki eşitliğin sağ tarafındaki toplam sayı elde edilene kadar tekrarlanır.

Tip 2, Tip 3 ve Tip 4 kodlar için benzer şekilde sınıflandırma algoritmaları elde edilir.

6. BÖLÜM

KOD İNŞASI

Güvenli ve sorunsuz iletişimin sağlanması, kaydedilen verilerin daha az yer kaplaması ve depolanmış verilerde oluşan hataların giderilmesi gibi çeşitli amaçlara hizmet edecek kod türleri, araştırmacılar tarafından kodlama ve bilgi teorisinin başlangıcından bu yana bulunmaya çalışılmıştır. Bunun için çok çeşitli kod inşa metodları geliştirilmiştir. Bu metodlarla, belirli parametrelere sahip kodlar bazen direkt olarak inşa edilmeye çalışılmış, bazen de var olan kodlardan daha iyi kodlar elde edilmeye çalışılmıştır. Bu bölümde, genel olarak kullanılan bazı kod inşa metodları ve daha özel amaçlarla geliştirilmiş olan çeşitli metodlar incelenmiştir.

6.1 GENEL KOD İNŞA METODLARI

6.1.1 Kodların Delinmesi

Tanım 6.1.1. C, F_q üzerinde tanımlı bir $[n, k, d]$ kod olsun. Kod kümesindeki herbir kod kelimesinde belirli bir i. koordinatın silinmesiyle elde edilen koda C nin *delinmiş kodu* denir ve C^* ile gösterilir. C^* kodu $n-1$ uzunluklu lineer koddur. C kodunun üreteç matrisi G ise, C^* kodunun üreteç matrisi G matrisinden i. sütunun çıkarılmasıyla elde edilir [2].

Teorem 6.1.2. *C, F_q üzerinde bir $[n, k, d]$ kod ve C kodunun i. koordinatının delinmesiyle elde edilen kod C^* olsun. Buna göre,*

(i) $d > 1$ için eğer C kodu, i. koordinatı sıfırdan farklı minimum ağırlıklı bir kod kelimesi içeriyorsa C^* bir $[n-1, k, d^*]$ kod olur ve burada $d^* = d-1$ dir. Aksi takdirde $d^* = d$ dir.

(ii) $d = 1$ *için eğer* C *kodu,* i*. koordinatı sıfırdan farklı ve ağırlığı 1 olan hiçbir kod kelimesi içermiyorsa* C^* *bir* $[n-1, k, d^*]$ *kod olur. Aksi takdirde eğer* $k > 1$ *ise* C^* *bir* $[n-1, k-1, d^*]$ *koddur ve burada* $d^* \geq 1$ *dir [4].*

İspat . C kodu, q^k adet kod kelimesi içermektedir. Bu yüzden, C^* kodunun daha az sayıda kod kelimesine sahip olabilmesinin tek yolu, C kodunda sadece i. koordinatı birbirinden farklı ve diğer bütün koordinatları aynı olan iki kod kelimesinin var olmasıdır. Bu durumda $d = 1$ olur ve C kodunda sadece i. koordinatı sıfırdan farklı olan 1 ağırlıklı bir kod kelimesi vardır. Buna göre, C^* kodunun parametrelerinin belirlenebilmesi için $d = 1$ ve $d > 1$ durumlarına ayrı ayrı bakılır.

(i) $d > 1$ olsun. Bu durumda, C^* kodunun boyutu k olacaktır. Eğer C kodunda i. koordinatı sıfırdan farklı minimum ağırlıklı bir kod kelimesi varsa $d^* = d - 1$ olur. Aksi takdirde $d^* = d$ dir. Dolayısıyla C^*, $[n-1, k, d^*]$ tipinde bir kod olur.

(ii) $d = 1$ olsun. Bu durumda, $k > 1$ için, C kodunda sadece i. bileşeni sıfırdan farklı olan 1 ağırlıklı bir kod kelimesi varsa C^* kodunun boyutu $k-1$ olacaktır. Yani, $d^* \geq 1$ için, C^* kodu $[n-1, k-1, d^*]$ tipinde bir kod olur. Aksi takdirde, C^* kodunun boyutu k olur ve minimum uzaklık değişmez. C^*, $[n-1, k, 1]$ tipinde bir kod olur.

Örnek 6.1.3. C kodu üreteç matrisi aşağıda verilen bir [5,2,2] binary kod olsun.

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

C_1^* ve C_5^* sırasıyla C kodunun 1. ve 5. koordinatlarının silinmesiyle elde edilen delinmiş kodlar olsun. Buna göre bu kodların üreteç matrisleri aşağıdaki gibidir.

$$G_1^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \text{ ve } G_5^* = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Böylece, C_1^* binary [4,2,1] kod ve C_5^* binary [4,2,2] kod olur.

Örnek 6.1.4. D kodu üreteç matrisi aşağıda verilen bir [4,2,1] binary kod olsun.

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

D kodunun 1. ve 4. koordinatlarının silinmesiyle elde edilen delinmiş kodlar sırasıyla D_1^* ve D_4^* olsun. Buna göre bu kodların üreteç matrisleri sırasıyla aşağıdaki gibi olur.

$$D_1^* = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \text{ ve } D_4^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Böylece D_1^* bir [3,1,3] kod ve D_4^* bir [3,2,1] kod olur. Burada Örnek 6.1.3 te yer alan C_1^* kodu ile D kodunun aynı kodlar olduğu görülür. Dolayısıyla C kodundan {1,5} pozisyonlarının çıkarılmasıyla elde edilen delinmiş kod D kodu olur. Kod delinmesi daha genel bir ifadeyle tanımlanacak olursa T bir koordinat kümesi olmak üzere, F_q üzerinde tanımlı bir C $[n, k, d]$ kodu T kümesinde bulunan bütün koordinatların kod kelimelerinden silinmesiyle delinebilir. Eğer T kümesinin eleman sayısı t ise, $k^* \geq k - t$ ve $d^* \geq d - t$ olmak üzere C^T delinmiş kodu bir $[n - t, k^*, d^*]$ kod elde edilir.

6.1.2 Kodların Uzatılması

Var olan kodlara yeni bir koordinat eklenerek daha uzun kodlar oluşturulabilir. Bu ekleme çok çeşitli yollarla yapılabilmesine rağmen en çok kullanılan metod kod kümesindeki bütün kod kelimelerinin çiftimsi vektörler oluşturmasını sağlayacak şekilde ekleme yapılmasıdır. Dolayısıyla böyle bir ekleme yapılarak bir kodun uzatılması aşağıdaki gibi tanımlanır.

Tanım 6.1.5. C, F_q üzerinde bir $[n, k, d]$ kod olmak üzere C kodunun *uzatılmış kodu*

$$\widehat{C} = \{x_1x_2...x_nx_{n+1} \in F_q^{n+1} | x_1x_2...x_n \in \text{C ve } x_1 + x_2 + ... + x_n + x_{n+1} = 0\} \text{ dir.}$$

$\widehat{d} = d$ veya $\widehat{d} = d + 1$ olmak üzere, $\widehat{C}$ kodu lineer bir $[n + 1, k, \widehat{d}]_q$ koddur [4].

C kodunun üreteç matrisi G ve eşlik-kontrol matrisi H olsun. G matrisine her satırının koordinatlarının toplamı 0 olacak şekilde yeni bir sütun eklenmesiyle elde edilen $\widehat{G}$ matrisi, $\widehat{C}$ uzatılmış kodunun üreteç matrisi olur. $\widehat{C}$ kodunun eşlik-kontrol matrisi aşağıdaki gibi olur.

$$\widehat{H} = \left[\begin{array}{c|c} 1 \ldots 1 & 1 \\ \hline & 0 \\ \text{H} & \vdots \\ & 0 \end{array}\right]$$

Bir C kodunun bu şekilde uzatılması *ayrıntılı eşlik-kontrol (overall parity-check) biti eklenmesi* olarakta isimlendirilir.

Örnek 6.1.6. $[7,4,3]_2$ H_3 Hamming koduna ayrıntılı eşlik-kontrol biti eklenmesi ile $[8,4,4]_2$ $\widehat{H}_3$ uzatılmış Hamming kodu elde edilir. Bu kodların üreteç matrisleri sırasıyla aşağıda verilmiştir.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \text{ ve } \widehat{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Eğer C bir $[n,k,d]$ binary kod ise $\widehat{C}$ uzatılmış kodu sadece çift ağırlıklı vektörleri içeren $[n+1,k,\widehat{d}]$ binary koddur. Burada eğer d çiftse $\widehat{d} = d$, d tek ise $\widehat{d} = d+1$ dir.

Tanım 6.1.7. F_q üzerinde tanımlı C kodu bir $[n,k,d]$ kod olsun. C kodunda çiftimsi kod kelimelerinin minimum ağırlığına kısaca *minimum çiftimsi ağırlık* denir ve d_e ile gösterilir; tekimsi kod kelimelerinin minimum ağırlığına ise *minimum tekimsi ağırlık* denir ve d_o ile gösterilir. Böylece $d = min\{d_e, d_o\}$ olur. Eğer $d_e \leq d_o$ ise $\widehat{C}$ uzatılmış kodunun minimum ağırlığı $\widehat{d} = d_e$ dir. Aksi takdirde $d_o < d_e$ ise $\widehat{d} = d_o + 1$ dir.

Örnek 6.1.8. F_3 üzerinde tanımlı $H_{3,2}$ tetrakodun üreteç ve eşlik kontrol matrisi sırasıyla aşağıdaki gibidir.

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix} \text{ ve } H = \begin{bmatrix} -1 & -1 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix}$$

(1,0,1,1) kod kelimesinin uzatılmış hali (1,0,1,1,0) ve (0,1,1,-1) kod kelimesinin uzatılmış hali (0,1,1,-1,-1) dir. Böylece $d = d_e = d_o = 3$ ve $\widehat{d} = 3$ olur. $\widehat{H}_{3,2}$ kodunun üreteç ve eşlik kontrol matrisleri sırasıyla aşağıdaki gibidir.

$$\widehat{G} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & -1 & -1 \end{bmatrix} \text{ ve } \widehat{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Bir kod önce uzatılır, daha sonra elde edilen yeni koordinat silinerek delinirse çıkan sonuç yine kodun kendisi olur. Bu işlemlerin uygulanma sırası değiştiği takdirde elde edilen kod orjinal koddan farklı olabilir.

Örnek 6.1.9. Üreteç matrisi G olan binary C kodunun son koordinatını silerek bu kodu delelim ve daha sonra elde edilen delinmiş kodu uzatalım. Bu durumda;

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

iken sonuçta elde edilen kodun üreteç matrisi

$$G_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

olur ki bu matrisin ürettiği kod C kodundan farklıdır.

6.1.3 Kodların Kısaltılması

Tanım 6.1.10. C, F_q üzerinde tanımlı bir $[n, k, d]$ kod ve T, t adet koordinat içeren bir küme olsun. C kodunda, T kümesindeki koordinatları 0 olan kod kelimelerinin kümesi $C(T)$ ile gösterilsin. Bu durumda $C(T)$, C kodunun bir alt kodu olur. $C(T)$ kodunun T kümesinde yer alan koordinatlarının silinmesiyle elde edilen delinmiş kod F_q üzerinde $n - t$ uzunluklu bir kod olur. Bu kod, C kodunun T üzerinde *kısaltılmış kodu* olarak isimlendirilir ve C_T ile gösterilir [4].

Örnek 6.1.11. C, üreteç matrisi G olan [6,3,2] binary kod olsun.

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

$C^{\perp}$ kodu ise yine [6,3,2] kod olur ve üreteç matrisi aşağıdaki gibidir.

$$G^{\perp} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Kod kelimelerinin koordinatları sırasıyla 1,2,. . .,6 ile etiketlensin ve $T = \{5, 6\}$ olsun. C kodunun kısaltılmış kodu C_T ve delinmiş kodu C^T ise bu kodların üreteç matrisleri sırasıyla aşağıdaki gibidir.

$$G_T = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \text{ ve } G^T = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Dual kodun kısaltılmış ve delinmiş kodları sırasıyla $(C^{\perp})_T$ ve $(C^{\perp})^T$ ise bu kodların üreteç matrisleri sırasıyla aşağıdaki gibidir.

$$(G^{\perp})_T = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \text{ ve } (G^{\perp})^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

C_T ve C^T kodlarının duallerinin üreteç matrisleri sırasıyla aşağıdaki gibidir.

$$(G_T)^{\perp} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ ve } (G^T)^{\perp} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

Bu matrislere göre $(C^{\perp})_T = (C^T)^{\perp}$ ve $(C^{\perp})^T = (C_T)^{\perp}$ olur.

Teorem 6.1.12. *C, F_q üzerinde tanımlı bir $[n, k, d]$ kod ve T, t elemanlı bir koordinat kümesi olsun. Buna göre, $(C^{\perp})_T = (C^T)^{\perp}$ ve $(C^{\perp})^T = (C_T)^{\perp}$ dir [3].*

İspat . C kodunda T kümesindeki koordinatları sıfır olan bir kod kelimesi c olsun. c kelimesinden T kümesindeki koordinatların çıkarılmasıyla elde edilen kelime c^* olsun. O zaman, $c^* \in (C^{\perp})_T$ dir. $x \in C$ ise $0 = x \cdot c = x^* \cdot c^*$ olur ve burada x^*, x kelimesinin T kümesindeki koordinatlara göre delinmesiyle oluşan kelimedir. Böylece $(C^{\perp})_T \subseteq (C^T)^{\perp}$ olur. Herhangi bir $c \in (C^T)^{\perp}$ kod kelimesi, T kümesindeki koordinatlarına sıfır yerleştirilerek $\widehat{c}$ kelimesine uzatılabilir. $x \in C$ ise, x kelimesinin T ye göre delinmesiyle elde edilen kelime x^* olsun. $0 = x^* \cdot c = x \cdot \widehat{c}$ ve buradan $c \in (C^{\perp})_T$ elde edilir. Dolayısıyla, $(C^{\perp})_T = (C^T)^{\perp}$ olur. Benzer şekilde $(C^{\perp})^T = (C_T)^{\perp}$ elde edilir.

6.1.4 Direkt Toplam

Tanım 6.1.13. C_1 ve C_2 aynı F_q cisimi üzerinde tanımlı sırasıyla $[n_1, k_1, d_1]$ kod ve $[n_2, k_2, d_2]$ kod olsunlar. Buna göre bu kodların *direkt toplamı*

$$C_1 \oplus C_2 = \{(c_1, c_2) | c_1 \in C_1, c_2 \in C_2\}$$

şeklinde tanımlanan bir $[n_1 + n_2, k_1 + k_2, min\{d_1, d_2\}]$ koddur [3].

C_1 ve C_2 kodlarının üreteç matrisleri ve eşlik-kontrol matrisleri sırasıyla G_1, G_2, H_1 ve H_2 olsun. Buna göre;

$$G_1 \oplus G_2 = \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix} \text{ ve } H_1 \oplus H_2 = \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix}$$

matrisleri $C_1 \oplus C_2$ direkt toplam kodunun sırasıyla üreteç ve eşlik kontrol matrisleridir.

Örnek 6.1.14. $D = \{00, 11\}$ binary [2,1,2] kodunun $D \oplus D \oplus D$ direkt toplamı G üreteç matrisi aşağıda verilen [6,3,2] binary koddur.

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

6.1.5 (u|u+v) Yapılanması

$(u|u+v)$ ile gösterilen kod yapılanması, aynı uzunluğa sahip iki kodun direkt toplama benzer bir yolla birleştirilerek 2 kat daha uzun üçüncü bir kod elde edilmesidir.

Tanım 6.1.15. C_1 ve C_2 aynı F_q cisimi üzerinde tanımlı sırasıyla $[n_1, k_1, d_1]$ kod ve $[n_2, k_2, d_2]$ kod olsunlar. $(u|u+v)$ yapılanması ile bu iki koddan elde edilen C kodu,

$$C = \{(u|u+v) : u \in C_1, v \in C_2\}$$

biçiminde tanımlanır [3].

Burada C bir $[2n, k_1 + k_2, min\{2d_1, d_2\}]$ q-luk kod olur. C_1 ve C_2 kodlarının üreteç ve eşlik-kontrol matrisleri sırasıyla G_1, G_2 ve H_1, H_2 olsun. Buna göre C kodunun üreteç ve eşlik-kontrol matrisleri sırasıyla

$$G = \begin{bmatrix} G_1 & G_1 \\ 0 & G_2 \end{bmatrix} \text{ ve } H = \begin{bmatrix} H_1 & 0 \\ -H_2 & H_2 \end{bmatrix} \text{ matrisleridir.}$$

Örnek 6.1.16. Üreteç matrisi G olan [8,4,4] binary kodu C olsun.

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

Üreteç matrisleri sırasıyla aşağıda verilen G_1 ve G_2 matrisleri olan, C_1 binary [4,3,2] ve C_2 binary [4,1,4] kodlarına $(u|u+v)$ yapılanması uygulanarak C kodu üretilir.

$$G_1 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \text{ ve } G_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

Ayrıca C_1 kodu, üreteç matrisleri sırasıyla,

$$G_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ ve } G_4 = \begin{bmatrix} 1 & 1 \end{bmatrix}$$

olan [2,2,1] binary C_3 kodu ve binary [2,1,2] C_4 koduna $(u|u+v)$ yapılanması uygulanarak üretilir.

6.1.6 Döngüsel Yapılanmalar

Kod inşasında döngüsel yapılanmaların sıkça kullanılmasının başlıca sebebi, tek bir satırın bütün matrisi belirlemeye yeterli olmasıdır. Bu durum, döngüsel yapılanmaya sahip kodların elde edilebilmesi, parametlerinin belirlenebilmesi, sınıflandırılabilmesi gibi konularda bir hayli kolaylık sağlamaktadır.

Tanım 6.1.17.

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_m \\ a_m & a_1 & a_2 & \dots & a_{m-1} \\ a_{m-1} & a_m & a_1 & \dots & a_{m-2} \\ & & \vdots & & \\ a_2 & a_3 & a_4 & \dots & a_1 \end{bmatrix}, B = \begin{bmatrix} \alpha & \beta & \dots & \beta \\ \gamma & & & \\ \gamma & & A & \\ \vdots & & & \\ \gamma & & & \end{bmatrix}$$

Yukarıdaki gibi verilen mxn tipinde bir A matrisine *döngüsel matris*, $(m+1)$x$(m+1)$ tipinde bir B matrisine ise *kenarlı döngüsel matris* denir. Bu durumda I_n birim matrisi hem döngüsel hem de kenarlı döngüsel matris olur [1,4] .

Tanım 6.1.18. A, mxm tipinde bir döngüsel matrisi ve B, $(m+1)$x$(m+1)$ tipinde bir kenarlı döngüsel matrisi temsil etsin. Buna göre, bir kodun üreteç matrisi $[I_m|A]$ formunda ise bu matrise *çift döngüsel üreteç matris*, $[I_{m+1}|B]$ formunda ise bu matrise *kenarlı çift döngüsel üreteç matris* denir. Üreteç matrisinin tipine göre bir kod *çift döngüsel yapılanmaya* ya da *kenarlı çift döngüsel yapılanmaya* sahiptir denir [4].

Örnek 6.1.19.

$$G = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & -1 & -1 & 1 \\ 1 & 1 & 0 & 1 & -1 & -1 \\ 1 & -1 & 1 & 0 & 1 & -1 \\ 1 & -1 & -1 & 1 & 0 & 1 \\ 1 & 1 & -1 & -1 & 1 & 0 \end{bmatrix}$$

Üreteç matrisi $[I|G]$ ile verilen [12,6,6] uzatılmış ternary Golay kodu kenarlı çift döngüsel bir yapılanmaya sahiptir.

Tanım 6.1.20.

$$R = \begin{bmatrix} r_1 & r_2 & r_3 & \dots & r_m \\ r_2 & r_3 & r_4 & \dots & r_1 \\ r_3 & r_4 & r_5 & \dots & r_2 \\ & & \vdots & & \\ r_m & r_1 & r_2 & \dots & r_{m-1} \end{bmatrix}, B = \begin{bmatrix} \alpha & \beta & \dots & \beta \\ \gamma & & & \\ \gamma & & R & \\ \vdots & & & \\ \gamma & & & \end{bmatrix}$$

Yukarıdaki gibi verilen mxm tipinde bir R matrisi *ters döngüsel matris*, $(m+1)$x$(m+1)$ tipinde bir P matrisi ise *kenarlı ters döngüsel matris* olarak isimlendirilir [4].

Tanım 6.1.21. Döngüsel yapılanmaya benzer şekilde eğer bir kodun üreteç matrisi $[I_m|R]$ formunda ise bu matrise *ters döngüsel üreteç matris*; $[I_{m+1}|P]$ formunda ise bu matrise *kenarlı ters döngüsel üreteç matris* denir. Üreteç matrisinin tipine göre bir kod *ters döngüsel* ya da *kenarlı ters döngüsel yapılanmaya* sahiptir denir [4].

Örnek 6.1.22.

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

Üreteç matrisi $[I|A]$ olan [24,12,8] uzatılmış binary Golay kodu kenarlı ters döngüsel bir yapılanmaya sahiptir.

6.2 BAZI ÖZEL KOD İNŞA METODLARI

6.2.1 Negatif Döngüsel Yapılanma

Kod inşasında döngüsel yapılanma kullanılmasının sağladığı avantajlara 6.1.6 alt bölümünde yer verilmişti. Negatif döngüsel yapılanmalar, bu avantajlara ek olarak üretilen kodun self-dual olmasını sağlamak için kullanılır. Ayrıca sembol sayısı çok olan alfabelerde self-dual kod oluşturmak için sağladığı işlem kolaylığından dolayı özellikle tercih edilen bir metoddur [12]. Bu metodla elde edilen bazı kodlara ait üreteç matrislerine yedinci bölümde yer verilmiştir.

Tanım 6.2.1.

$$X = \begin{bmatrix} x_0 & x_1 & x_2 & \dots & x_{m-1} \\ -x_{m-1} & x_0 & x_1 & \dots & x_{m-2} \\ -x_{m-2} & -x_{m-1} & x_0 & \dots & x_{m-3} \\ & & & \vdots & \\ -x_1 & -x_2 & -x_3 & \dots & x_0 \end{bmatrix}$$

Yukarıdaki gibi verilen mxm tipinde bir X matrisine negatif döngüsel matris denir [12].

Tanım 6.2.2. A ve B mxm tipinde iki döngüsel matris ve $AA^T + BB^T = -I_m$ ise,

$$\begin{bmatrix} I_{2m} & A & B \\ & -B^T & A^T \end{bmatrix}$$

matrisi F_q cisimi üzerinde $[4m, 2m]$ bir self-dual kod üretir. Böyle bir koda *dörtlü negatif devreden kod* denir [12].

6.2.2 Eksiltme Metodu

Var olan bir binary self-dual kod kullanılarak bu kodun kapsadığı daha küçük bir binary self-dual kod, eksiltme metodu yardımıyla elde edilir. Dolayısıyla bu şekilde, binary bir self-dual kodun bütün self-dual alt kodları belirlenebilir [4].

Tanım 6.2.3. C, $[2n, n, d]$ tipinde bir binary self-dual kod olsun. C kodunun bir alt kodu olan C_1 kodu aşağıdaki gibi tanımlansın.

$$C_1 = \{(x_1, x_2, \dots, x_{2n}) \in C | x_1 = x_2 = 0 \text{ veya } x_1 = x_2 = 1\}$$

Bu durumda C_1, $n-1$ boyutlu self-ortogonal bir kod olur. C_1 kodu yardımıyla, C koduna *eksiltme* uygulanarak elde edilen C_2 kodu aşağıdaki biçimde tanımlanır.

$$C_2 = \{(x_3, x_4, \dots, x_{2n}) | (x_1, x_2, \dots, x_{2n}) \in C_1\}$$

Burada C_2 kodu $[2n-2, n-1]$ bir binary self-dual kod olur [4].

Örnek 6.2.4. Bir üreteç matrisi $G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$ olan [6,3,2] binary self-dual kodu C olsun. Bu durumda,

$C = \{000000, 111111, 110000, 001100, 000011, 110011, 111100, 001111\}$ olur.

$C_1 = \{(x_1, \ldots, x_6) \in C | x_1 = x_2 = 0\}$ ile tanımlarsak,

$C_1 = \{000000, 001100, 000011, 001111\}$ elde edilir. Böylece C kodunun eksiltilmesiyle elde edilen C_2 kodu; $C_2 = \{0000, 1100, 0011, 1111\}$ şeklindedir. Sonuç olarak elde edilen C_2 kodu [4,2,2] bir binary self-dual kod olur.

6.2.3 Binary Self-Dual Kodların İnşası ve Sınıflandırılması İçin Yinelemeli Algoritma

Self-dual kodlar için, $[n+2, \frac{n}{2}+1, d+2]$ bir koddan $[n, \frac{n}{2}, \geq d]$ bir self-dual kod elde etmeyi mümkün kılan kod eksiltme metodu bilinen bir metoddur. Burada bu metodun tersinin de mümkün olduğu yani, $[n+2, \frac{n}{2}+1, d+2]$ tipinde bütün self-dual kodların $[n, \frac{n}{2}, \geq d]$ tipinde self-dual kodlardan elde edilebileceği gösterilmiştir [11] .

$d \geq 2$ olmak üzere $[n+2, \frac{n}{2}+1, d+2]$ tipinde bir self-dual C_{n+2} kodunun bir üreteç matrisi

$$G = \begin{bmatrix} 1 & 1 & y \\ 0 & 0 & D \\ 0 & 1 & z \end{bmatrix}$$

şeklinde yazılabilir. Burada y, n uzunluklu ve d ağırlıklı bir kod kelimesi, D, $[n, \frac{n}{2}-1]$ tipinde C_{n+2} den birinci ve ikinci koordinatların silinmesiyle elde edilen bir alt kodu ve z, C_{n+2} koduna ait bir kod kelimesinin ilk iki koordinatının silinmesiyle elde edilen bir kod kelimesidir.

$\begin{bmatrix} y \\ D \end{bmatrix}$, $[n, \frac{n}{2}, d]$ tipinde bir self-dual koddur. Burada $[n, \frac{n}{2}, d]$ tipinde bir self-dual kod ile $[n+2, \frac{n}{2}+1, d+2]$ tipinde bir self-dual kodun ilişkilendirilebileceği görülür.

Şimdi, kod eksiltme metodunun aksine C_{n+2} kodunun yukarıda verilen üreteç matrisinin iki sütununun silinmesi ve uygun bir satırının çıkarılmasıyla elde edilen $[n, \frac{n}{2}, d]$ tipindeki bir self-dual C_n kodu ile işleme başlanıldığını varsayalım. Bu durumda y', d ağırlıklı bir kelime olmak üzere, C_n kodunun bir üreteç matrisi $\begin{bmatrix} y' \\ E \end{bmatrix}$ biçiminde yazılabilir. $a_i \in \{0,1\}$ olmak üzere, üreteç matrisi aşağıdaki biçimde olan $[n+2, \frac{n}{2}]$ tipinde C kodlarının kümesi MC ile gösterilsin.

$$\begin{bmatrix} 1 & 1 & y' \\ a_1 & a_1 & \\ \vdots & \vdots & E \\ a_{\frac{n}{2}-1} & a_{\frac{n}{2}-1} & \end{bmatrix}$$

Burada C, self-ortogonal bir koddur. C kodundan C_{n+2} kodunun yeniden elde edilebilmesi için C ye ekleme yapılması gerekmektedir. Yapılacak ekleme sonucunda elde edilecek kodun self-dual olması gerektiğinden bu ekleme için $C^{\perp}/C$ nin kosetlerinin kullanılması uygun olacaktır. Dolayısıyla $C^{\perp}/C$ nin C den farklı herhangi bir elemanının C ye eklenmesi yeterlidir. Burada $C^{\perp}/C$ nin C den farklı bir elemanı $C+x$ ile gösterilirse, $dim(C \cup (C+x)) = \frac{n}{2}+1$ olur ve C self-ortogonal olduğundan $C \cup (C+x)$ de self-ortogonaldir. Bu durumda $dim(C \cup (C+x))^{\perp} = (n+2) - (\frac{n}{2}+1) = \frac{n}{2}+1 = dim(C \cup (C+x))$ dir. Bu da $C \cup (C+x)$ in self-dual olduğunu gösterir.

Böylece $[n+2, \frac{n}{2}+1, d+2]$ tipinde bir C_{n+2} kodunun, $[n, \frac{n}{2}, d]$ tipinde bir C_n alt kodu bilindiği takdirde bu kod kullanılarak C_{n+2} kodunun yeniden inşa edilebileceği ispatlanmış olur.

Burada tek bir $[n, \frac{n}{2}, d]$ kodu ile başlamaktan ziyade birbirine denk olmayan bütün $[n, \frac{n}{2}, d]$ self-dual kodların kümesi ile başlayıp bütün $[n+2, \frac{n}{2}+1, d+2]$ tipindeki kodların yeniden inşa edilmesi daha anlamlı olacaktır. a_i lerin seçiminde ise $2^{\frac{n}{2}-1}$ adet olasılık vardır. C_n kodundaki d ağırlıklı bütün kod kelimelerinin $d+2$ ağırlıklı hale getirilmesi gerekliliği

a_i lerin seçim olasılığının hesaplanmasını daha kısa hale getirir. d ağırlıklı bütün kod kelimelerinin oluşturduğu kod C_d ve bu kodun üreteç matrisi G_d ile gösterilsin. C_d kodunun boyutu k ise $k \leq \frac{n}{2}$ dir. Böylece $a_i$lerin seçiminde $2^{\frac{n}{2}-1}$ yerine $2^{\frac{n}{2}-k}$ adet olasılığa bakmak yeterli olacaktır.

Bütün bu işlemlerin basamaklar halinde yazılmasıyla *yinelemeli algoritma* elde edilir [11].

Yinelemeli Algoritma

Mevcut self-dual kodlardan, bu kodlar kullanılarak daha uzun yeni self-dual kodlar üreten bu algoritma girdisi, çıktısı ve basamakları ile aşağıdaki gibidir.

Girdi : S_n, permütasyon denkliğe göre $[n, \frac{n}{2}, d]$ tipindeki denk olmayan bütün self-dual kodların kümesi

Çıktı : $[n+2, \frac{n}{2}+1, d+2]$ tipindeki denk olmayan bütün self-dual kodlar

S_n deki herbir C_n kodu için,

1) d ağırlıklı bütün kod kelimeleri listelenir ve k boyutlu C_d alt kodu oluşturulur. Sadece d ağırlıklı kelimelerin oluşturduğu bu C_d kodunun bir üreteç matrisi G_d ile belirlenir.

2) $C_n = C_d + E$ eşitliğini sağlayan $n-k$ boyutlu E kodunun bir üreteç matrisi G_E olsun. $1 \leq i \leq n-k$ ve $a_i \in \{0,1\}$ olmak üzere, C_n kodunun uzatılmasıyla elde edilen C kodunun üreteç matrisi aşağıdaki biçimdedir.

$$\begin{bmatrix} 1 & 1 & \\ \vdots & \vdots & G_d \\ 1 & 1 & \\ a_1 & a_1 & \\ \vdots & \vdots & G_E \\ a_{\frac{n}{2}-k} & a_{\frac{n}{2}-k} & \end{bmatrix}$$

3) Bir önceki basamakta elde edilen bütün C kodları $C^{\perp}/C$ nin C den farklı elemanlarından biriyle tamamlanarak self-dual kodlar elde edilir. Elde edilen bu kodların minimum ağırlıklarının $d+2$ olup olmadığı kontrol edilir. $d+2$ ağırlıklı kodların da birbirine denkliği kontrol edilerek bütün $[n+2, \frac{n}{2}+1, d+2]$ binary self-dual kodlar elde edilir.

6.2.4 Self-Dual Kodlar İçin Üst-Yapı İnşa Metodu

Mevcut bir self-dual koddan, bu kod kullanılarak daha uzun yeni bir self-dual kod üreten bu metod binary kodlar için aşağıdaki teoremle açıklanmıştır. Binary olmayan self-dual kodlar için de geçerli olan bir metoddur ve bu bölümün devamında diğer durumlar teoremler ve önermeler yardımıyla açıklanacaktır.

Teorem 6.2.5. *$\{1, 2, ..., 2n\}$ koordinat kümesinin bir alt kümesi S ve $|S|$ bir tek sayı olsun. $1 \leq i \leq n$ olmak üzere $2n$ uzunluklu C_o kodunun bir üreteç matrisi $G_o = (L|R) = (l_i|r_i)$ olsun. Burada l_i ve r_i sırasıyla L ve R nin satırlarıdır. $1 \leq j \leq n$ olmak üzere S kümesinin karakteristik vektörü $x = (x_1, x_2, ..., x_n, x_{n+1}, ..., x_{2n})$ olsun. Burada x vektörü $j \in S$ ise $x_j := 1$ ve $j \notin S$ ise $x_j := 0$ ile tanımlıdır. Ayrıca $1 \leq i \leq n$ için $y_i := (x_1, x_2, ..., x_n, x_{n+1}, ..., x_{2n}) \cdot (l_i|r_i)$ ile tanımlansın. Bu durumda*

$$G = \left[\begin{array}{cc|cc} 1 & 0 & x_1 \ldots x_n & x_{n+1} \ldots x_{2n} \\ \hline y_1 & y_1 & & \\ \vdots & \vdots & L & R \\ y_n & y_n & & \end{array}\right]$$

matrisi $2n+2$ uzunluklu bir self-dual C kodunu üretir [8].

İspat. Üreteç matrisi G olan C kodunun boyutu $n+1$ dir. Buna göre, ispatlanması gereken tek durum G matrisinin herhangi iki satırının ortogonal olmasıdır. $|S|$ tek olduğundan ilk satır kendisi ile ortogonaldir. $G_o = (L|R)$ matrisi self-dual C_o kodunun üreteç matrisi olduğundan ilk satır haricindeki G matrisinin bütün satırları birbirine ve kendine ortogonal olur. Son olarak ilk satırın diğer bütün satırlara ortogonal olduğu aşağıdaki hesaplama yardımıyla görülür. $1 \leq i \leq n$ için,

$$(1, 0, x_1, x_2, ..., x_n, x_{n+1}, ..., x_{2n}) \cdot (y_i, y_i, l_i, r_i)$$

$= y_i + (x_1, x_2, ..., x_n, x_{n+1}, ..., x_{2n}) \cdot (l_i|r_i)$

$= y_i + y_i = 0$

Böylece ispat tamamlanmış olur.

Teorem 6.2.5 e göre, $G_o = (I_n|A)$ biçiminde yazılarak ve $x_1 = x_2 = \ldots = x_{2n} = 1$ alınarak, Harada[13]ya ait aşağıdaki sonuç elde edilir. Bu da bu metodun Harada'nın sonucunun daha genel bir hali olduğunu gösterir.

Sonuç 6.2.6. *$\{1, 2, \ldots, n\}$ koordinat kümesinin bir alt kümesi S olsun öyle ki burada eğer $2n \equiv 0(mod4)$ ise $|S|$ tek ve eğer $2n \equiv 2(mod4)$ ise $|S|$ çift olsun. $2n$ uzunluklu self-dual C_o kodunun standart formda bir üreteç matrisi $G_o = (I_n|A)$ olsun. $1 \leq i \leq n$ için; eğer $i \in S$ ise $x_i := 1$ ve $i \notin S$ ise $x_i := 0$ ayrıca $y_i := x_i + 1$ ile tanımlansın. Bu durumda,*

$$G = \left[\begin{array}{cc|cc} 1 & 0 & x_1 \ldots x_n & 1 \ldots 1 \\ \hline y_1 & y_1 & & \\ \vdots & \vdots & I_n & A \\ y_n & y_n & & \end{array}\right]$$

G matrisi $2n + 2$ uzunluklu bir self-dual C kodunu üretir [13].

Önerme 6.2.7. (GF(q) Üzerinde Tanımlı Self-Dual Kodlar İçin Üst-Yapı İnşa Metodu: $q \equiv 1(mod4)$)

$q \equiv 1(mod4)$ ve bir $c \in GF(q)$ için $c^2 = -1$ olsun. $GF(q)$ üzerinde tanımlı $2n$ uzunluklu self-dual bir C_0 kodunun üreteç matrisi $G_0 = (L|R) = (l_i|r_i)$ olsun. $1 \leq i \leq n$ olmak üzere, burada l_i ve r_i sırasıyla L ve R matrislerinin satırlarıdır. $x = (x_1, x_2, ..., x_n, x_{n+1}, ..., x_{2n})$, $GF(q)^{2n}$ uzayına ait $x \cdot x = -1$ koşulunu sağlayan bir vektör olsun. Bu takdirde $y_i := (x_1, x_2, ..., x_n, x_{n+1}, ..., x_{2n}) \cdot (l_i|r_i)$ ile tanımlandığında aşağıda verilen G matrisi $GF(q)$ üzerinde $2n + 2$ uzunluklu self-dual bir C kodunu üretir.

$$G = \left[\begin{array}{cc|cc} 1 & 0 & x_1 \ldots x_n & x_{n+1} \ldots x_{2n} \\ \hline -y_1 & cy_1 & & \\ \vdots & \vdots & L & R \\ -y_n & cy_n & & \end{array}\right]$$

Burada [8] de binary self-dual kod inşası için verilen üst-yapı inşa metodu $q \equiv 1(mod4)$ olmak üzere $GF(q)$ üzerine genelleştirilmiştir [14]. Bu önermenin ispatı binary self-dual kodlar için üst-yapı inşa metodunda verilen Teorem 6.2.5 e benzer şekilde yapılır.

Önerme 6.2.8. (GF(q) Üzerinde Tanımlı Self-Dual Kodlar İçin Üst-Yapı İnşa Metodu: $q \equiv 3(mod4)$)

n bir çift sayı, $q \equiv 3(mod4)$ ve $\alpha, \beta \in GF(q)^*$ için $\alpha^2 + \beta^2 + 1 = 0$ olsun. $GF(q)$ üzerinde tanımlı $2n$ uzunluklu C_0 self-dual kodunun bir üreteç matrisi $G_0 = (r_i)$ olsun. $1 \leq i \leq n$ olmak üzere, burada r_i, G_0 matrisinin satırlarıdır. $i = 1, 2$ için, x_1 ve x_2, $GF(q)^{2n}$ uzayına ait $x_1 \cdot x_2 = 0$ ve $x_i \cdot x_i = -1$ koşullarını sağlayan iki vektör olsun. $1 \leq i \leq n$ için $s_i := x_1 \cdot r_i$, $t_i := x_2 \cdot r_i$ ve $y_i := (-s_i, -t_i, -\alpha s_i - \beta t_i, -\beta s_i + \alpha t_i)$ ile tanımlansın. Burada y_i, 4 uzunluklu bir vektör olur. Bu takdirde aşağıda verilen G matrisi $GF(q)$ üzerinde $2n + 4$ uzunluklu self-dual bir C kodunu üretir.

$$G = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & x_1 \\ 0 & 1 & 0 & 0 & x_2 \\ \hline & & y_1 & & r_1 \\ & & \vdots & & \vdots \\ & & y_n & & r_n \end{array}\right]$$

Burada Önerme 6.2.7 de verilen üst-yapı inşa metodu $q \equiv 3(mod4)$ durumuna genişletilmiştir [7]. Bu önermenin ispatı binary self-dual kodlar için üst-yapı inşa metodunda verilen Teorem 6.2.5 e benzer şekilde yapılır.

$q \equiv 1(mod4)$ durumunun yanısıra $GF(2^m)$ üzerinde tanımlı kodlar için de üst-yapı inşa metodu geliştirilmiştir [14].

Önerme 6.2.9. (Çift Karakteristik İçin Üst-Yapı İnşa Metodu)

$GF(2^m)$ üzerinde tanımlı $2n$ uzunluklu C_0 self-dual kodunun bir üreteç matrisi $G_0 = (L|R) = (l_i|r_i)$ olsun. $1 \leq i \leq n$ olmak üzere, burada l_i ve r_i sırasıyla L ve R matrislerinin satırlarıdır. $x = (x_1, x_2, ..., x_n, x_{n+1}, ..., x_{2n})$ $GF(2^m)^{2n}$ uzayına ait $x \cdot x = 1$ koşulunu sağlayan bir vektör olsun. $y_i := (x_1, x_2, ..., x_n, x_{n+1}, ..., x_{2n}) \cdot (l_i|r_i)$ ile tanımlandığında aşağıda verilen G matrisi $GF(2^m)$ üzerinde $2n+2$ uzunluklu self-dual bir C kodunu üretir.

$$G = \left[\begin{array}{cc|cc} 1 & 0 & x_1 \dots x_n & x_{n+1} \dots x_{2n} \\ \hline y_1 & y_1 & & \\ \vdots & \vdots & L & R \\ y_n & y_n & & \end{array}\right]$$

Bu önermenin ispatı da binary self-dual kodlar için üst-yapı inşa metodunda verilen Teorem 6.2.5 in ispatına benzer şekilde yapılır.

6.2.5 Üst-Yapı İnşa Metodu ile Yinelemeli Algoritmanın Karşılaştırılması

Önerme 6.2.10. Binary durumda yinelemeli algoritma, üst-yapı inşa metodunun özel bir hali olur.

İspat . Yinelemeli algoritmanın ikinci basamağında elde edilen $[n+2, \frac{n}{2}]$ tipindeki kodlarının kümesi MC ile gösterilmişti ve üreteç matrisi aşağıdaki biçimde idi.

$$\left[\begin{array}{ccc} 1 & 1 & y' \\ a_1 & a_1 & \\ \vdots & \vdots & E \\ a_{\frac{n}{2}-1} & a_{\frac{n}{2}-1} & \end{array}\right]$$

Bu kümeye ait bir C kodu $C^{\perp}/C$ nin C den farklı bir elemanı ile tamamlanarak $[n+2, \frac{n}{2}+1]$ tipinde bir self-dual kod elde edilmişti. C kodu self-ortogonal bir kod olduğundan, yinelemeli algoritma sonucu elde edilen self-dual kodun bir üreteç matrisi aşağıdaki gibi

yazılabilir.

$$\begin{bmatrix} 1 & 1 & y' \\ a_1 & a_1 & \\ \vdots & \vdots & E \\ a_{\frac{n}{2}-1} & a_{\frac{n}{2}-1} & \\ 1 & 0 & x_1 \dots x_n \end{bmatrix}$$

Burada C yi tamamlayan $C^{\perp}/C$ nin C den farklı bir elemanı $C+y$ ile gösterilirse $(1, 0, x_1, \dots x_n) \in C + y$ olduğu açıktır ve $(x_1 \dots x_n)$ vektörü aşağıdaki koşulları sağlayacak biçimde seçilir. Ayrıca $(x_1 \dots x_n)$ vektörü tekimsi bir vektör olmalıdır. $E = [e_{ij}]$, $(\frac{n}{2}-1)$xn tipinde bir matris olmak üzere;

$$[e]_{ij} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_{\frac{n}{2}-1} \end{bmatrix} \text{ ve } (y'_1, \dots, y'_n) \cdot (x_1, \dots, x_n) = 1$$

şeklindedir. Burada elde edilen lineer denklem sisteminin çözümü $(x_1 \dots x_n)$ vektörünü verir. Yinelemeli algoritma sonucu elde edilen kodun bir üreteç matrisi olan G matrisinin ilk ve son satırları yer değiştirilirse aşağıdaki matris elde edilir.

$$\begin{bmatrix} 1 & 0 & x_1 \dots x_n \\ 1 & 1 & y' \\ a_1 & a_1 & \\ \vdots & \vdots & E \\ a_{\frac{n}{2}-1} & a_{\frac{n}{2}-1} & \end{bmatrix}$$

Burada $\begin{bmatrix} y' \\ E \end{bmatrix} = [L|R]$ biçiminde yazılır ve $n = 2m$ için $(x_1, \dots, x_{\frac{n}{2}}, \dots, x_n) = (x_1, \dots, x_m, \dots, x_{2m})$ ve $y_1 = 1$, $y_2 = a_1$, $y_3 = a_2$, ..., $y_m = a_{\frac{n}{2}-1}$ dönüşümleri uygulanırsa aşağıdaki matris elde edilir.

$$G = \left[\begin{array}{cc|cc} 1 & 0 & x_1 \dots x_m & x_{m+1} \dots x_{2m} \\ \hline y_1 & y_1 & & \\ \vdots & \vdots & L & R \\ y_m & y_m & & \end{array}\right]$$

G' matrisi yukarıdaki biçimde olur. Bu da üst-yapı inşa metodunun binary durumda var olan bir self-dual bir koddan daha uzun bir self-dual kod üretmeye yarayan üreteç matrisidir. Dolayısıyla burada yinelemeli algoritmanın aslında üst-yapı inşa metodunun $GF(2)$ üzerindeki özel bir hali olduğu görülür.

7. BÖLÜM

TABLOLAR ve MATRİSLER

$$G = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 20 & 52 & 4 & 45 & 53 & 21 & 57 & 39 & 58 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16 & 1 & 20 & 52 & 4 & 3 & 53 & 21 & 57 & 39 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 57 & 16 & 1 & 20 & 52 & 22 & 3 & 53 & 21 & 57 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 57 & 16 & 1 & 20 & 4 & 22 & 3 & 53 & 21 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 41 & 9 & 57 & 16 & 1 & 40 & 4 & 22 & 3 & 53 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 8 & 58 & 39 & 57 & 21 & 1 & 16 & 57 & 9 & 41 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 40 & 8 & 58 & 39 & 57 & 20 & 1 & 16 & 57 & 9 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4 & 40 & 8 & 58 & 39 & 52 & 20 & 1 & 16 & 57 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 22 & 4 & 40 & 8 & 58 & 4 & 52 & 20 & 1 & 16 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 22 & 4 & 40 & 8 & 45 & 4 & 52 & 20 & 1
\end{bmatrix}$$

[44] te, GF(61) üzerinde negatif döngüsel yapılanma kullanılarak elde edilen [20,10,10] self-dual kodunun üreteç matrisi yukarıdaki gibidir.

$$G = \begin{bmatrix}
1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 49 \\
33 & 17 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 35 & 20 & 24 & 22 & 17 & 34 & 37 & 8 & 1 \\
17 & 20 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 52 & 35 & 20 & 24 & 22 & 17 & 34 & 37 & 8 \\
1 & 23 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 45 & 52 & 35 & 20 & 24 & 22 & 17 & 34 & 37 \\
32 & 47 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 16 & 45 & 52 & 35 & 20 & 24 & 22 & 17 & 34 \\
7 & 2 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 19 & 16 & 45 & 52 & 35 & 20 & 24 & 22 & 17 \\
41 & 42 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 36 & 19 & 16 & 45 & 52 & 35 & 20 & 24 & 22 \\
29 & 31 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 31 & 36 & 19 & 16 & 45 & 52 & 35 & 20 & 24 \\
40 & 19 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 29 & 31 & 36 & 19 & 16 & 45 & 52 & 35 & 20 \\
30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 33 & 29 & 31 & 36 & 19 & 16 & 45 & 52 & 35
\end{bmatrix}$$

Bu çalışmada GF(53) üzerinde üst-yapı inşa metodu kullanılarak elde edilen [20,10,9] self-dual kodunun üreteç matrisi yukarıdaki gibidir.

$$G = \begin{bmatrix}
1 & 0 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 \\
19 & 26 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 58 & 36 & 31 & 3 & 29 & 7 & 2 & 1 \\
11 & 60 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 60 & 1 & 58 & 36 & 31 & 3 & 29 & 7 & 2 \\
44 & 57 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 59 & 60 & 1 & 58 & 36 & 31 & 3 & 29 & 7 \\
32 & 47 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 54 & 59 & 60 & 1 & 58 & 36 & 31 & 3 & 29 \\
38 & 52 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 32 & 54 & 59 & 60 & 1 & 58 & 36 & 31 & 3 \\
3 & 33 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 58 & 32 & 54 & 59 & 60 & 1 & 58 & 36 & 31 \\
29 & 14 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 30 & 58 & 32 & 54 & 59 & 60 & 1 & 58 & 36 \\
7 & 16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 25 & 30 & 58 & 32 & 54 & 59 & 60 & 1 & 58 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 25 & 30 & 58 & 32 & 54 & 59 & 60 & 1
\end{bmatrix}$$

Bu çalışmada GF(61) üzerinde üst-yapı inşa metodu kullanılarak elde edilen [20,10,9] self-dual kodunun üreteç matrisi yukarıdaki gibidir.

$$G = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 27 & 28 & 40 & 31 & 9 & 52 & 27 & 8 & 59 & 58 & 52 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 50 & 55 & 8 & 24 & 40 & 60 & 40 & 15 & 5 & 29 & 20 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 58 & 53 & 15 & 40 & 6 & 21 & 40 & 9 & 23 & 40 & 55 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 49 & 41 & 42 & 33 & 60 & 30 & 12 & 5 & 7 & 0 & 27 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 60 & 38 & 5 & 60 & 60 & 6 & 7 & 44 & 41 & 58 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 17 & 60 & 4 & 23 & 10 & 34 & 37 & 29 & 33 & 27 & 28 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 24 & 52 & 49 & 31 & 0 & 10 & 22 & 39 & 50 & 56 & 16 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 28 & 31 & 3 & 29 & 18 & 1 & 41 & 6 & 49 & 20 & 8 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 7 & 0 & 40 & 25 & 14 & 52 & 30 & 29 & 14 & 14 & 16 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 39 & 60 & 44 & 42 & 32 & 46 & 38 & 54 & 16 & 3 & 35 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 31 & 54 & 15 & 42 & 28 & 1 & 57 & 54 & 46 & 11 & 47
\end{bmatrix}$$

Bu çalışmada GF(61) üzerinde üst-yapı inşa metodu kullanılarak elde edilen [22,11,10] self-dual kodunun üreteç matrisi yukarıdaki gibidir.

$$G(C_{36}) = \left[\begin{array}{c|cccccccccccccccccc} & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ I_{18} & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

[8] de 38 uzunluklu binary extremal kodların elde edilebilmesi için kullanılan C_{36} kodunun üreteç matrisi yukarıdaki gibidir.

Tablo 7.1 [8] de üstyapı-inşa metodu kullanılarak elde edilen 38 uzunluklu binary extremal kodlardan bazıları

C kodları	$x=(x_1,\ldots, x_{18},1,\ldots,1)$ vektörleri	|PAut(C)|	kullanılan kod
1-6	273274; 472536;143755;715174;702657;672435	1;1;1;1;1;1	C_{36}
7-12	361574;703574;263574;652635;272635;746135	1;1;1;1;1;1	C_{36}
13-18	666135;467135;340757;760335;364335;312735	1;1;1;2;1;1	C_{36}
19-24	433437;346237;077075;431637;346137;773302	1;1;1;1;1;1	C_{36}
25-30	433175;250737;337642;614277;773320;676641	1;1;1;2;1;1	C_{36}
31-36	537641;773222;766341;374741;771660;337322	1;1;1;1;1;1	C_{36}
37-42	727360;770543;374543;732543;375462;175543	1;1;1;1;1;1	C_{36}
43-48	173543;517543;457543;371662;171743;353662	1;1;1;1;1;1	C_{36}
49-54	633662;671562;772223;376223;575223;477223	1;1;1;1;1;1	C_{36}
55-60	766261;656661;555661;535661;533661;674323	1;1;1;2;1;1	C_{36}
61-66	555323;175323;533323;354761;334761;771512	1;1;1;1;1;1	C_{36}
67-72	455761;553463;773150;763312;565263;353712	1;1;1;1;1;2	C_{36}
73-78	155663;135663;527163;467163;664363;523363	1;1;1;1;1;1	C_{36}
79-84	676511;766311;573413;157711;572352;476352	1;1;1;1;1;1	C_{36}
85-90	273352;567213;571613;535613;553613;517613	1;1;1;1;1;2	C_{36}
91-96	437613;137613;371730;672513;771432;565651	1;1;1;1;1;1	C_{36}
97-102	517513;565313;677070;371632;433713;636351	1;1;1;1;1;1	C_{36}
103-108	637270;364751;545751;5257751;267332;472732	1;1;1;1;1;1	C_{36}

8. BÖLÜM

SONUÇ VE ÖNERİLER

Bu çalışmada birçok önemli kod örneğini barındıran self-dual kod ailesi ele alınmış ve genel kod inşa metodları ile özel olarak self-dual kod elde eden bazı metodlar incelenmiştir. Bu inceleme esnasında önemli parametrelere sahip kodlar inşa edilmesini sağlayan yinelemeli algoritma ile bilinen bir metod olan üst-yapı inşa metodu karşılaştırılarak, yinelemeli algoritmanın aslında üst-yapı inşa metodunun özel bir hali olduğu gösterilmiştir.

Altıncı bölümde GF(2) üzerinde gösterilen durum, üst-yapı inşa metodunun çeşitli birçok yapı üzerinde tanımlı olmasından dolayı bütün bu yapılara taşınabilir. Sonuç olarak, bu yolla önemli parametrelere sahip kodlar elde edilebilir.

KAYNAKLAR

1. MacWilliams F.J. and Sloane N.J., The Theory of Error-Correcting Codes, Amsterdam, The Netherlands:North Holland, 1977.

2. Hill R., A First Course in Coding Theory, Oxford Applied Mathematics and Computing Science Series, 1990.

3. Pless V., Introduction to the Theory of Error-Correcting Codes, John Wiley and Sons Publication, 3rd edt, 1998.

4. Huffman W.C. and Pless V., Fundamentals of Error-Correcting Codes, Cambridge, Cambridge University Press, 2003.

5. Rains E. M. and Sloane N. J., Self Dual Codes in Handbook of Coding Theory, Pless V. and Huffman W., Eds. Amsterdam, The Netherlands : Elseiver , 1998.

6. Kim J.-L., Gulliver T.A., Lee Y., New MDS or Near MDS Self-Dual Codes, IEEE Transacitons on Information Theory, vol. 34 , 4354-4360, 2008.

7. Kim J.-L., Lee Y., Self-Dual Codes Using the Building-Up Construction, 2400-2402, ISIT June 28-July 3, Seoul, Korea, 2009.

8. Kim J.-L., New Extremal Self-Dual Codes of Lengths 36,38 and 58, IEEE Transactions on Information Theory, vol. 47, 386-393, 2001.

9. Kim J.-L., Lee Y., Euclidean and Hermitian Self-Dual Codes Over Large Finite Fields, Journal of Combinatorial Theory Ser. A, vol. 105, 407-422, 2008.

10. Huffman W.C., On the Classification and Enumeration of Self-Dual Codes, Finite Fields and Applications, vol. 11, 451-490, 2005.

11. Melchor C.A. and Gaborit P., On the Classification of Extremal [36,18,8] Binary Self-Dual Codes, IEEE Transactions on Information Theory, vol. 54, 4743-4750, 2008.

12. Harada M., MDS Self-Dual Codes of Length 36 and 38, International Journal of Information and Coding Theory, vol. 1, 208-213, 2010.

13. Harada M., The Existence of a Self-Dual [70,35,12] Code and Formally Self-Dual Codes, Finite Fields and Their Applications, vol. 3, 131-139, 1997.

14. Kim J.-L., Harada M., Gulliver A. T., Construction of New Extremal Self-Dual Codes, Discrete Mathematics, vol.263, 81-91, 2003.

15. Shannon C., A Mathematical Theory of Communication, Bell System Tech. J., vol. 27, 379-423, 1948.

16. Pless V., On the Uniqueness of the Golay Codes, J. Combin. Theory, vol. 5, 215-228, 1968.

17. Pless V., A Classification of Self-Ortogonal Codes Over GF(2), Discrete Math., vol. 3, 209-244, 1972.

18. Pless V., The Children of the [32,16] Doubly-even Codes, IEEE Trans. Inform. Theory, vol. 24, 738-746, 1978.

19. Conway J. H. and Pless V., On the Enumeration of Self-Dual Codes, J. Combin. Theory Ser. A, vol. 28, 26-53, 1980.

20. Sloane N. J. A., Pless V. and Conway J. H., The Binary Self-Dual Codes of Length up to 32 : A Revised Enumeration, J. Combin. Theory Ser. A, vol. 60, 183-195, 1992.

21. Betsumiya K., Harada M. and Munemasa A., A Complete Classification of Doubly-even Self-Dual Codes of Length 40, preprint, 2011.

22. Cannon J. and Playoust C., An Introduction to Magma, University of Sydney, Sydney, Australia, 1994.

23. Best M., Brouwer A., MacWilliams F. J., Odlyzko A., Sloane N. J., Bounds for Binary Codes of Length Less Than 25, IEEE Trans. Inform. Theory, vol. 24, 81-93, 1979.

24. Sloane N. J., A Survey of Constructive Coding Theory and A Table of Binary Codes of Highest Known Rate, Discrete Math. vol. 3, 265-294, 1972.

25. Doughtery S., Kim J.-L., Kulosman H., Liu H., Self-Dual Codes Over Frobenius Rings, Finite Fields and Its Appl., vol. 16, 14-26, 2010.

26. Bachoc C., Applications of Coding Theory to the Construction of Modular Lattices, J. Combin. Theory, vol. 78A, 92-119, 1997.

27. Bannai E., Harada M., Ibukiyama T., Munemasa A., Oura M., Type II Codes Over $F_2 + uF_2$ and Applications to Hermitian Modular Forms, Abh. Math. Sem. Univ. Hamburg, 73, 13-42, 2003.

28. Calderbank A. R., Sloane N. J., Double Circulant Codes Over Z_4 and Even Unimodular Lattices, J. Algebraic Combin., vol. 6, 119-131, 1997.

29. Conway J. H. and Sloane N. J., Sphere Packings, Lattices and Groups, Springer, New York, 1999.

30. Harada M., Sole P., Gaborit P., Self-Dual Codes Over Z_4 and Unimodular Lattices: A Survey, Algebras and Combinatorics, An International Congress, ICAC'97, Hong Kong, 1997.

31. Tonchev V., Codes and Designs in: Pless V., Huffman W. C., Handbook of Coding Theory, Elsevier, Amsterdam, 1229-1267, 1998.

32. MacWilliams F.J., Odlyzko A. M., Sloane N. J., Ward H. N., Self-Dual Codes Over GF(4), J. Combin. Theory, vol. 25A, 474-488, 1978.

33. Rains E. M., Shadow Bounds for Self-Dual Codes, IEEE Trans. Inform. Theory, vol. 44, 134-139, 1998.

34. Mallows C. L. and Sloane N. J., An Upper Bound for Self-Dual Codes, Inform. and Control, vol. 22, 188-200, 1973.

35. Hamming R., Error Detecting and Error Correcting Codes, The Bell System Technical Journal, vol. 29, 147-160, 1950.

36. Golay M. J. E., Notes on Digital Coding, Proc. I.R.E., vol. 37, 657, 1949.

37. Ward H. N., Divisible Codes, Archiv. Mat. (BASEL), vol. 36, 485-494, 1981.

38. Ward H. N., Divisibility of Codes Meeting the Griesmer Bound, J. Combin. Theory, vol. 83A, 79-93, 1998.

39. Assmus E. I., Mattson H. F., Turyn Jr. and R.J., Research to Develop the Algebraic Theory of Codes, Report AFCRL 67035, Air Force Cambridge Res. Labs., Bedford, MA, 1967.

40. Gleason A. M., Weight Polynomials of Self-dual Codes and the MacWilliams Identities, Congrès International de Mathèmatiques (Nice,1970), Gauthèrs-Villars, Paris, vol. 3, 211-215, 1971.

41. Harada M., A Complete Classification of Doubly-even Self-dual Codes of Length 40, submitted to EJC, 31/05/2011.

42. Harada M.,Munemasa A., Classification of Self-dual Codes of Length 36, submitted to ACM, 02/06/2011.

43. Pless V., Lam C. W. H., Thereis no (24,12,10) Self-dual Quaternary Codes, IEEE Trans. on Inform. Theory, vol. 36, 1153-1156, 1990.

44. Harada M., Kharaghani H., Orthogonal Designs and MDS Self-dual Codes, Australasian J. Combin., vol. 35, 57-67, 2006.

45. Çallıalp F., Örneklerle Soyut Cebir, Birsen Yayınevi, İstanbul, 2001.

46. Sabuncuoğlu A., Lineer Cebir, 3. baskı, Nobel Yayınevi, Ankara, 2008.

ÖZGEÇMİŞ

Hatice TOPCU 1986 yılında Afyonkarahisar ilinin Şuhut ilçesinde doğdu. İlk öğrenimini Şuhut Zaferyolu İlköğretim okulunda, orta öğrenimini Afyonkarahisar Anadolu İmam Hatip Lisesinde tamamladı. 2000 yılında Konya Meram Fen Lisesi'ni kazandı. Aynı yıl Afyonkarahisar Süleyman Demirel Fen Lisesine geçiş yaptı ve lise öğrenimini burada tamamladı. 2003 yılında Hacettepe Üniversitesi Matematik Bölümünü kazandı ve 2007 yılında buradan mezun oldu. 2007-2008 yılları arasında Afyon Kocatepe Üniversitesi Eğitim Fakültesinde Ortaöğretim Matematik Eğitimi alanında tezsiz yüksek lisans yaptı. Aynı zamanda 2007-2009 yılları arasında Afyonkarahisar Final Dergisi Dersanelerinde, Afyonkarahisar Anadolu İmam Hatip Lisesinde ve Afyonkarahisar Kocatepe Anadolu Lisesinde Matematik ve Geometri öğretmenliği yaptı. 2009 yılında Nevşehir Üniversitesi Matematik Bölümünde araştırma görevlisi olarak göreve başladı ve aynı yıl Nevşehir Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalında yüksek lisansa başladı. Evli olup halen Nevşehir Üniversitesi Matematik Bölümünde görevine devam etmektedir.

Adres: Nevşehir Üniversitesi Fen Edebiyat Fakültesi
Matematik Bölümü Kat:2 Oda:13

Telefon: 03842281000-1325

e-posta: hatice.kamit@nevsehir.edu.tr