



# Düzce Üniversitesi Bilim ve Teknoloji Dergisi

*Derleme Makalesi*

## Haberleşme Sistemlerinde Turbo Kodlama ve Turbo İlkesinin Bazı Pratik Uygulamaları

Selman KULAÇ<sup>a,\*</sup>, Murat Hüsni SAZLI<sup>b</sup>

<sup>a</sup> Elektrik-Elektronik Mühendisliği Bölümü, Mühendislik Fakültesi, Düzce Üniversitesi, Düzce, TÜRKİYE

<sup>b</sup> Elektrik-Elektronik Mühendisliği Bölümü, Mühendislik Fakültesi, Ankara Üniversitesi, Ankara, TÜRKİYE

\* Sorumlu yazarın e-posta adresi: selmankulac@duzce.edu.tr

### ÖZET

Bu çalışmada, bir kanal kodlama türü olan turbo kodlama incelenmiş ve turbo kod çözme algoritmasından hareketle türetilmiş olan turbo ilkesi ele alınmıştır. Turbo mantığına dayalı turbo ilkesi pratik hayatta ciddi önem kazanmıştır. Bu nedenle turbo ilkesinin yer aldığı bazı pratik uygulamalar hakkında da bilgiler verilmiştir.

**Anahtar Kelimeler:** Turbo kodlama, Turbo kod çözme, Kanal kodlama, Turbo ilkesi

## Turbo Coding in Communication Systems and Some Practical Applications of Turbo Principle

### ABSTRACT

In this study, turbo coding which is a kind of channel coding is analyzed and turbo principle that was derived from turbo decoding algorithm is described. Turbo principle based turbo reason has obtained great importance in practical life. Therefore, information about some practical applications of turbo principle is given.

**Keywords:** Turbo coding, Turbo decoding, Channel coding, Turbo principle

### I. GİRİŞ

**G**ÜNÜMÜZE kadar haberleşme sistemlerinin vazgeçilmez bir unsuru olan kanal kodlama konusu üzerinde çok sayıda çalışmalar yapılmış olup halen konu üzerinde araştırmacılar yoğun bir şekilde çalışmalar yapmaya devam etmektedirler. Özellikle de telsiz haberleşme sistemlerinde son derece önemli bir yer teşkil eden kanal kodlama işleminin amacı, kanalda iletilen verinin maruz kaldığı bir takım bozucu etkilere karşı koymak ve verinin en az hata ile alıcıya ulaştırılmasını

sağlamaktır. Bu amaca yönelik olarak çok çeşitli kanal kodlama teknikleri ve algoritmaları geliştirilmiştir.

Kanal kodlama ile verinin korunması amaçlanırken verinin çoğaltılması gerektiği ve bunun sonucu olarak haberleşme ortamında sınırlı olarak mevcut olan bant genişliğinin israf edilmemesi gerektiği de dikkate alınmalıdır. Bu amaçla kanal kodlama öncesinde yapılan kaynak kodlama işlemi ile verinin sıkıştırılması da gereklidir. Bu sıkıştırma ile verinin içerdiği esas bilgi kaybedilmeksizin artıkların atılması sağlanmalıdır. Bu amaca yönelik olarak, ses, video ve veri iletimi için çok çeşitli kaynak kodlayıcılar geliştirilmiş ve veri sıkıştırma alanında da ciddi başarımlar elde edilmiştir.

Kanal kodlama konusunda ise ilk akla gelen isim C.E. Shannon'dur. Shannon'un 1948'de yayınladığı "Haberleşmenin Matematiksel Teorisi" isimli makalesi ile Hata Kontrol Kodlaması (Error Control Coding) alanı doğmuştur [1]. Shannon, kanal kapasitesini tarif ederek ve bu kanal kapasitesinin altındaki iletim hızlarında çok düşük bit hata oranlarında (BER) verinin karşı tarafa iletilebileceğinden söz etmiştir. Shannon, veri iletim hızı için teorik bir sınır tanımlamış ve Shannon'dan günümüze kadar bu sınıra yaklaşma çabaları sürmüştür.

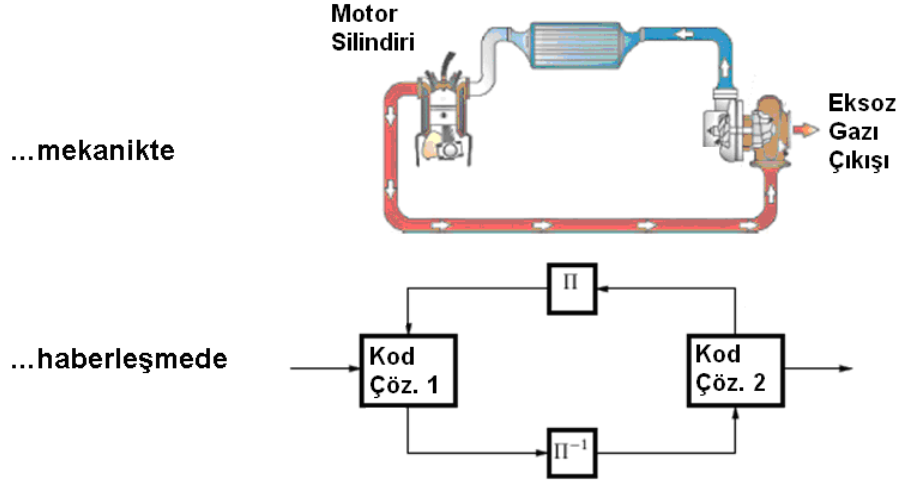
Shannon'dan sonra çok sayıda etkin kodlama ve kod çözme teknikleri bulunmuş olmasına rağmen, 90'lı yılların başlarında turbo kodların bulunmasına kadar, kanal kapasite sınırına ciddi bir düzeyde yaklaşan kodlama ve kod çözme yöntemleri bulunamamıştır. Başka bir ifade ile, turbo kodlar kodlama kazancı ile kanal kapasite sınırı arasındaki önemli boşluğu uygun kodlama ve kod çözme yöntemlerinin kullanımı sayesinde kapatmıştır denilebilir.

1993 yılında Berrou, Glavieux ve Thitimajshima [2,3] tarafından turbo kodların bulunmasından sonra, sayısal haberleşme ve veri kayıt sistemleri alanında performans analizi, tasarım, gerçekleştirme ve uygulamaya yönelik olarak birçok araştırma yapılmıştır.

Temelde turbo kodlama kod çözme mimarisi, aralarında serpiştiricilerin de yer aldığı özyineli sistematik katlamalı kodlayıcıların (Recursive Systematic Convolutional Encoder, RSC) paralel sıralanması (concatenation), SISO (soft input, soft output – yumuşak giriş, yumuşak çıkış) kod çözücüler arasında döngülü (iterative) bilgi değişimine dayalı kod çözme yaklaşımı sunan bir yapı içermektedir. Bu yapı, zamanla başka alanlarda da uygulanmış ve bir 'ilke' olarak kabul edilerek "turbo ilkesi" olarak literatüre girmiştir [4,5].

## II. TURBO KODLAR

Turbo kodların bulunması ile 1948'de Shannon'un belirlemiş olduğu sınıra daha önce hiç olmadığı kadar yaklaşarak, hata düzelten kodlar alanında çok büyük bir gelişme sağlanmıştır. Turbo kodların hata başarımı açısından Shannon sınırının hemen hemen bitişiğinde bir başarımla sağladığı yapılan çalışmalarda gösterilmiştir. Bu döngülü kodlayıcıda katlamalı veya blok kodların paralel sıralanması (concatenation) ve turbo motor ilkesine benzer bir kod çözme tekniğinin kullanılması ile mümkün olmuştur.



**Şekil 1.** Mekanikte turbo motor ve haberleşmede turbo kod çözücü

Şekil 1’de basit bir turbo motor ile basit bir turbo kod çözücü yer almaktadır [6]. Temelde turbo motorlar tüm sistemin başarımını artırmak için bir geri besleme döngüsü kullanırlar. Turbo kod çözücüler de benzer şekilde çalışırlar. Turbo kod çözücünün bileşen kod çözücüleri (component decoders) arasında geri besleme ile döngü (iteration) sağlanarak başarım artırılır. Deneysel sonuçlar göstermektedir ki, yeterince büyük blok uzunluğu ile döngülü turbo-kod çözme, sinyal/gürültü oranı açısından en çok-olabilirlik (maximum-likelihood) kod çözme eşik değerine yaklaşımda en iyi (optimal) başarımı sağlayabilmektedir [4].

Sıralı kodların kod çözülmesinin önceden kullanılmış olmasına rağmen, turbo kodların bulunması ile birlikte SISO algoritmaları daha önce hiç olmadığı kadar popülerlik kazanmıştır. SISO algoritmalarının iki ana kategorisi vardır: Sembol (bit) hata olasılığını en küçükleyen MAP (maximum a posteriori) algoritması ve sözcük (dizi) hata olasılığını en küçükleyen Yumuşak Çıkış Viterbi Algoritmasıdır (SOVA).

1967’de A. J. Viterbi [7] yayınladığı makalesinde katlamalı kodların kod çözümünde kullanılan bir algoritmayı anlatmıştır. Bu algoritma, bulunduğu yıllarda, optimumluğu (sözcük hatasını en küçükleştirme bakımından optimum) ve düşük hesap karmaşıklığı sayesinde kısa sürede yaygın bir şekilde kullanılmaya başlanmıştır. Ancak, bu algoritma ile kod çözülen dizideki sözcük (word) hatası en küçükleştirilirken, bu algoritmanın “sert çıkışlı (hard output)” bir algoritma olması ve kod çözülmüş bitlerin her birinin ayrı ayrı kestirimindeki doğruluk için “yumuşak bilgi (soft information)” veya “güvenilirlik değerleri (reliability values)” sunmaması, “yumuşak çıkış” elde edilen algoritmalara yönelim sağlamıştır.

1974’te Bahl, Cocke, Jelinek ve Raviv [8] bir gürültülü kesikli hafızasız kanal (DMC, discrete memoryless channel) üzerinden gözlemlenen Markov kaynağın durumlarının ve geçişlerinin, sonsal olasılıklarını (a posteriori probabilities) özyineli olarak tahmin eden bir algoritma sunmuşlardır. Sözü edilen makalede yazarlar, bu problemin özel bir durumu olarak doğrusal kodların (katlamalı ve blok) kod çözümünü ve bu algoritmanın doğrusal katlamalı ve doğrusal blok kodların kod çözümünde kullanılabilir bir algoritma olduğunu göstermişlerdir. BCJR algoritması katlamalı ve blok kodlara doğrudan uygulanabilir olmasına rağmen, artan karmaşıklık nedeni ile Viterbi algoritmasına göre

cazip bir algoritma olmamıştır. Bunun nedeni “döngülü kod çözmenin (iterative decoding)” veya “önyükleyici kod çözmenin (bootstrap decoding)” o tarihlerde öneminin yeterince anlaşılmasını sağlamamış olmasıdır. Sonraki zamanlarda, “sıralı kodlara”(concatenated codes) olan ilginin artması ve “turbo kodların” gelişi, BCJR algoritmasını daha popüler yapmış ve turbo kod çözücülerin bileşen kod çözücülerini ve diğer sıralı kodların kod çözümünde döngülü kod çözme şemaları için en fazla kullanılan algoritma olmasını sağlamıştır. Bu da BCJR algoritmasının optimalliğine ve orijinal algoritmanın değiştirilmesine gerek olmaksızın yumuşak çıkışlar üretme yeteneğine atfedilebilir. Bu algoritma MAP algoritmasının verimli bir gerçekleştirilmesi olup, mucitlerinin anısına BCJR algoritması olarak bilinmektedir.

Belirtildiği üzere, Viterbi algoritması yumuşak çıkışlar üretme yeteneğine sahip değildir ve bu yüzden orijinal hali ile döngülü kod çözücülerde kullanıma uygun değildir. Hagenauer ve Hoeher daha az karmaşık ama optimal altı (suboptimal) olan SOVA (Yumuşak Çıkışlı Viterbi Algoritması) olarak isimlendirilen bir alternatif önermişlerdir [9]. SOVA'nın azaltılmış hesap yüküne karşılık, BCJR algoritması ve onların optimal altı türevleri turbo kod çözücülerde ve diğer döngülü kod çözücülerde daha fazla tercih edilmekte ve kullanılmaktadır. Şu hususun da belirtilmesi gerekir ki, her iki algoritma (BCJR ve Viterbi) da uygun şekilde formüle edildiklerinde eşdeğer oldukları gösterilmiştir [10].

### III. TURBO KODLAYICI

Bir turbo kodlayıcı iki (veya daha fazla) sistematik kodun paralel sıralanmasından oluşur. Her bir kodlayıcı/kod çözücü, “bileşen kodlayıcı/kod çözücü (component encoder/decoder)” olarak adlandırılır. Bir turbo kodlayıcıda iki bileşen kodlayıcı varsa, “iki boyutlu turbo kod” olarak isimlendirilir.

Orijinal olarak turbo kodlar, RSC (Özyineli Sistematik Katlamalı, Recursive Systematic Convolutional) kodlayıcılar kullanılmaktadırlar. Normalde, yüksek SNR’larda (Signal-to-Noise Ratio, İşaret-Gürültü Oranı) aynı kod belleğine sahip sistematik olmayan katlamalı (NSC, Non Systematic Convolutional) kod ile klasik sistematik koddan daha düşük BER (bit hata oranı) elde edilebilmektedir. Ancak RSC kodlar, herhangi bir SNR’da yüksek kod oranları için en iyi NSC’den daha iyi başarımlar sunabilmektedirler [2].

#### *A. ÖZYİNELİ SİSTEMATİK KATLAMALI KODLAR (RSC)*

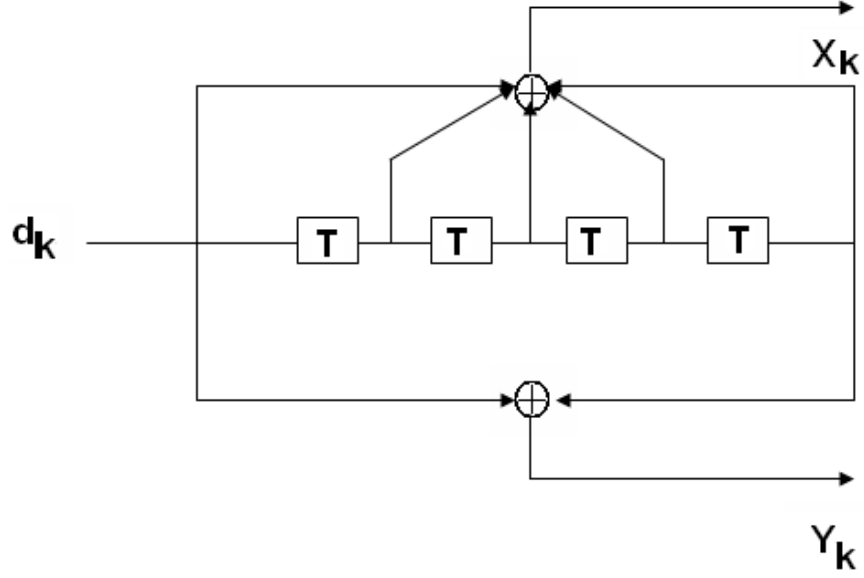
Bu alt bölümde, RSC kodları ele almadan önce ilk olarak klasik sistematik olmayan katlamalı (NSC) kodlayıcılar anlatılacaktır. Kod oranı  $R=1/2$ , kod belleği  $v=K-1$  ve kısıt uzunluğu (constraint length)  $K$  olan bir katlamalı kodlayıcı için; kodlayıcıya  $k$  anında giren bit  $d_k$  ve uygun kod sözcüğü ikili çift  $(X_k, Y_k)$  ile gösterilen  $C_k$  olsun.  $C_k$ 'nin bileşenleri  $X_k$  ve  $Y_k$

$$X_k = \sum_{i=0}^{K-1} g_{1i} d_{k-i} \quad \text{kip.2(mod.2)} \quad g_{1i} = 0,1 \quad (1)$$

$$Y_k = \sum_{i=0}^{K-1} g_{2i} d_{k-i} \quad \text{kip.2(mod.2)} \quad g_{2i} = 0,1 \quad (2)$$

eşitlikleri ile ifade edilir.

Burada  $G_1: \{g_{1i}\}$  ve  $G_2: \{g_{2i}\}$ , genelde sekizlik sistemde ifade edilen iki kod üreticidir. Şekil 2’de hafızası  $v=4$  ve kod üreteçleri  $G_1=37$  ve  $G_2=21$  olan bir NSC kodlayıcı gösterilmiştir [11].



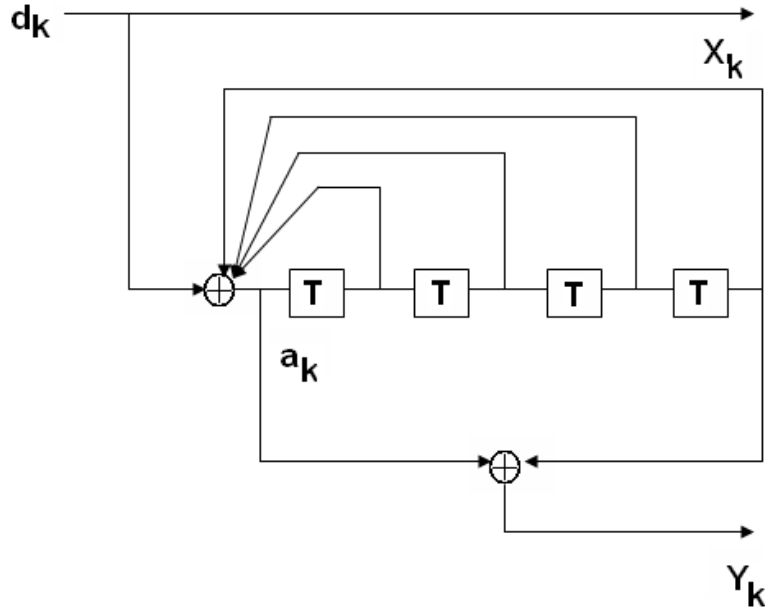
Şekil 2. Klasik sistematik olmayan katlamalı (NSC) kodlayıcı[11]

Şekil 3’te ise şekil 2’de gösterilen NSC kodlayıcıdan elde edilen geri besleme döngüsü kullanarak NSC’den elde edilen RSC kodlayıcı yer almaktadır. Burada dikkat edilmesi gereken husus kodu sistematik yapmak için çıkışlardan birinin (bu durumda  $X_k$ ) bilgi biti  $d_k$ ’ya eşit olarak ayarlanması gerektiğidir. Ayrıca, kaydırmalı kaydedici-belleğin (shift register-memory) girişi bilgi biti  $d_k$  değil de, yeni bir değişken olan  $a_k$  olarak alınmıştır. Bu da şekil 3’teki kodlayıcı yapısındaki geri beslemeden kaynaklanmaktadır. Burada  $X_k=d_k$  olduğundan  $Y_k$  çıkışı,  $d_k$ ’nın yerini alan  $a_k$  olacak şekilde 2 numaralı eşitlik ile hesaplanır. Ayrıca  $\gamma, X_k = d_k$  ise  $g_{1i}$ ’ye,  $Y_k = d_k$  ise  $g_{2i}$ ’ye eşit olacak şekilde  $a_k$  özyineli olarak 3 numaralı eşitlik ile hesaplanır.

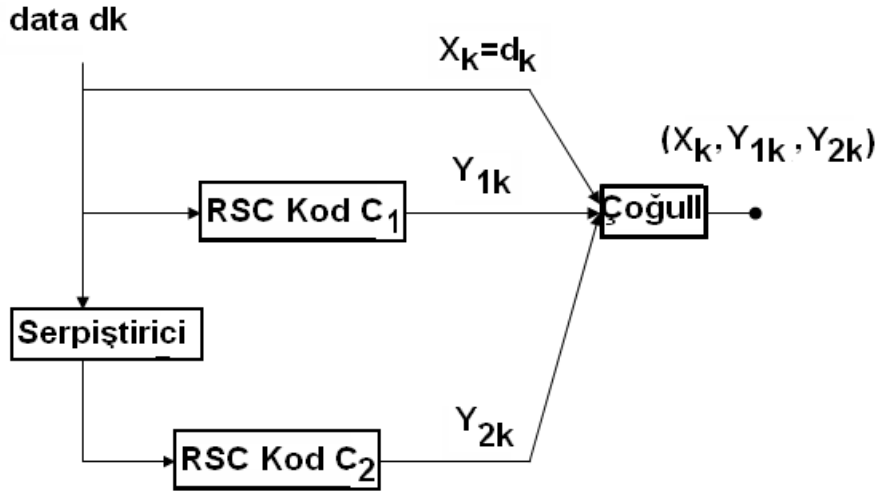
$$a_k = d_k + \sum_{i=1}^{K-1} \gamma_i a_{k-i} \quad \text{kip}.2(\text{mod}.2) \quad (3)$$

[2] ve [3]’te  $a_k, d_k$  ile aynı istatistiksel özelliğe sahip olduğunu göstermiştir. Bunun sonucu olarak da, RSC ve NSC kodların kafes yapısı aynı olduğu ve bu kodların aynı serbest uzaklıklara (free distance,  $d_f$ ) sahip oldukları bilinmektedir. Ancak RSC ve NSC kodlar için aynı giriş dizisi  $\{d_k\}$ ’ya karşılık olarak aynı çıkış dizileri  $\{X_k\}$  ve  $\{Y_k\}$ ’nın elde edilmediği de literatürde belirtilmiştir. İki kod arasındaki temel fark bundan ibarettir.

Kafes yapısı katlamalı kodların ayırıcı bir özelliği olup hemen hemen her haberleşme kitabında yer almaktadır (örneğin [12]).



Şekil 3. Özyineli sistematik katlamalı (RSC) kodlayıcı (oran 1/2) [11]



Şekil 4. 1/3 oranlı turbo kodlayıcı [11]

### B. RSC KODLARIN PARALEL SIRALANMASI

Şekil 4'te  $\frac{1}{3}$  oranlı turbo kodlayıcı gösterilmiştir. Bu şekle göre  $(d_1, d_N)$  veri dizisi bloklarla kodlanmaktadır. Blok uzunluğu, serpiştirici uzunluğu tarafından belirlenir. Orijinal veri dizisi birinci bileşen kodlayıcı tarafından kodlanır ve  $(Y_{11}, Y_{1N})$  eşlik bit dizisi üretilir. Serpiştirici, orijinal veri dizisini karıştırır ve karıştırılmış veri dizisi, ikinci eşlik bit dizisini  $(Y_{21}, Y_{2N})$  üreten ikinci bileşen kodlayıcı tarafından kodlanır. Serpiştirme tasarlanmış bir kurala göre olabildiği gibi veya tamamen rasgele de olabilir. Burada önemle üzerinde durulması gereken konu ise serpiştirici uzunluğudur (boyutudur). Genellikle daha uzun bir serpiştirici daha iyi başarımlar sağlar [11].

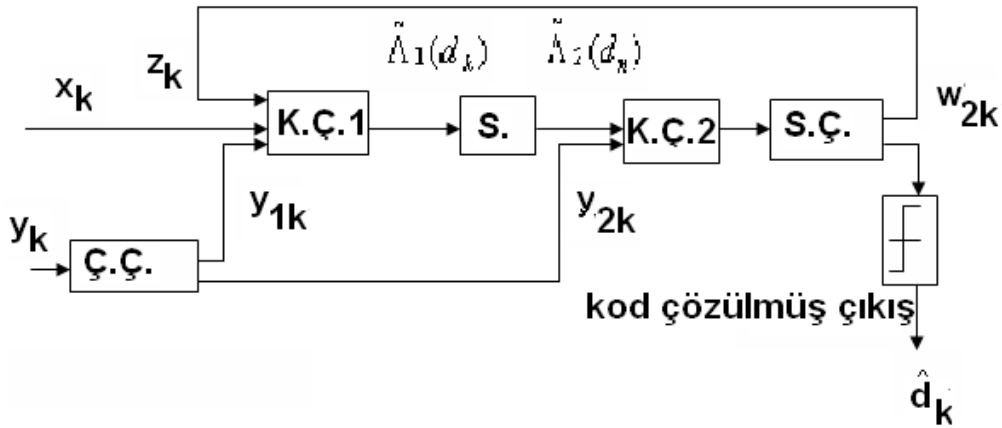
Turbo kod çözücündeki her bir bileşen kod çözücü, aynı  $d_k$  bitinin kod çözülmesini benzer şekilde gerçekleştirir. Ancak serpiştirme nedeni ile orijinal veri dizisinin, hem orijinal hali hem de

karıştırılmış hali için iki farklı kod çözümü gerçekleştirilir. Böylelikle döngülü kod çözüm planında her bir bileşen kod çözücünün katkısı ile oluşturulan kararı iyileştirmek için, bileşen kod çözümler birbirlerine dizi içinde yer alan her bir bitin kod çözümünün doğruluğu hakkında ilave bir bilgi sağlarlar. Bu nedenle arttırılmış blok (serpiştirici) uzunluğu, orijinal veri dizisindeki herhangi bir bitin konumu ile aynı bitin karıştırılmış dizideki konumu arasında daha iyi bir ayırma sağlayabilir. Bunun nedeni, artan blok uzunluğu sayesinde her bir bileşen kodlayıcı tarafından üretilen eşlik bitlerinin kod çözücünde algılandığında “daha az ilintili (weakly correlated)” hale getirilmesidir. Bu da kodlamanın sağladığı bir “çeşitlilik (diversity)” etkisi olarak düşünülebilir.

Turbo kodlayıcılarla ilgili bir diğer konu ise “kod delme (code puncturing)”dir. Her iki bileşen kodlayıcının eşlik bitleri bilgi biti ile çoğullanarak iletilebilir. Bu durumda kod oranı  $\frac{1}{3}$  olur. Alternatif olarak, her bir bileşen kodlayıcının eşlik bitleri, önceden belirlenen bir düzene göre bilgi biti ile çoğullanabilir. Bu, “kod delme” olarak adlandırılır. Sonuçta oluşan kod ise “delinmiş kod (punctured code)” olarak adlandırılır. Örneğin, kod sözcüğü bir sinyal aralığında  $(X_k, Y_{1k})$  olarak gönderilirken, diğer aralıkta  $(X_{k+1}, Y_{2(k+1)})$  olarak gönderilir. Böylelikle, her bir bilgi biti için sadece bir eşlik biti iletilir ve kod oranı  $\frac{1}{2}$ 'ye düşürülmüş olur. Kod delme işlemi ile istenilen herhangi bir kod oranı elde edilebilir [13].

#### IV. TURBO KOD ÇÖZÜCÜ

Şekil 5’te bölüm 2’de bahsedilen turbo kod çözücüne ait blok şema verilmiştir.



Şekil 5. Turbo kod çözücü[11]

Turbo kod çözücünün birinci yinelemesi (iteration),  $z_k$ 'lar sıfır alınarak başlatılır.  $x_k$ 'lar ile bileşen kod çözümler için uygun eşlik bitlerini içeren  $y_k$ 'lar çoğullanırlar. Eşlik bitleri turbo kodlayıcıda kullanılan delme (varsa) düzenine göre uygun kod çözümlere sağlanır. Delme işlemi yapılmamışsa her bir sinyal aralığında  $y_{1k}$  kod çözümler 1'e,  $y_{2k}$  kod çözümler 2'ye uygun bir şekilde sağlanır [11].

Kod çözümler 1, BCJR algoritması ile blokta yer alan her bir  $d_k$  biti için karşılık gelen  $x_k$  ve  $y_{1k}$ 'ları kullanarak bu  $d_k$  bitlerinin “yumuşak (soft)” kestirimlerini üretir. Bu yumuşak kestirimlerden bilginin “harici bilgi (extrinsic information)” olarak bilinen ilgili parçaları çıkarılır ve serpiştirilerek (kodlayıcıda kullanılan serpiştirici ile aynı serpiştiricinin kullanılması şartı ile) kod çözümler 2'ye

aktarılır. Kod çözücü 2, bloktaki tüm  $d_k$  bitlerinin daha iyi kestirimlerini elde etmek amacı ile  $y_{2k}$  ile bu harici bilgileri kullanır. Kod çözücü 2 tarafından üretilen harici bilgi, serpiştirme çözme işleminden sonra bloktaki tüm  $d_k$  bitleri üzerindeki kestirimleri iyileştirmek amacı ile, bir sonraki yineleme için kod çözücü 1'e aktarılır. Tatminkar bir BER elde edilene kadar döngülü işleme devam edilir. Genellikle 20 yinelemenin altında,  $10^{-5}$  düzeyinde bir BER elde edilir.

Önceden belirlenmiş yineleme sayısına ulaşıldığında, kod çözücü 2'nin yumuşak çıkışı (serpiştirme çözülmüş), vericiden iletilmiş olan bilgi bitlerinin kestirimlerini elde etmek için sıfır eşik değerli bir sert sınırlayıcıya (hard limiter) aktarılarak bit dizisi olarak çıkartılır.

## V. TURBO İLKESİ

J-SAC (IEEE Journal on Selected Areas in Communications) dergisinin Şubat 1998'te yayınlanan "(Sıralı Kodlama Teknikleri ve Döngülü Kod Çözme: Kanal Kapasitesine Yelken Açma)" isimli özel sayısı, kodlama teorisinin yeni bir konusu olan turbo kodlar ile ilgili çarpıcı gelişmeleri kapsamlı bir şekilde içermektedir [4].

Bu özel sayının yayıncıya ait ön sözünde turbo kod çözme algoritmasının altı çizilerek, haberleşme alıcılarındaki farklı bloklar arasındaki döngülü yumuşak bilgi alışverişi stratejisini bir turbo motordaki kompresör ve türbin arasındaki alışverişe benzeterek, bir anahtar kelime tanımlanmaya çalışılmış ve bu da "turbo ilkesi" olarak vurgulanmıştır. Turbo ilkesi, önceden de sezinlendiği üzere, haberleşme alanındaki çok çeşitli problemlerde güçlü bir araç olarak yaygın bir şekilde kullanılmaya başlamıştır. Bunlardan bazıları, uzay-zaman (space-time) kodları da içeren yüksek dereceden kodlanmış-modülasyon şemalarının kod çözme uygulamaları (applications to decoding of higher order coded-modulation schemes), birleşik kaynak-kanal kod çözme (joint source-channel decoding), birleşik kanal tahmini/denkleştirme ve kod çözme (joint channel estimation/equalization and decoding), çok kullanıcı ve yayılı-izge haberleşme sistemlerinde algılama ve kod çözme (detection and decoding of multiuser and spread-spectrum communication systems) olarak sıralanabilir. Dahası da, turbo ilkesinin haberleşme teorisi, enformasyon teorisi, dinamik sistemler, grafik teorisi ve yapay zeka disiplinleri gibi alanlarla da önemli bağlar kurduğu bilinmektedir.

Turbo ilkesinin çıkış alanı olan turbo kod tasarımı alanında, "turbo-benzeri (turbo-like)" seri ve karma (seri/paralel) kodlama şemaları sıralanması mimarileri üzerinde çalışmalar yapılmıştır. Özellikle bu tasarımlarda geniş blok uzunlukları için kapasite sınırına en çok yaklaşma sağlandığı önemle belirtilmiştir.

Turbo ilkesi içerisinde yer alan bir önemli kod sınıfı ise, Gallager tarafından 1963'de çıkarılan LDPC (Düşük Yoğunluklu Eşlik Denetim, Low Density Parity Check) kodları olup, özellikle geliştirilmiş hali ile epeyce dikkatleri üzerinde toplamış bir kod sınıfı olmuştur [14]. Döngülü mesaj-geçme (message-passing) kod çözme algoritması ile, bu kodlama teknikleri bazen orijinal turbo kodlarla karşılaştırılabilir bir duruma gelmiştir.

### *A. TURBO İLKESİNİN PRATİK UYGULAMALARI*

Turbo ilkesinin teoriden pratiğe geçirilmesinde ciddi adımlar atılmıştır. Güçlü turbo kodların geliştirilmesinden en fazla yararlanan uygulama alanlarından biri de derin uzay haberleşmesi alanıdır. Bu yarar iki şekilde sağlanmıştır. Birincisi, Cassini ve Pathfinder görevleri için kullanılan en



karmaşık sıralı kodlardaki başarıyı geliştirme ve daha önemlisi de kod çözme karmaşıklığında önemli bir azalmadır. Bu da 16-durumlu bileşen kod çiftinin kullanılması ile bir paralel-sıralı kod ailesinin seçimine ve NASA/JPL ve Avrupa Uzay Kuruluşu (ESA, European Space Agency)'nın kod tasarımına dayalı olarak, bir yeni Uzay Data Standartları için Danışma Kurulu (CCSDS, Consultative Committee for Space Data Standards) standardı oluşturulmasına imkan sağlamıştır. Bu kodlar, NASA görevlerinde ve NASA'nın diğer kuruluşlarla gerçekleştireceği ortak görevlerde Derin Uzay Ağı (DSN, Deep Space Network) için kullanılmaya devam edilmiştir. Ayrıca, turbo kodların, oldukça düşük sinyal/gürültü oranlarında alıcı eşzamanlama döngüsündeki işlemini esnasında oluşan problemi aşmak için yeni alıcı/kod çözücü şemaları geliştirilmeye çalışılmıştır. Turbo ilkesi uygulamalarından, bu yeni şemalar ile eşzamanlama kayıpları giderilebilecek duruma gelinmiştir [4].

Telsiz karasal haberleşme alanında ise Üçüncü Nesil (3G) Hücresel Standartlar, veri servisleri için turbo kodların kullanımını hedeflemiştir. 3G kapsamında yer alan CDMA2000 (IMT-2000: 1xMC, 3xMC) standardı, normalde ses ve veri hizmetlerini birlikte sunmaktadır. Veri hizmetleri için ise turbo serpiştiricisi ile beraber iki tane özdeş, sekiz durumlu, paralel, RSC bileşen kodlayıcıdan elde edilen ana kod oranı 1/5 olan bir turbo kod kullanılmaktadır. Özellikle 1x modunda, kod delinmesi ile kod oranı 1/2 ve 1/4 oranlarına düşürülerek, 19,2 kbps'dan 307,2 kbps'a kadar kanal hızı imkanı sunulabilmektedir [4].

Turbo ve LDPC kodlar, kayıt sistemleri alanında da tercih edilmiştir. Özellikle manyetik kayıt sistemlerinde, sabit disk sürücülerin kısıtlı sektör boyutları ile kodun blok uzunluğunu sınırlandırmasına ve yüksek kod oranı gereksinimine karşılık, oldukça basit döngülü kod çözme şemaları ile teorik kapasite sınırına 1,5 dB'ye kadar yaklaşacak başarımlar sağlanabilmektedir [4].

Sayısal Video Yayını – Uydu Dönüş Kanalı (DVB-RCS, Digital Video Broadcasting – Return Channel Satellite) standardı genişbant erişimde ADSL ve Kablo Modem'e alternatif olarak ETSI tarafından öne çıkarılmış bir uydu iletişim standardıdır. Bu standart ile uzak bölgelere iki yönlü yüksek hızda internet hizmeti sunulması hedeflenmiştir. Doğal kod oranı 1/3 olan bu standardda 8-durumlu bir turbo kod kullanılmaktadır. Kod delinmesi ile kod oranı 6/7 oranına da düşürülebilmektedir. Bu standart geliştirilerek mobil senaryoları da içerir (DVB-RCS+M) hale dönüştürülmüş ve mobil araçlardan (uçak, tren, otobüs vb.) iki yönlü genişbant uydu iletişimi olanağına kavuşulmuştur [15].

Yine bir DVB standardı olan ve karasal dönüş kanalı içeren DVB-RCT (Digital Video Broadcasting – Return Channel Terrestrial) standardı da sayısal TV hattı üzerinden karasal geri iletim imkanı sunmaktadır. 65 km'ye kadar kapsama olan bu standart için kullanılan turbo kodlayıcı 1/2 ve 3/4 kod oranlarını destekleyebilmektedir [16].

Yeni nesil DVB-RCS standardı olan DVB-RCS2 standardında da turbo kod kullanımı yer almıştır. Daha gelişmiş bir turbo kodlama türü olan ve turbo- $\Phi$  (turbo-phi) olarak adlandırılan bu tür ile kanal kodlamada 16-durumlu kodek ile (QPSK, 8PSK, ve 16QAM için) çok daha düşük hata oranlarına ulaşılmaktadır [17].

LTE (Long Term Evolution) standardı da yüksek bit hızlarını maliyet etkin bir şekilde sunabilen bir mobil genişbant 3GPP standardıdır [18,19]. Bu standart için de kanal kodlayıcı olarak turbo kodlayıcı yer almaktadır. Bu turbo kodlayıcı iki 8-durumlu bileşen kodlayıcı ve bir turbo kod iç serpiştirici içermektedir. Ana kod oranı 1/3 olup, kod delme ile bu oran değişebilmektedir. LTE standardı olgunlaştıktan sonra güncellenen LTE-Advanced ile de 4G işaret edilmektedir.

## VI. SONUÇ

Son yıllarda etkileyici gelişmeler olması ile birlikte, turbo kod çözme algoritmasından hareketle türetilmiş olan turbo ilkesi pratikte geniş yer bulmuştur. Turbo, LDPC ve ilişkili kodların inandırıcılığı deneyler yolu ile ispatlanmış ve sonra da bu kodlama yöntemlerinin çoğu pratik uygulamalarda sistem seviyesinde ana parçalar arasında yerini almıştır.

## VII. KAYNAKLAR

- [1] C.E. Shannon *The Bell System Technical Journal* **27** (1948) 379.
- [2] C. Berrou, A. Glavieux, P. Thitimajshima, *Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes (1)*, **IEEE International Conference on Communications**, (1993) 1064.
- [3] C. Berrou, A. Glavieux *IEEE Transactions on Communications*, **44(10)** (1996) 1261.
- [4] P.H. Siegel, D. Divsalar, E. Eleftheriou, J. Hagenauer, D. Rowitch, W.H. Tranter *IEEE Journal On Selected Areas In Communications* **19(5)** (2001) 793.
- [5] P.H. Siegel, D. Divsalar, E. Eleftheriou, J. Hagenauer, D. Rowitch *IEEE Journal On Selected Areas In Communications*. **19(9)** (2001) 1657.
- [6] Anonim, <http://www.cjbyron.com/3000GTTurbo.htm> (Erişim tarihi: 01/05/2013)
- [7] A.J. Viterbi *IEEE Transactions on Information Theory* **13(2)** (1967) 260.
- [8] L. R. Bahl, J. Cocke, F. Jelinek, J. Raviv *IEEE Transactions on Information Theory* **20(2)** (1974) 284.
- [9] J. Hagenauer, P. Hoeher, *A Viterbi algorithm with soft-decision outputs and its applications*. **IEEE Global Telecommunications Conference, GLOBECOM'89**, (1989) 1680.
- [10] R.J. McEliece *IEEE Transactions on Information Theory* **42(4)** (1996) 1072.
- [11] M.H. Sazlı, *Neural network applications to turbo decoding*, Doktora Tezi, Syracuse Üniversitesi, Amerika Birleşik Devletleri, (2003).
- [12] J.G. Proakis, *Digital Communications*, McGraw Hill (2008).
- [13] C. Heegard, S.B. Wicker, *Turbo Coding*, Kluwer Academic Publishers (1999).
- [14] R.G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA:MIT Press, (1963).
- [15] Anonim, *Interaction Channel for Satellite Distribution Systems*, DVB, ETSI EN 301 790, (V1.5.1), 2009.
- [16] Anonim, *Interaction Channel for Digital Terrestrial Television*, DVB, ETSI EN 301 958, (V1.1.1), 2002.
- [17] Anonim, *Second Generation DVB Interactive Satellite System (DVB-RCS2)*, DVB, ETSI EN 301 545-2, Şubat 2013.
- [18] Anonim, *Evolved Universal Terrestrial Radio Access (E-UTRA), LTE Physical Layer; General Description (Release 10)*, 3GPP TS 36.201. Aralık 2010.
- [19] T. Inseher, F. Kienle, C. Weis, N. Wehn, *A 2.15 GBit/s Turbo Code Decoder for LTE Advanced Base Station Applications*, **Turbo Codes and Iterative Information Processing, ISTC 2012**, (2012) 21.