

Alt Uzay Yöntemleri Kullanarak İşaret Kodlama

Serkan Keser

DOKTORA TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Aralık 2017



Signal Coding by Using Subspace Methods

Serkan Keser

DOCTORAL DISSERTATION

Department of Electric-Electronics Engineering

December 2017

Alt Uzay Yöntemleri Kullanarak İşaret Kodlama

Serkan Keser

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Telekomünikasyon ve Sinyal İşleme Bilim Dalında
DOKTORA TEZİ
Olarak Hazırlanmıştır

Danışman: Prof. Dr. M. Bilginer Gülmezoğlu

Aralık 2017

ONAY

Elektrik-Elektronik Mühendisliđi Anabilim Dalı Doktora öđrencisi Serkan Keser'in DOKTORA tezi olarak hazırladıđı "Alt Uzay Yöntemleri Kullanarak İşaret Kodlama" başlıklı bu çalıřma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek oybirliđi ile kabul edilmiřtir.

Danıřman : Prof. Dr. M. Bilginer Gülmezođlu

İkinci Danıřman : -

Doktora Tez Savunma Jürisi:

Üye : Prof. Dr. M. Bilginer Gülmezođlu

Üye : Prof. Dr. M. Ömer Nezih Gerek

Üye : Yrd. Doç. Dr. Erol Seke

Üye : Doç. Dr. Semih Ergin

Üye : Yrd. Doç. Dr. Muammer Akçay

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve
..... sayılı kararıyla onaylanmıřtır.

Prof. Dr. Hürriyet ERŐAHAN
Enstitü Müdürü

ÖZET

Bu tez çalışmasında, ses sinyallerinin kodlanması için Karhunen Loeve Transform (KLT) ve Independent Component Analysis (ICA) temelli iki yeni yöntem önerilmiştir. Özdeğer büyüklükleri ile uğraşmak yerine, KLT ve ICA tabanlı yöntemler, kovaryans matrislerinin özvektörlerini (veya ICA için bağımsız bileşenleri) daha az sayıdaki vektöre geometrik olarak gruplayarak kullanır. Bu şekilde bir veri sıkıştırması gerçekleştirilmiştir. Daha fazla sıkıştırma işlemi, küçük özdeğerlere karşılık gelen otokovaryans özvektörlerini atarak ve kalan özvektörlerde vektör nicemleme uygulayarak elde edilmiştir. Buna ek olarak bu çalışma, daha iyi SNR için, hatanın etkin bir şekilde iletilmesi amacı ile kullanılabilir bant genişliğinin geri kalan kısmını kullanan, tekrar eden bir hata düzeltme sürecini önermektedir. Tüm süreçler, etkili ses kodlamaya yeni bir yaklaşım oluşturmaktadır; ICA, ilk kez alt uzay ses kodlamasında kullanılmaktadır. Önerilen yöntemlerle Sabit Bit Hızı (CBR) ve Değişken Bit Hızı (VBR) kodlama algoritmaları kullanılmıştır. Deneysel çalışmalarda TIMIT ses veri tabanı kullanılmıştır. Ses sinyalleri, çeşitli çerçeve uzunluklarını kullanılarak 2.4 kbps, 8 kbps, 12.2 kbps, 16 kbps, 16.4 kbps ve 19.85 kbps oranlarında sentezlenmiştir. Sentezlenen ses sinyallerinin kaliteleri, LPC (2.4 kbps), G728 (LD-CELP, 16 kbps), G729A (CS-CELP, 8 kbps), EVS (16.4 kbps), AMR-NB (12,2 kbps) ve AMR-WB (19,85 kbps) ses kodlayıcılar ile karşılaştırılmıştır. Deneysel çalışmalar sonucunda, KLT_VBR(A1) yönteminin ses kodlamada diğer yöntemlere göre daha iyi sonuçlar verdiği görülmüştür.

Anahtar Kelimeler: Bağımsız Bileşen Analizi(ICA), Karhunen Loeve Dönüşümü(KLT), Ses kodlayıcılar, Altuzay Yöntemler.

SUMMARY

In this thesis, two novel methods, which are based on Karhunen Loeve Transform (KLT) and Independent Component Analysis (ICA), are proposed for coding of speech signals. Instead of immediately dealing with eigenvalue magnitudes, the KLT- and ICA-based methods use eigenvectors of covariance matrices (or independent components for ICA) by geometrically grouping these vectors into fewer numbers of vectors. In this way, a data representation compaction is achieved. Further compression is achieved through discarding autocovariance eigenvectors corresponding to the small eigenvalues and applying vector quantization on the remaining eigenvectors. Additionally, this study proposes an iterative error refinement process, which uses the rest of the available bandwidth in order to transmit an efficient representation of the description error for better SNR. The overall process constitutes a new approach to efficient speech coding, with ICA being used in subspace speech coding for the first time. Constant bit rate (CBR) and variable bit rate (VBR) coding algorithms are employed with the proposed methods. TIMIT speech database is used in the experimental studies. Speech signals are synthesized at 2.4 kbps, 8 kbps, 12.2 kbps, 16 kbps, 16.4kbps and 19.85 kbps rates by using various frame lengths. The qualities of synthesized speech signals are compared to those of available speech codecs, i.e., LPC (2.4 kbps), G728 (LD-CELP, 16 kbps), G729A (CS-CELP, 8 kbps), EVS (16.4 kbps), AMR-NB (12.2 kbps) and AMR-WB (19.85 kbps). The results of experimental studies show that KLT_VBR(A1) method provides better results compared to other speech coding methods.

Keywords: Independent Component Analysis (ICA), Karhunen Loeve Transform (KLT), Speech codecs, Subspace methods.

TEŐEKKÜR

Tüm doktora programı boyunca desteęini hiçbir zaman esirgemeyen, danıőman hocam Prof. Dr. M. Bilginer Gülmezoęlu'na, ayrıca tez izleme komitesinde yer alan deęerli hocalarım, Prof. Dr. Ömer Nezh Gerek ve Yrd. Doę. Dr. Erol Seke'ye yardımlarından dolayı teőekkürlerimi sunarım.



İÇİNDEKİLER

Sayfa

ÖZET	vi
SUMMARY	vii
TEŞEKKÜR	viii
İÇİNDEKİLER	ix
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xiii
SİMGELER VE KISALTMALAR DİZİNİ	xv
1. GİRİŞ VE AMAÇ	1
2. LİTERATÜR ARAŞTIRMASI	3
3. KONUŞMA SİNYALLERİNİN ÖZELLİKLERİ	5
3.1 İnsan Konuşma Sistemi Yapısı.....	5
3.1.1 Ötümlü sesler.....	5
3.1.2 Ötümsüz sesler.....	6
3.2 Konuşma Sinyalinin Modellenmesi.....	6
4. SES KODLAMA	8
4.1 Ses Kodlayıcı Çeşitleri.....	8
4.1.1 Dalga biçim kodlayıcılar.....	9
4.1.2 Parametrik kodlayıcılar.....	9
4.1.3 Hibrit kodlayıcılar.....	10
4.2 Ses Kodlayıcıların Teorik Açıklanması.....	10
4.2.1 Darbe kod modülasyonu (PCM).....	10
4.2.2 Farksal darbe kod modülasyonu (DPCM).....	14
4.2.3 Uyarlanırlı Farksal darbe kod modülasyonu (ADPCM).....	16
4.2.4 Doğrusal öngörü kodlayıcı (LPC).....	17
4.2.5 Kod uyartımlı doğrusal öngörü kodlayıcı (CELP).....	23
4.2.6 Cebirsel CELP kodlayıcı (Algebraic-CELP).....	28
4.2.7 Eşlenik yapılı CELP kodlayıcı(Conjugate Structured-CELP).....	29
4.2.8 Düşük gecikmeli CELP kodlayıcı (Low Delay-CELP).....	31
4.2.9 Uyarlanırlı Çoklu Oran kodlayıcı (AMR).....	37

İÇİNDEKİLER (devam)

Sayfa

4.2.10 Gelişmiş Ses Hizmetleri kodlayıcı (EVS).....	38
5. YÖNTEM.....	40
5.1 Karhunen-Loeve Dönüşümü.....	40
5.2 Bağımsız Bileşen Analizi.....	46
5.3 Tez Çalışmasında Gerçekleştirilen Kodlayıcılar.....	47
5.3.1 KLT Temelli Kodlayıcı.....	49
5.3.2 Özvektör Nicemleme.....	54
5.3.3 ICA Temelli Kodlama.....	57
5.3.4 Sabit ve Değişken Bit Oranlı Kodlama.....	61
5.4 Ses Kodlayıcıların Kalite Ölçütleri.....	65
5.4.1 Öznel Kalite Ölçütü.....	66
5.4.2 Ses Kalitesinin Algısal Değerlendirmesi (PESQ).....	67
5.4.3 Ağırlıklandırılmış Spektral Eğim (WSS).....	68
5.4.4 Logaritmik Tahmin Oranı (LLR).....	68
5.4.5 Birleşik ölçüt.....	69
5.5 KLT Temelli Resim Sıkıştırma.....	69
6. BULGULAR VE TARTIŞMA.....	74
7. SONUÇ VE ÖNERİLER.....	84
KAYNAKLAR DİZİNİ.....	87
ÖZGEÇMİŞ.....	91

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
3.1. Konuşma sinyalinin üretim modeli.....	7
4.1. Üç bit nicemleme çözünürlüğü için düzgün nicemleme.....	11
4.2. Düzgün olmayan nicemleyici yapısı.....	12
4.3. μ kuralı için $g(x)$ fonksiyonu.....	13
4.4. A kuralı için $g(x)$ fonksiyonu.....	13
4.5. DPCM alıcı (a) ve verici (b) yapısı.....	14
4.6. Geriye doğru uyarlamalı ADPCM için kodlayıcı (a) ve kod çözücü (b).....	17
4.7. Ses üretimi için LPC modeli.....	18
4.8. LPC kodlayıcı şeması.....	22
4.9. LPC kod çözücü şeması.....	23
4.10. CELP için ses üretim modeli.....	24
4.11. CELP kodlayıcının temel bileşenleri.....	25
4.12. CELP kodlayıcı yapısı.....	26
4.13. CELP kod çözücünün akış diyagramı.....	27
4.14. Farklı α değerleri için bir son filtrenin genlik cevapları.....	28
4.15. Cebirsel (sabit) kod kitabının grafiksel gösterimi.....	30
4.16. LPC'lerin belli çerçeveler ile güncellenmesi.....	33
4.17. LD-CELP (G728) kodlayıcı (a) ve kod çözücü (b) yapısı.....	34
4.18. LD-CELP kodlama döngüsü.....	35
4.19. EVS için kullanılan çeşitli bant frekans aralıkları.....	39
5.1. KLT için 2 boyutta veri dağılımı.....	41
5.2. Bir ses birim sinyalinden $N \times N$ 'lik veri matrislerinin oluşturulması.....	50
5.3. KLT temelli alt uzay yöntem için kodlayıcı blok şeması.....	53
5.4. KLT temelli yöntem için kod çözücü blok şeması.....	54
5.5. İki boyutlu uzayda vektör nicemlemenin gösterimi.....	55
5.6. ICA temelli alt uzay yöntem için kodlayıcı blok şeması.....	60
5.7. ICA temelli yöntem için kod çözücü blok şeması.....	61
5.8. Önerilen KLT_CBR ve ICA_CBR için kodlayıcı-kod çözücü kısımlardaki bit atama yapısı.....	63

ŞEKİLLER DİZİNİ (devam)**Sekil****Sayfa**

5.9. Önerilen KLT_VBR ve ICA_VBR için (a) sesli ve (b) sessiz çerçevelere ait bit dizisi yapıları.....	64
5.10. PESQ algoritması uygulanarak PESQ değerinin elde edilmesi.....	67
5.11. JPEG için kodlayıcı ve kod çözücü blok diyagram.....	70
5.12. KLT ile resim kodlama için kodlayıcı ve kod çözücü blok diyagramı.....	71
6.1. TIMIT veri tabanında bulunan 61 adet ses birim.....	74
6.2. TIMIT veri tabanına ait sesli ses birimlerin sayıları.....	75
6.3. TIMIT veri tabanına ait sessiz ses birim sayıları.....	75
6.4. KLT_VBR (A1) ve diğer kodlayıcılar için elde edilen PESQ değerleri.....	82
6.5. KLT_VBR (A1) ve diğer kodlayıcılar için elde edilen Cov değerleri.....	82

ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
4.1. Üç bitlik nicemleme çözünürlüğü için nicemleme.....	12
4.2. FS1015 kodlayıcı için bit ataması.....	21
4.3. Uyarım vektörü için darbe konumları.....	30
4.4. G729 kodlayıcı için bit ataması.....	31
4.5. 3GPP mobil kodlayıcılar.....	39
5.1. KLT_CBR ve ICA_CBR yöntemleri için bit atamaları.....	62
5.2. KLT_VBR ve KLT_CBR için farklı bit oranlarındaki bit ataması.....	65
5.3. MOS değerleri ve karşılık gelen ses kalitesi dereceleri.....	66
5.4. Birinci yöntem (Y1) ile bulunan PSNR değerleri.....	72
5.5. İkinci yöntem (Y2) ile bulunan PSNR değerleri.....	72
5.6. Üçüncü yöntem (Y3) ile bulunan PSNR değerleri.....	73
6.1. 16 kbps için PESQ değerleri (KLT_VBR (A1))	77
6.2. 16 kbps için Cov değerleri (KLT_VBR (A1))	77
6.3. 16 kbps için PESQ değerleri (KLT_CBR (A1))	77
6.4. 16 kbps için Cov değerleri (KLT_CBR (A1))	77
6.5. 8 kbps için PESQ değerleri (KLT_VBR (A1))	78
6.6. 8 kbps için Cov değerleri (KLT_VBR (A1))	78
6.7. 8 kbps için PESQ değerleri (KLT_CBR (A1))	78
6.8. 8 kbps için Cov değerleri (KLT_CBR (A1))	78
6.9. 2.4 Kbps için ortalama MOS, PESQ ve Cov değerleri.....	79
6.10. Nicemleme yöntemi ile 16 Kbps ve 19.85 Kbps için bulunan ortalama PESQ ve Cov değerleri.....	79
6.11. 8 Kbps için ortalama MOS, PESQ ve Cov değerleri.....	80
6.12. 12.2 Kbps için ortalama MOS, PESQ ve Cov değerleri.....	80
6.13. 16 Kbps için ortalama MOS, PESQ ve Cov değerleri.....	81
6.14. 16.4 Kbps için ortalama MOS, PESQ ve Cov değerleri.....	81

ÇİZELGELER DİZİNİ (devam)

<u>Cizelge</u>	<u>Sayfa</u>
6.15. 19.85 Kbps için ortalama MOS, PESQ ve Cov değerleri.....	81
6.16. KLT ve ICA tabanlı yöntemler için çerçeve başına kodlayıcı gecikmeleri.....	83
6.17. KLT ve ICA tabanlı yöntemler için çerçeve başına kod çözücü gecikmeleri.....	83



SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

$R(\tau)$

Açıklama

Zamana bağlı korelasyon fonksiyonu

Kısaltmalar

ACELP

Algebraic Code Excited Linear Prediction (Cebirsel CELP)

ADM

Adaptive Differential Modulation (Uyarlamalı Farksal Modülasyon)

ADPCM

Adaptive Differential PCM (Uyarlamalı Darbe Kod Modülasyonu)

AMR

Adaptive Multi Rate (Uyarlamalı Çoklu Oran)

AMR-NB

AMR- Narrow Band (Uyarlamalı Çoklu Oran-Dar Bant)

AMR-WB

AMR – Wide Band (Uyarlamalı Çoklu Oran-Geniş Bant)

CBR

Constant Bit Rate (Sabit Bit Oranı)

CELP

Code Excited Linear Prediction (Kod Uyarımlı Doğrusal Öngörü)

CRC

Cyclic Redundancy Check (Çevrimsel Artıklık Denetimi)

CS-ACELP

Conjugate Structure - ACELP (Eşlenik Yapılı- ACELP)

DCT

Discrete Cosine Transform (Ayrık Kosinüs Dönüşümü)

DPCM

Differential PCM (Farksal Darbe Kod Modülasyonu)

EVS

Enhanced Voice Services (İyileştirilmiş Ses Servisleri)

FIR

Finite Impulse Response (Sonlu Darbe Cevabı)

FFT

Fast Fourier Transform (Hızlı Fourier Dönüşümü)

GSM

Global System for Mobile Communication

ICA

Independent Component Analysis (Bağımsız Bileşen Analizi)

IIR

Infinite Impulse Response (Sonsuz Darbe Cevabı)

ITU

International Telecommunication Union (Uluslararası Tel. Birliği)

KLT

Karhunen Loeve Transform (Karhunen Loeve Dönüşümü)

LD-CELP

Low Delay - CELP (Düşük Gecikmeli CELP)

LPC

Linear Predictive Coding (Doğrusal Öngörü Kodlayıcı)

MELP

Mixed Excitation Linear Prediction (Karma Uyarımlı LP)

MOS

Mean Opinion Score (Ortalama Görüş Puanı)

SİMGELER VE KISALTMALAR DİZİNİ (Devam)

<u>Kısaltmalar</u>	<u>Açıklama</u>
PCM	Pulse Code Modulation (Darbe Kod Modülasyonu)
PESQ	Perceptual Evaluation of Speech Quality
PSNR	Peak Signal Noise Ratio (Tepe Sinyal Gürültü Oranı)
VBR	Variable Bit Rate (Değişken Bit Oranı)
WT	Wavelet Transform (Dalgacık Dönüşümü)



1. GİRİŞ VE AMAÇ

Günümüzde konuşma veya ses sinyallerinin kodlanması için çok geniş bir yelpazede kodlayıcılar kullanılmaktadır. Kodlayıcılar alanında yapılan yoğun çalışmalar ve teknolojik gelişmeler ile birlikte daha kaliteli kodlayıcılar geliştirilmektedir. Gerek yüksek kaliteli ses sentezleme, gerekse düşük kodlama gecikmesini sağlayan kodlayıcılar genellikle mobil iletişim alanında kullanılmaktadır. Düşük bit oranlarında yüksek kalite sağlayan bu kodlayıcıların geliştirilme süreci devam etmekte ve kullanıcıların memnuniyetinin sağlanması esas amaç olarak görülmektedir. Bu kodlayıcılar genel olarak, ses üretim yolunun matematiksel modellerinin temel alındığı doğrusal öngörüye dayanmaktadır (Deng ve O'Shaughnessy, 2003).

Doğrusal öngörü dışında bir sinyal, Karhunen Loeve Dönüşümü, Bağımsız Bileşen Analizi, Ayrık Kosinüs Dönüşümü, Hızlı Fourier Dönüşümü ve Dalgacık Dönüşümü gibi uygun dönüşümler uygulanarak elde edilen dönüşüm katsayılarının küçük bir kısmından yeniden oluşturulabilir. Bizim çalışmamızda KLT ve ICA gibi alt uzay yöntemleri ile ses kodlama gerçekleştirilmiştir. Temel olarak bu yöntemlerin literatürde ses kodlama için kullanılmadığı görülmüştür. Dolayısıyla, bu alt uzay yöntemleri ile daha fazla araştırma yapmak, çalışmaya başlamadan önceki temel düşüncemiz olmuştur. Bu çalışmada, İngilizce ses veri tabanı TIMIT'in içerdiği fonemlerin tümü kullanılarak tüm dili kapsayacak bir ses kodlayıcı oluşturulabileceği, bant genişliğinin bir hata azaltma işlemi ile tamamen kullanılarak sinyalin SNR değerinin iyileştirilebileceği ve KLT tabanlı kodlamada kullanılan özvektör matrislerine nicemleme uygulayarak hesaplama gecikmesinin azaltılması ile iletişim sistemlerinde kullanılacak bir kodlayıcı geliştirilebileceği savunulmaktadır. Bu tezde yapılan çalışma, doğrusal öngörü temelli kodlayıcıların yapısından tamamen bağımsızdır. KLT ve ICA tabanlı yöntem, vektörleri az sayıda vektör ile geometrik olarak gruplandırarak elde edilen kovaryans matrislerinin özvektörlerini (veya ICA için bağımsız bileşenleri) kullanır. KLT temelli yöntem ile daha fazla veri sıkıştırma, küçük özdeğerlere karşılık gelen oto-kovaryans özvektörlerini atarak ve kalan özvektörlere de vektör nicemleme uygulayarak elde edilir. Buna ek olarak bu çalışmada, daha iyi SNR değeri elde etmek için bant genişliğinin geri kalan kısmını

kullanan, tekrar eden bir hata düzeltme süreci önerilmektedir. Tüm bu süreçler, etkili ses kodlamak için yeni bir yaklaşım oluşturmaktadır.

Ayrıca bu tez çalışmasında, Sabit Bit Oranlı (CBR) ve Değişken Bit Oranlı (VBR) olmak üzere iki farklı bit oranı kullanılmaktadır. Bu şekilde KLT ve ICA için KLT_VBR, KLT_CBR, ICA_VBR ve ICA_CBR olmak üzere dört yöntem uygulanmıştır. Elde edilen başarımlar, farklı bit oranları ve farklı çerçeve uzunlukları için bulunmuş ve geleneksel ses kodlama yöntemlerinin başarımları ile karşılaştırılmıştır. Karşılaştırılan bu ses kodlama yöntemleri sırası ile LPC, Eşlenik-Yapı Cebirsel Kod Uyarımlı Doğrusal Öngörü (ACELP) G729A, LD-CELP (G728), AMR-NB (12.2kbps), AMR-WB (19.85kbps) ve EVS (16.4kbps)'dir. Sentezlenen sesler için, ortalama görüş puanı (Mean Opinion Score=MOS), Ses Kalitesinin Algısal Değerlendirilmesi (PESQ) ve birleşik ölçüt (Composite Measure = Cov) olmak üzere 3 adet değerlendirme ölçütü kullanılmıştır. Ses veri tabanı olarak, Amerikan İngilizcesi içeren TIMIT kullanılmıştır. Veri tabanındaki ses verileri, 16 Khz örnekleme frekansına sahip dijital sinyallerden oluşturulmuştur (16 bit, PCM format). Veri tabanındaki ses birimlerinin (fonemlerin) sınırları ve sembolleri etiketlenmiştir. Bu etiketlerdeki bilgiler kullanılarak 45 ses birim sınıfına ait ses verileri, KLT ve ICA yöntemler ile ses sentezlemek için kullanılmıştır.

Bu tez çalışmasında, standart kodlayıcıların yapısı dışında bir çalışma gerçekleştirilmiştir. Çalışmadaki temel hedef, hem hesaplama gecikmesi bakımından, hem de sentezlenen ses sinyallerinin kalitesi açısından standart kodlayıcılar ile karşılaştırıldığında benzer ya da daha iyi sonuçlar elde etmektir. Bunu sağlamak için ses ve görüntü kodlamada yeni bazı teorik yaklaşımlar geliştirilmiştir. Özellikle KLT için geliştirilen yaklaşımlarda, özvektör matrislerinin temel bileşen yönlerinde kümelenerek nicemleme yapılması ile oldukça az bir hesaplama gecikmesi elde edilmiştir. Ayrıca sentezlenen ses sinyallerinin kalitesinin nicemleme ile çok fazla azalmaması için nicemleme sonucu bulunan özvektör kümeleri içinde arama yapmayı içeren farklı bir yöntem de kullanılmıştır. Literatürde ses kodlama için hiç kullanılmamış olan ICA, ilk kez bu tez çalışmasında kullanılmıştır. Ayrıca bu çalışma, istenilen bit oranı elde edilinceye kadar, çıkıştaki hata sinyalini iteratif olarak girişe besleyen ve bu sayede Sinyal Gürültü Oranını (SNR) iyileştiren bir hata azaltma işlemi de içermektedir.

2. LİTERATÜR ARAŞTIRMASI

Ses kodlamada temel amaç yüksek kalitede ve düşük bit oranında konuşma sentezlemektir (Gibson, 2005). Gerek yüksek kaliteli ses sentezleme, gerekse düşük kodlama gecikmesini sağlayan kodlayıcılar genellikle mobil iletişim alanında kullanılmaktadır. Düşük bit oranlarında yüksek kalite sağlayan bu kodlayıcıların geliştirilme süreci devam etmekte ve kullanıcıların memnuniyetinin sağlanması esas amaç olarak görülmektedir. Bu kodlayıcılar genel olarak, ses üretim yolunun matematiksel modellerinin temel alındığı doğrusal öngörüye dayanmaktadır (Deng ve O'Shaughnessy, 2003). Ayrıca ses kodlayıcıların ilk örneklerinden olan darbe kod modülasyonu (PCM), ses sinyalinin örneklerini temel alarak kodlama gerçekleştirmektedir (Kondo, 2007). Bundan dolayı PCM, dalga-biçim (waveform) kodlayıcı olarak isimlendirilir. PCM dünyadaki ilk dalga biçim kodlama standartıdır. Bu kodlayıcı 64kbps bit oranını kullanmaktadır. Kodlayıcı kalitesi olarak en iyi kodlayıcı biçimidir ancak yüksek bit oranına sahiptir. Sonraki yıllarda daha ileri teknikler kullanan ADM, ADPCM gibi kodlayıcılar geliştirilmiştir (Kondo, 2007).

Kodlayıcılarda, yüksek bit oranında yüksek kaliteli ses sentezlenmesi ve düşük kodlama gecikmesi önemli özelliklerdir. Bundan dolayı, doğrusal öngörü temelli ses kodlama teknikleri (CELP, MELP, VSELP vb.), günümüzde yaygın olarak kullanılmaktadır (Chen vd.,1992; Gerson ve Jasiuk,1990). Kullanılan bant genişliğinin artması ile birlikte ses kalitesi de artmıştır. Geniş bant, süper geniş bant ve tam bant kullanan İyileştirilmiş Ses Servisleri (EVS) gibi kodlayıcılar günümüz mobil iletişimde yaygın olarak kullanılmakta olan doğrusal öngörü temelli kodlayıcılardır (Atti vd., 2005).

Öncelikle çalışma konusu olarak, alt uzay yöntemleri kullanarak ses kodlama belirlenmiştir. Literatür incelendiğinde, ses kodlama standartlarının daha çok doğrusal öngörü temelli olduğu görülmüştür. Alt uzay tabanlı ses kodlama çalışması literatürde oldukça azdır (Kim ve Kleijn, 2004; Lee ve Kim, 2010; Ozerov ve Kleijn, 2011; Ju vd., 2014 , Oger vd., 2006). Ses veri tabanı olarak, literatürdeki son 10 yıla ait makaleler incelenerek seçilen TIMIT veri tabanı kullanılmıştır. TIMIT ses veri tabanı ses birim temelli oluşturulmuş ve genel olarak otomatik konuşma tanıma sistemlerinde, çeviri, dikte

etme, komut ile kontrol etme gibi alanlarda kullanılmıştır (Zue vd., 1990). Rakam tanıma, ses birim tanıma alanlarında da pek çok çalışma yapılmıştır. Tez çalışmasında, ses ve resim kodlama için KLT ve ICA temelli alt uzay yöntemler önerilmiştir. Literatür incelendiğinde işaret kodlama için KLT kullanılmasına karşın, ICA'nın kullanılmadığı görülmüştür. KLT, hem görüntü kodlamada hem de ses sinyallerinin kodlanmasında kullanılmaktadır. Ses kodlamada KLT alt uzay yöntemi, doğrusal öngörü temelli kodlayıcıların (LPC, CELP vb.) temel yapıları içerisinde bulunan sese ait filtre katsayılarının kodlamasını içermektedir (Kim ve Kleijn, 2004; Lee ve Kim, 2010; Ozerov ve Kleijn, 2011; Ju vd., 2014 , Oger vd., 2006). KLT alt uzay yöntemini kullanan bazı çalışmalarda, hesaplama maliyetinin ve karmaşıklığın standart ses kodlayıcılara göre daha az olduğunu belirtmektedir (Lee ve Kim, 2010; Oger vd., 2006). Ancak bu çalışmaların tümü standart kodlayıcıların yapısı içerisindeki çalışmalardan oluşmaktadır. ICA temelli yöntemin daha çok görüntü işleme alanında yüz ve nesne tanıma, boyut azaltma (Hyvärinen, 2013; Wang ve Chang, 2006) gibi yöntemler için kullanıldığı görülmüştür.

3. KONUŞMA SINYALLERİNİN ÖZELLİKLERİ

Konuşma sinyali, ses üretim sistemini meydana getiren anatomik yapıların hareketleri sonucunda oluşan akustik bir basınç dalgasıdır. Bu sistemin ana bölümleri ciğerler, nefes borusu, gırtlak, boğaz, ağız boşluğu ve burun boşluğudur. Burada boğaz ve ağız boşluğu 'ses yolu' olarak tanımlanır. Dolayısıyla ses yolu, gırtlak çıkışından başlayıp, dudaklarda sona erer (Selen, 1979).

3.1 İnsan Konuşma Sistemi Yapısı

Ses üretimi için kritik olan anatomik yapılar, ses telleri, damak, dil, dişler ve dudaklardır. Ses yolunu oluşturan bu anatomik yapılar farklı pozisyonlar alarak değişik sesleri oluştururlar. Ses üretimi bir akustik filtreleme işlemi olarak düşünülebilir. Akustik filtre, ses üretim yollarının özelliklerini gösterir. Ses üretim yolu, ham sesi formant frekanslarında rezonansa uğratarak 1 ms'nin katlarında korelasyonlara sahip olacak şekilde biçimlendirir. Söz konusu formantların frekansları, sinir sistemince ses yolundaki organların pozisyonları değiştirilerek ayarlanır. Matematiksel anlamda ses üretim yolu, kısa süreli (short-term) özellikli bir süzgeçtir ve bahse konu bu süzgecin parametreleri zamanla, 20 ms'nin katlarında yavaş bir şekilde değişim gösterir (Dunn vd., 2001). Konuşma sinyali, maddeden oluşan bir ortamda moleküllerin titreşmeleriyle, sıkışıp genleşmeleri ile dalgalar halinde yayılır ve genellikle 500 – 2000 Hz frekansına sahip enine ve boyuna sinüzoidal dalgalardan meydana gelmektedir. Konuşma sinyali, anlık uyarım durumuna göre ötümlü ve ötümsüz sesler olmak üzere iki farklı sınıfa ayrıştırılabilir (Dunn vd., 2001).

3.1.1 Ötümlü sesler (voiced sounds)

Ötümlü sesler, ses tellerinin açılıp kapanmasıyla ve böylece ciğerlerden gelen havanın ara ara kesilip serbest bırakılmasıyla üretilirler. Ses tellerinin bu şekilde kullanılması neticesinde yarı periyodik (quasi-periodic) darbeler oluşur ve bu darbelerin oluşma sıklığı (rate) konuşma sinyaline ait perde periyodunu (pitch period) belirler. Perde

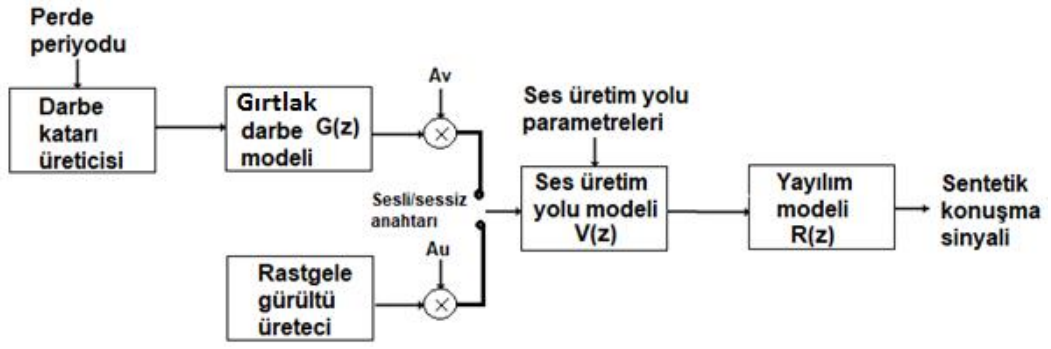
periyotları 2 ile 20 ms'lik değerler arasında deęişkenlik gösterir. Ötümlü seslerde, hem kısa süreli hem de uzun süreli korelasyonlar bulunur.

3.1.2 Ötümsüz sesler (unvoiced sounds)

Ötümsüz sesler gırtlak açıkken (ses telleri devrede deęilken) cięerlerden yola çıkan yüksek hızlı hava akımlarının ses üretim yolunda sürtünmeye uğratarak akıtılmasıyla oluşturulur. Ses üretim yolunun engelleyici ve sürtünmeyi arttırıcı bir yapıya getirilmesi hızlı hava akımının türbülansa girmesine yol açar. Söz konusu türbülstan geçen hava akımı gürültüyle benzer özellikler gösterir. Ötümsüz seslerde sadece kısa süreli korelasyonlar bulunur.

3.2 Konuşma Sinyalinin Modellenmesi

İnsanların duyabildięi frekans aralığı ses spektrumu olarak adlandırılır. Ses spektrumunun bant genişlięi 20 Hz ile 20 KHz arasındadır. Ancak insanların çoęu daha dar bir bant aralığını duyabilir. Konuşma sinyalinin sadece dalga biçimi dikkate alındığında, sinyal 4 ila 10 kHz bant genişlięine yayılabilmektedir (Rabiner ve Schafer, 2007) ve buna baęlı olarak 8 ila 20 kHz'de örneklenmelidir. Dalga şeklindeki ses sinyallerinin karakteristięi kısa zaman aralıkları için sabittir. Bu yüzden genel olarak 10-30ms aralıęındaki çerçeveler kullanılarak konuşma sinyalleri analiz edilir. Şekil 3.1'de en genel ses üretim modeli verilmiştir (Rabiner ve Schafer, 2007). Bu modelde ötümlü ve ötümsüz sesler için farklı şekillerde uyartımlar oluşturulmaktadır. Anahtarlanmış uyartım sinyali ses üretim yoluna girmekte, $V(z)$ tarafından biçimlendirilmektedir. Girdi uyartım sinyaline kısa süreli korelasyonlar kazandırılarak konuşma sinyali elde edilmektedir. Bahse konu ses üretim yolu birbirlerine peş peşe eklenmiş kayıpsız içi boş silindirlere benzeyen tüpler olarak modellenmiştir (Kelly ve Carol, 1962). Kayıpsız tüplerin genişlikleri ve uzunlukları sinir sistemi tarafından ayarlanmakta, bu sayede istenilen niteliklerde süzgeçler yaratılabilmektedir.



Şekil 3.1. Konuşma sinyalinin üretim modeli

4. SES KODLAMA

Sayısal ses sinyallerinin sıkıştırılmasına 1960'lı yıllarda telefon şirketleri tarafından başlandı. 1990'lı yıllarda ise kullanılan algoritmaların iyileştirilmesi ile ses kalitesinde artma gözlemlendi. Ses kodlama, bir konuşma sinyalini temsil etmek için gerekli enformasyon miktarını azaltan bir yöntemdir ve ses sıkıştırma olarak da adlandırılmaktadır. Ses analogdur ve işlenebilmesi için önce örneklenebilir, nicemleme yapılmalı ve ardından sayısal forma dönüştürülmelidir. Sayısal forma dönüştürülen sinyallerin iletişim hatlarından gönderilmesi için verici tarafta kodlayıcı ve alıcı tarafta kod çözücüye ihtiyaç duyulur. Örneğin, örnekleme frekansı 8khz ve örnek başına 16 bit kullanılarak gerçekleştirilen sayısal sinyal için 128 kbps'lik bir veri akışı sağlanır. Bu değerden daha küçük değerler için yapılan iletim hızı için sıkıştırma işlemi yapmak gerekir. Ses kodlama olabildiğince az bit ile gerçekleştirilmeli ve kod çözücü kısmında sentezlenen ses sinyalinin kalitesinde fazla bir azalma olmamalıdır. Ses kodlayıcıların bit oranı, gecikme süresi, karmaşıklık düzeyi ve kalite olmak üzere 4 ana parametresi vardır (Kleijn ve Kuldip, 1995). Bu parametrelerden bit oranı, sıkıştırma başarımının ölçüsüdür. Bu değerden küçük değerler sıkıştırılmış olarak kabul edilir. Kodlama gecikmesi, gerçek zamanlı veri iletiminde konuşma sinyalinin kodlayıcı ve kod çözücüde ne kadar sürede üretildiğini belirtmektedir. Bu gecikmenin konuşmanın anlaşılır olabilmesi için 300 ms'den az olması gerekir. Sentezlenen konuşma sinyalinin kalitesi ise neredeyse tamamen öznelidir. Ancak günümüzde nesnel kalite ölçütleride geliştirilmiştir. Ayrıca kodlayıcı ve kod çözücüde kullanılan algoritmanın karmaşıklığı da maliyet, işlem gücü ve depolama alanı bakımından önemlidir. Ses kodlamanın pek çok çeşidi genellikle kayıplı bir algoritmaya sahiptir. Ancak kod çözücüde elde edilen konuşma sinyalindeki kalite kaybı kabul edilebilir seviyede olmalıdır.

4.1 Ses Kodlayıcı Çeşitleri

International Telecommunications Union (Uluslararası Telekomünikasyon Birliği, kısaca ITU), merkezi Cenevre'de bulunan ve telekomünikasyon dalında birçok standardı belirleyen bir kurumdur. Bu kurum, pek çok kodlayıcı çeşidini standartlaştırmıştır.

Standartlaştırılan bu ses kodlayıcılar parametrik, hibrit ve dalga biçim olmak üzere 3 farklı çeşide sahiptir (Kondoç, 2007).

4.1.1 Dalga biçim kodlayıcılar

Ses sinyalinin üretim şekline dair herhangi bir bilgiden yararlanmaksızın, sentezlenen ses sinyaline ait dalga biçiminin orijinal ses sinyaline en az hata ile yaklaştırmaya çalışan kodlayıcı çeşididir. Bu kodlayıcılar sinyalin içeriğinden bağımsız çalıştıkları için ses sinyalleri dışında akustik sinyallerin kodlanmasında da kullanılırlar. Genel olarak yüksek bit oranlarında yüksek kaliteli ses sentezlerler. En basit dalga biçimi kodlaması PCM'dir. 64 kbps gibi yüksek bir bit oranına sahiptir. Bu kodlayıcının bir avantajıda düşük gecikme süresine sahip olmasıdır. Yüksek bit oranını düşürmek için ses sinyallerindeki ardışık değerler arasındaki korelasyondan yararlanır. Böylece bir sonraki değer, önceki bir kaç değerden tahmin edilebilmektedir. Bu kodlayıcı biçimi ITU G721 ile standartlaştırılan ADPCM olarak adlandırılmaktadır. ADPCM 16, 24, 32 ve 40 kbps gibi bit oranlarına sahiptir. 32 kbps bit oranına sahip versiyonunun kalitesi 64 kbps'lik PCM kodlayıcının ses kalitesine yakındır.

4.1.2 Parametrik kodlayıcılar

Bu kodlayıcı çeşidi, ses sinyallerini oldukça fazla sıkıştırma kabiliyetine sahiptir. İnsan ses yolunun çalışma prensibini temel alır. Filtre ve doğrusal öngörü katsayılarının kullanılması ile oluşturulmuştur. Kodlayıcı kısmında ses yolu parametreleri ve uyarım sinyallerinin periyodu bulunur. Bu veriler nicemlenir ve iletim hattından kod çözücüye gönderilir. Bu kodlayıcıların en çok bilineni LPC'dir ve 2.4 kbps bit oranında çalışır (Gibson, 2005). Kod çözücü kısmında elde edilen ses sinyalinin doğallıktan uzak olduğu görülür. Bu durum kodlayıcının insan ses üretim yolunun basit bir modellemesini içermesinden oluşmaktadır. Bit oranları artırılrsa bile ses kalitesinin belli bir seviyeyi geçmediği görülür. Bu kodlayıcı, yüksek bir işlem gücü gerektirir ve dalga biçim kodlayıcılara göre daha fazla gecikme süresine sahiptir.

4.1.3 Hibrit kodlayıcılar

Bu kodlayıcılar, dalga biçim ve parametrik kodlayıcıların bir karışımıdır. Kullanılan kodlama teknikleri, 4.8 kbps ile 16 kbps bit oranı aralığını kullanır. Ses üretim modeli parametrik kodlayıcılara benzerdir ancak uyartım sinyalleri farklıdır. Sentetik ses sinyalinin orijinal sinyale benzemesi için AbS (Sentez ile Analiz = Analysis by synthesis) yöntemlerini kullanarak uyartım girdilerini ayarlar. Hibrit kodlayıcılar, tüm olası durumların olduğu bir kod kitabı yapısı kullanır. Bu kod kitabı yardımı ile orijinal sinyal, her bir sentezlenmiş sinyal ile karşılaştırılarak en iyi algısal uygunluğu sağlayan sentezlenmiş sinyal bulunur. Karşılaştırma işlemi ağırlıklandırılmış en küçük kareler yöntemi ile yapılır. En iyi uyartım sinyalinin bulunması için algısal bozunum ölçütü kullanıldığından diğer kodlayıcılardan daha iyi sonuç verir. CELP yaygın olarak kullanılan bir hibrit kodlayıcıdır ve pek çok farklı türevi oluşturulmuştur. Orta seviye bit oranlarında iyi kalitede ses sentezleme yaparlar.

4.2 Ses Kodlayıcıların Teorik Açıklanması

Bu bölümde dalga biçim kodlayıcılar ile birlikte çalışmada kullanılan kodlayıcılar teorik olarak geniş biçimde açıklanmaktadır. Böylece her kodlayıcının nasıl çalıştığına dair daha kapsayıcı bilgilere yer verilmiştir.

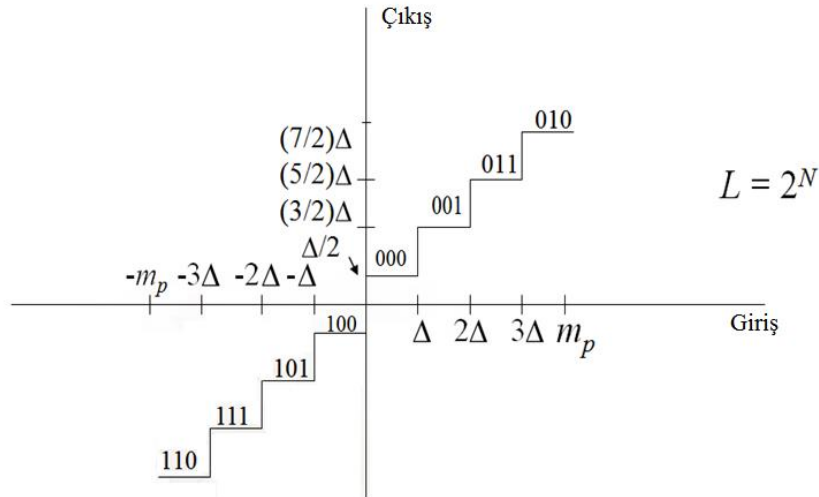
4.2.1 Darbe kod modülasyonu (PCM)

Teorik olarak PCM, ayrık zamanlı sinyal örneklerini nicemleme temeline dayanır (Cattermole,1969). Böylece hem zaman hemde genlik ayrık biçimde oluşturulur. Darbe kod modülasyonu üç kademeli bir işlemdir. İlk işlem, sürekli zamanlı analog sinyalin ayrık zamanlı sinyale dönüştürüldüğü örneklemedir. Bu teorem Shannon örnekleme teoremi veya Nyquist örnekleme teoremi olarak da adlandırılır. Teorem temel olarak, sürekli zamanlı ve sınırlı bantlı bir $x(t)$ analog sinyalinin bozulmadan elde edilebilmesi için örnekleme işleminin nasıl olması gerektiğini belirtmektedir. Burada $x(t)$ sinyalinin en yüksek frekansının (f_m) en az iki katı frekansında ($2f_m$) örnekleme yapılır. Burada örnekleme aralığı;

$$T_s \leq \frac{1}{2f_m}, \quad (4.1)$$

olmalıdır. Nicemleme, ayırık zamanlı bir sinyalin genlik değerlerinin belli bir oranda yeniden temsil edilmesidir. Kullanılan oran, nicemleme adımı olarak adlandırılır ve nicemleme çözünürlüğüne bağlı olarak değişmektedir. n bitlik bir nicemleme için nicemleme sayısı 2^n 'e eşittir. PCM düzgün ve düzgün olmayan nicemleyici çeşitlerine sahiptir.

Düzgün nicemleyicide ses sinyalinin büyük ve küçük genlikli kısımları için aynı büyüklükte nicemleme adımı kullanılır (Proakis, 2001). Ancak bu, ses sinyalindeki sinyal-gürültü oranının küçük ve büyük genlikli kısımlar arasında dengesizleşmesine sebep olmaktadır. Şekil 4.1'de 3 bitlik nicemleme çözünürlüğü için düzgün nicemleyicinin nicemleme değerleri ve ikili indisler verilmiştir.



Şekil 4.1. Üç bit nicemleme çözünürlüğü için düzgün nicemleme.

Şekil 4.1 de m_p , giriş sinyalinin maksimum genlik değerini göstermektedir. Çizelge 4.1 deki Δ değeri, nicemleme adım büyüklüğüdür ve analog sinyalin maksimum genlik değeri (m_p) ile ilişkilidir.

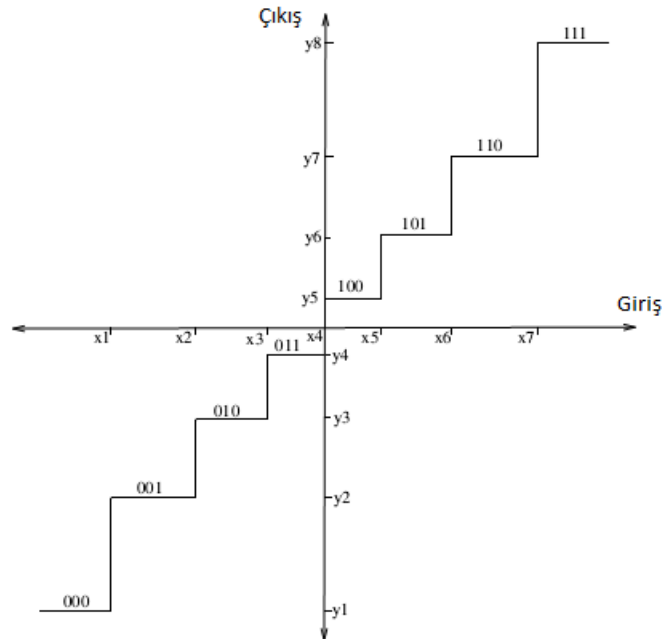
$$\Delta = \frac{2m_p}{2^n}. \quad (4.2)$$

Her nicemleme seviyesine karşılık gelen ikili indislere ait bitler Çizelge 4.1'de verilmiştir. 3 bitlik nicemleme çözünürlüğü için toplam 8 adet nicemleme seviyesi oluşur.

Çizelge 4.1. Üç bitlik nicemleme çözünürlüğü için nicemleme

İkili indisler bitleri	Giriş nicemleme değeri	Çıkış nicemleme değerleri
010	3Δ	$7\Delta/2$
011	2Δ	$5\Delta/2$
001	Δ	$3\Delta/2$
000	0	$\Delta/2$
100	$-\Delta$	$\Delta/2$
101	-2Δ	$-3\Delta/2$
111	-3Δ	$-5\Delta/2$
110	-4Δ	$-7\Delta/2$

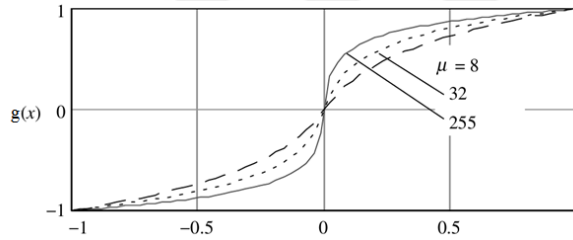
Ses sinyallerin kodlanmasında giriş sinyali dağılımı düzgün dağılımdan uzaklaşır. Ses dalga formunun düşük genlikli kısımları için yüksek olasılık ve yüksek genlikli kısımlar için düşük olasılık mevcuttur. Bu yüzden küçük genlikler için küçük nicemleme adımları ve büyük genlikler için büyük nicemleme adımları kullanarak sinyal-gürültü oranının aynı olmasını sağlayan düzgün olmayan nicemleyici geliştirilmiştir (Proakis, 2001). Bunu gerçekleştirmek için, iletişim sistemlerinde bir sıkıştırma (compression) yapılmaktadır. Şekil 4.2’de düzgün olmayan nicemleyici yapısı görülmektedir.

**Şekil 4.2.** Düzgün olmayan nicemleyici yapısı.

Alıcı uçta doğrusal olmayan işlemin tersi (genleştirme), örneklenmiş işaretleri elde etmek için kullanılır. Bu teknik, sıkıştırma-genleştirme (compressing-exanding) olarak da bilinir. Ses kodlama için yaygın olarak kullanılan 2 çeşit sıkıştırma-genleştirme mevcuttur. İlki μ kuralıdır. Verici tarafta $|x| \leq 1$ için aşağıdaki logaritmik fonksiyonu kullanır.

$$g(x) = \frac{\log(1 + \mu|x|)}{\log(1 + \mu)} \operatorname{sgn}(x). \quad (4.3)$$

μ miktarı, sıkıştırma-genleştirme oranını ayarlar. Şekil 4.3.'te μ kuralı için $g(x)$ fonksiyonunun grafiği verilmiştir.

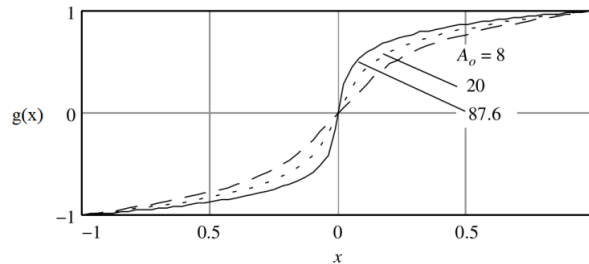


Şekil 4.3. μ kuralı için $g(x)$ fonksiyonu

İkinci yaygın kullanılan logaritmik sıkıştırıcı, A-kuralı sıkıştırıcıdır (genleştiricidir). Bu sıkıştırıcının (genleştirici) logaritmik fonksiyonu;

$$g(x) = \frac{1 + \log(A|x|)}{1 + \log A} \operatorname{sgn}(x), \quad (4.4)$$

olarak verilmiştir. Uygulamalarda $A=87.6$ olarak seçilir. μ kuralı ile A kuralı sıkıştırıcı-genleştirici performansları benzerdir. Şekil 4.4'de $A=8$, 20 ve 87.6 için sıkıştırıcı-genleştiricinin logaritmik fonksiyonu gösterilmiştir.

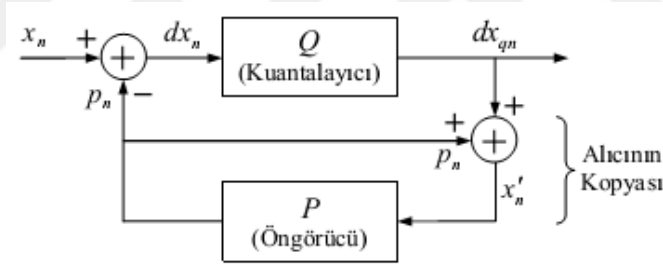


Şekil 4.4. A kuralı için $g(x)$ fonksiyonu

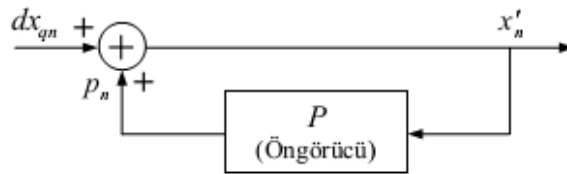
Ayrık zamanlı sinyalin her örneğine atanan n -bit, iletim hattı üzerinden kod çözücüye gönderilir. Kod çözücü, aynı nicemleme tablosunu kullanarak ve yuvarlatılmış değerleri yeniden oluşturarak ses sinyalini kayıplı şekilde yeniden oluşturur.

4.2.2 Farksal darbe kod modülasyonu (DPCM)

DPCM, nicemlenmesi gereken değerlerin değişim aralığını azaltmak için geçmişteki örnek değerleri kullanılarak mevcut örnek değeri için bir tahmin yapılmaktadır. Verici, öngörülen değer ile gerçek örnek değeri arasındaki farkı göndermekte, alıcı ise bu farkı yapılan öngörüye ekleyerek bilgi sinyalini geri oluşturmaktadır (Jayant, 1974). İyi bir öngörü yapıldığında, öngörülen ile gerçek değer arasındaki fark çok küçük olacağı için iletilen veri miktarında bir sıkıştırma yapılabilir. Şekil 4.5’de gösterilen DPCM alıcı ve verici yapısında, öngörücü (P), geçmiş örnek değerlerine bağlı olarak o anki örnek değeri için bir P_n öngörüsü yapmaktadır.



(a)



(b)

Şekil 4.5. DPCM alıcı (a) ve verici (b) yapısı

Uygulamalarda, nicemleme adım aralıkları çok küçük olduğunda öngörücü için x'_n yerine x_n kullanılabilir. Bu durumda p_n , girişin eski değerleri cinsinden ifade edilebilmektedir.

$$p_n = \sum_{i=1}^N a_i x_{n-i} \quad (4.5)$$

Burada n , öngörücünün derecesini göstermektedir. Öngörücü tasarımı için a_i katsayılarının bulunması yeterli olmaktadır. Bu katsayıların bulunması için kullanılan en yaygın yaklaşım, işaretin gerçek örnek değerleri ile öngörülen değerler arasındaki ortalama karesel hatanın en küçüklenmesidir. İşaretin gerçek ve örnek değerleri ile öngörü değerleri arasındaki ortalama karesel hata;

$$okh = E \left[\left(x_n - \sum_{i=1}^N a_i x_{n-i} \right)^2 \right], \quad (4.6)$$

olarak tanımlanır. Ortalama karesel hatayı en küçüklemek için öngörücü katsayılarına göre türev alınarak sonuç sıfıra eşitlenirse;

$$\frac{\partial OKH}{\partial a_1} = 2E \left[\left(x_n - \sum_{i=1}^N a_i x_{n-i} \right) x_{n-1} \right] = 0, \quad (4.7)$$

$$\frac{\partial OKH}{\partial a_2} = 2E \left[\left(x_n - \sum_{i=1}^N a_i x_{n-i} \right) x_{n-2} \right] = 0, \quad (4.8)$$

$$\frac{\partial OKH}{\partial a_N} = 2E \left[\left(x_n - \sum_{i=1}^N a_i x_{n-i} \right) x_{n-N} \right] = 0, \quad (4.9)$$

elde edilmektedir. Daha sade bir gösterim için, x_n işaretinin $R_{xx}(k) = E[x_n x_{n+k}]$ şeklinde tanımlanan özilinti fonksiyonu kullanıldığında, denklemler aşağıdaki şekli alırlar:

$$\sum_{i=1}^N a_i R_{xx}(i-1) = R_{xx}(1), \quad (4.10)$$

$$\sum_{i=1}^N a_i R_{xx}(i-2) = R_{xx}(2), \quad (4.11)$$

$$\sum_{i=1}^N a_i R_{xx}(i-N) = R_{xx}(N), \quad (4.12)$$

matris gösterimi için;

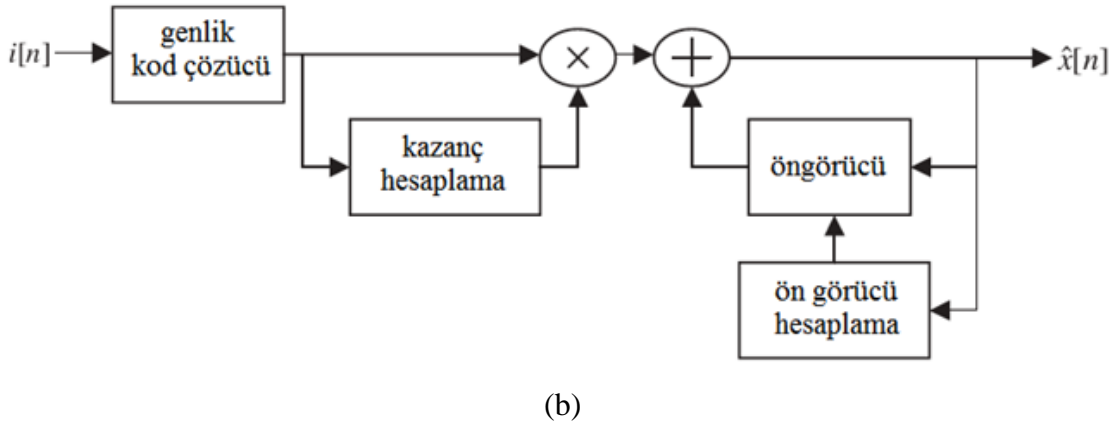
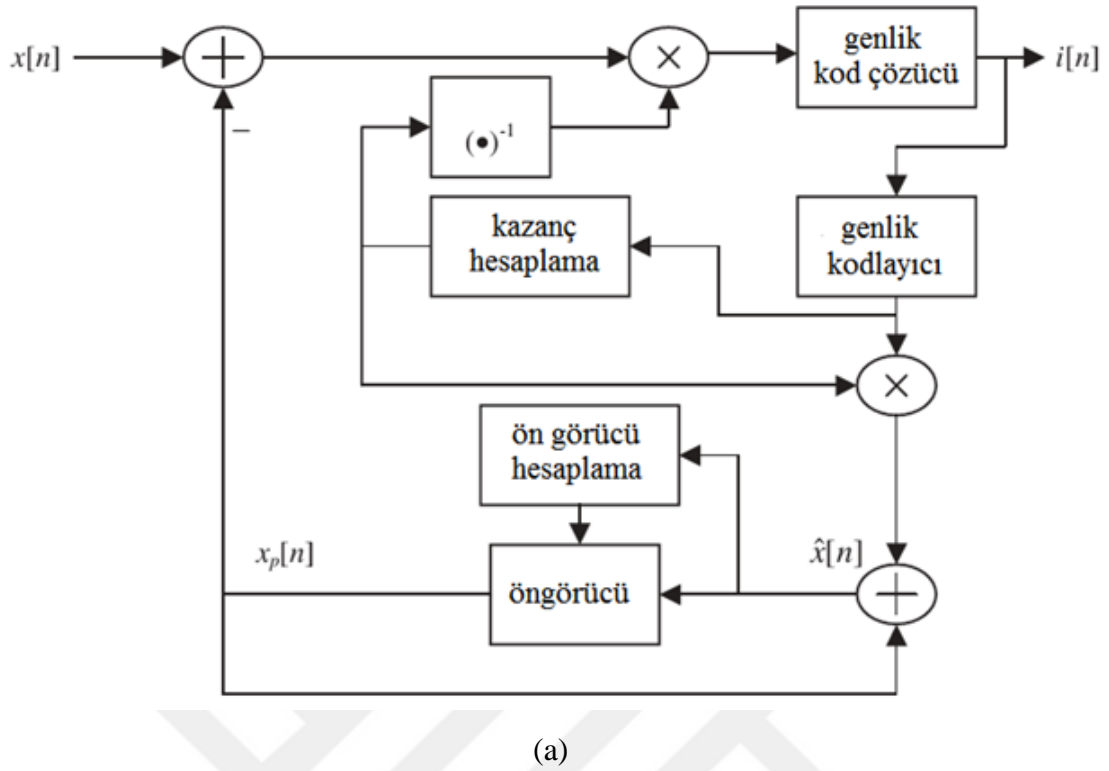
$$\mathbf{R} = \begin{bmatrix} R_{xx}(0) & R_{xx}(1) & R_{xx}(2) & \dots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & R_{xx}(1) & \dots & R_{xx}(N-2) \\ R_{xx}(2) & R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(N-3) \\ \dots & \dots & \dots & \dots & \dots \\ R_{xx}(N-1) & R_{xx}(N-2) & \dots & \dots & R_{xx}(0) \end{bmatrix}, \quad (4.13)$$

$$\mathbf{A} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ \dots \\ a_N \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} R_{xx}(1) \\ R_{xx}(2) \\ R_{xx}(3) \\ R_{xx}(4) \\ \dots \\ R_{xx}(N) \end{bmatrix}, \quad (4.14)$$

olarak tanımlandığında en küçük ortalama karesel hata için, $\mathbf{RA} = \mathbf{P}$ eşitliğinin sağlanması gerektiğinden, öngörücü katsayıları $\mathbf{A} = \mathbf{R}^{-1}\mathbf{P}$ olarak bulunmaktadır.

4.2.3 Uyarlanır farksal darbe kod modülasyonu (ADPCM)

DPCM, sabit bir öngörücü ile nicemleyici kullanır. Buna karşın, giriş sinyalinin değişimini takip eden uyarlanır bir sistem tasarlanabilir. Uyarlama, öngörücüye, nicemleyiciye ya da her ikisine birden uygulanabilir. Sonuçta oluşturulan sistem uyarlanır farksal darbe kod modülasyonu (ADPCM) olarak adlandırılır (Kondoç, 2005). ADPCM için ITU tarafından 4 farklı bit oranında (40, 32, 24 ve 16kbps) kodlama standartları geliştirilmiştir. Bu kodlayıcı G726 olarak da adlandırılır ve geriye doğru uyarlamalı kodlayıcı ve kod çözücü kullanır. DPCM ile benzer olarak, giriş sinyalinden tahmin edilen sinyal çıkarılarak tahmini hata sinyali bulunur, normalize edilir, nicemlenir ve kod çözücüye gönderilir. Kazancı hesaplamak için normalize edilip nicemlenmiş hata sinyali kullanılır. Nicemlenmiş giriş sinyallerini üretmek için öngörülen hata örnekleri, öngörülen sinyaller ile toplanır. Bulunan kazanç değeri, nicemlenmiş öngörülen hata örneklerinin de-normalizasyonunda kullanılır. Öngörücü, nicemlenmiş giriş sinyali $\hat{x}[n]$ 'den bulunur. Öngörücü için doğrusal tahminci hesaplama tekniği kullanılır. ADPCM için kodlayıcı ve kod çözücü kısımlar Şekil 4.6'da verilmiştir.



Şekil 4.6. Geriye doğru uyarlamalı ADPCM için kodlayıcı (a) ve kod çözücü (b)

4.2.4 Doğrusal öngörü kodlayıcısı (LPC)

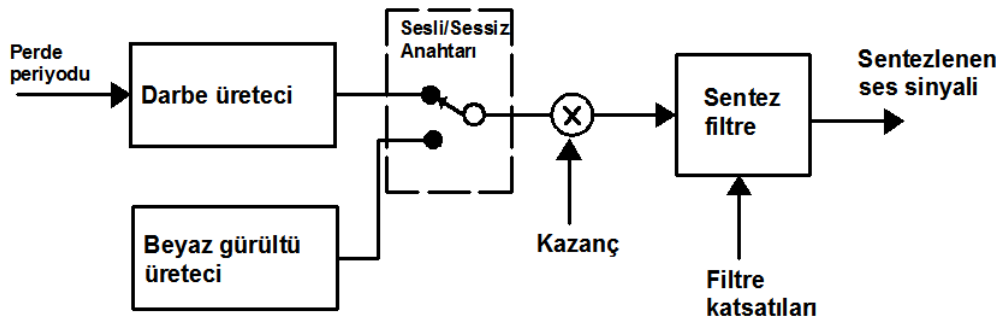
LPC, genel olarak 2.4 kbps oranında kullanılan bir ses kodlama yöntemidir ve sesin periyodik olarak veya rasgele gürültü ile uyarılan, doğrusal ve zamana göre değişen bir sistemin çıktısı ile modellenebileceği prensibine dayanır. Bu kodlayıcıdaki temel düşünce, bir ses örneğinin, geçmiş örneklerin doğrusal bir birleşimi (kombinasyonu) olarak yaklaşık elde edilmesine dayanır. LPC kodlayıcıda, sesin analiz edilip parametrelerinin çıkartılması ve işlenmesi gerekir. Bir ses çerçevesinin LPC ile kodlanması için her çerçeve

uzunluğunda, çerçevenin sesliliği, periyodikliği (perde periyodu), kazanç değeri ve model filtre parametreleri (doğrusal öngörü katsayıları) hakkında bilgiye ihtiyaç vardır. Bunu gerçekleştirmek için kısa süreli LPC modellemesi yapılacak ses sinyali, çerçevelere bölünür ve her çerçeve için ayrı ayrı doğrusal öngörü katsayıları hesaplanır. Burada perde periyodu, ses sinyalindeki sıralı uyarım darbeleri arasındaki zaman uzunluğunu; kazanç değeri, çerçevenin enerji seviyesini; seslilik bilgisi, çerçevenin sesli ya da sessiz olmasını, filtre katsayıları ise çerçeve için bulunan doğrusal öngörü katsayılarını belirtmektedir. Doğrusal Öngörü Kodlama yöntemi ile ses analizi ve sentezi, aşağıdaki sistem fonksiyonuna sahip ses izleme IIR filtresinin modellenmesidir.

$$H(z) = \frac{G}{1 + \sum_{i=1}^p a_i z^{-i}} \quad (4.15)$$

Burada a_i , i 'inci doğrusal öngörü katsayılarını; p , filtrenin derecesini, G ise filtre kazancını göstermektedir.

Kod çözücü kısımda, sesli ve sessiz sesleri oluşturmak için iki temel uyarma fonksiyonu vardır. Kısa bir zaman aralığında sesli sesler temel bir F_0 frekansı ile periyodiktir. Bu temel frekans konuşana bağlıdır. Bu durumda IIR filtresi, elde edilmek istenilen sesli sesin ana frekansı ile aynı frekansta bir periyodik darbe serisi ile uyarılmalıdır. Sessiz sesler ise IIR filtresini beyaz gürültü kaynağı ile uyararak oluşturulur. Bu anlatım Şekil 4.7'de blok diagramı verilen şemada da gösterilmektedir.



Şekil 4.7. Ses üretimi için LPC modeli

Biraz daha özelleştirmek gerekirse, çerçeveler için bahsi geçen kısa aralıklar, örnekleme frekansı 8 Khz olan bir sinyal için yaklaşık 20 ms'dir. Bir başka deyişle, yaklaşık 160 örnektir. Kodlayıcının çıkışından kodlanarak gönderilen ses işareti, alıcıda kodu çözülerek, modelden ve uyarma fonksiyonundan tekrar elde edilebilir. Doğrusal Öngörü yöntemi kullanılarak, ses örneklerinden filtrenin parametreleri kolaylıkla bulunabilir. FIR doğrusal öngörü filtresinin çıkışları :

$$\hat{s}[n] = -\sum_{i=1}^p a_i s[n-i], \quad (4.16)$$

ile bulunabilir. Tahmin edilen ve gözlenen örnek arasındaki hata ise;

$$e[n] = s[n] - \hat{s}[n], \quad (4.17)$$

ile bulunur. Hataların karelerinin beklenen değeri;

$$J = E\{e[n]^2\} = E\left\{\left(s[n] + \sum_{i=1}^p a_i s[n-i]\right)^2\right\}, \quad (4.18)$$

olur. Buradan kutup parametrelerini rahatlıkla bulabiliriz. Bunun için J 'nin her parametreye (a_i) göre türevinin sıfıra eşitlenmesi sonucunda ortaya p elemandan oluşan bir doğrusal denklem kümesi çıkar:

$$\frac{\partial J}{\partial a_k} = 2E\left\{\left(s[n] + \sum_{i=1}^p a_i s[n-i]\right)s[n-k]\right\} = 0. \quad (4.19)$$

Burada $k=1,2,\dots,p$ değerlerini almaktadır. Denklem 4.19 yeniden düzenlenirse;

$$E\{s[n]s[n-k]\} + \sum_{i=1}^p a_i E\{s[n-i]s[n-k]\} = 0, \quad (4.20)$$

elde edilir. Denklem otokorelasyon matrisi ile aşağıdaki gibi yazılabilir;

$$\sum_{i=1}^p a_i R_s [i-k] = -R_s [k] . \quad (4.21)$$

Burada $R_s [i-k] = E \{s[n-i]s[n-k]\}$ ve $R_s [k] = E \{s[n]s[n-k]\}$ değerlerine eşittir. Denklem 4.21'deki ifade matris formunda aşağıdaki gibi ifade edilebilir;

$$\mathbf{R}_s \mathbf{a} = -\mathbf{r}_s . \quad (4.22)$$

Burada \mathbf{R}_s , $s[n]$ sinyalinin otokorelasyon matrisidir ve aşağıdaki gibi gösterilir.

$$\mathbf{R}_s = \begin{bmatrix} R_s [0] & R_s [1] & \dots & R_s [p-1] \\ R_s [1] & R_s [0] & \dots & R_s [p-2] \\ \vdots & \vdots & \ddots & \vdots \\ R_s [p-1] & R_s [p-2] & \dots & R_s [0] \end{bmatrix} . \quad (4.23)$$

Doğrusal öngörü katsayıları;

$$\mathbf{a} = [a_1, a_2, \dots, a_p]^T , \quad (4.24)$$

ve

$$\mathbf{r}_s = [R_s [1], R_s [2], \dots, R_s [p]]^T , \quad (4.25)$$

olmaktadır. Doğrusal öngörü katsayıları için aşağıdaki denklem elde edilir.

$$\mathbf{a} = -\mathbf{R}_s^{-1} \mathbf{r}_s . \quad (4.26)$$

Doğrusal öngörü katsayıları, otokorelasyon matrisinin tersi alınarak ya da Levinson-Durbin algoritması ile çözülebilir. Öngörü kazancı, giriş sinyali $s[n]$ ile hata sinyali $e[n]$ 'in varyanslarının oranının desibel cinsinden değeridir ve aşağıdaki gibi hesaplanır.

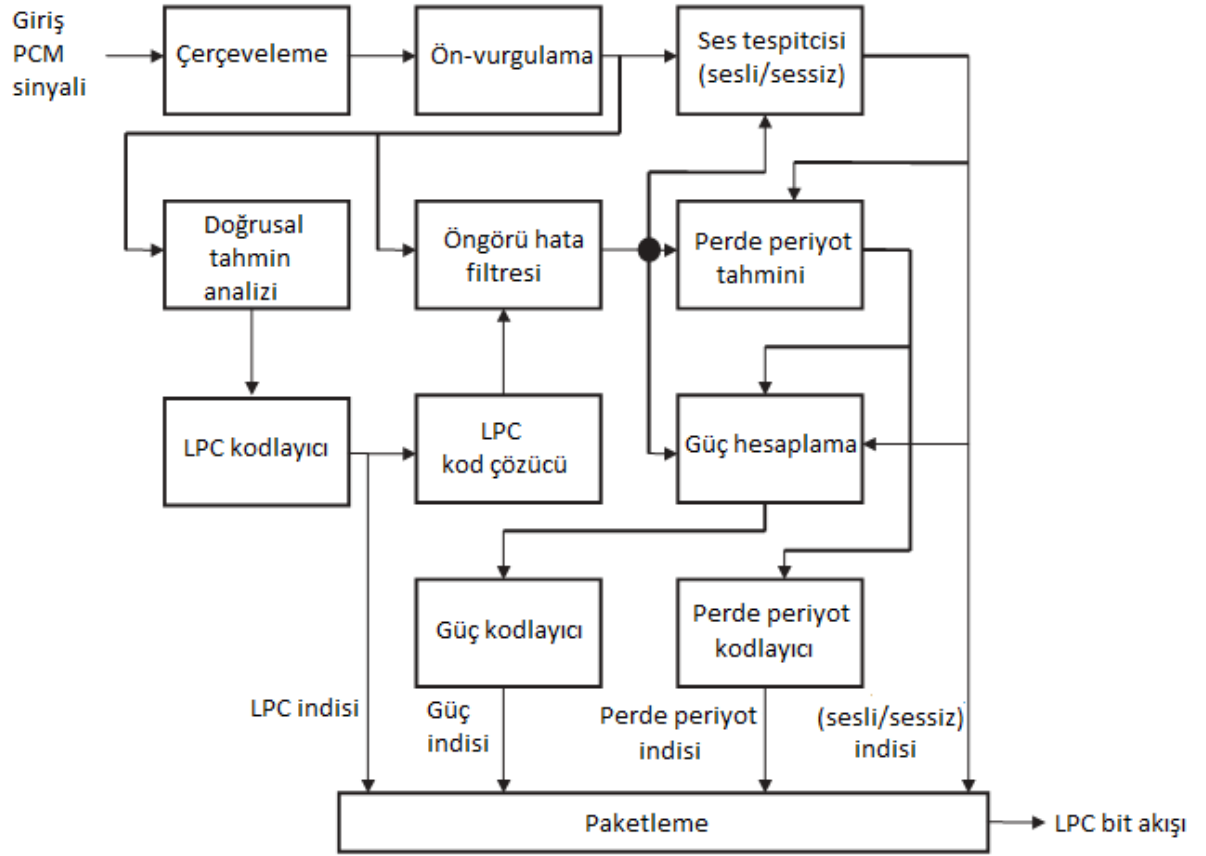
$$PG = 10 \log_{10} \left(\frac{\sigma_s^2}{\sigma_e^2} \right) = 10 \log_{10} \left(\frac{E\{s^2[n]\}}{E\{e^2[n]\}} \right). \quad (4.27)$$

Yüksek kazanç değeri iyi bir tahmin yapıldığını gösterir. LPC kodlayıcının ilk çeşidi, 2.4kbps bit oranına sahip FS-1015 olarak standartlaştırılmıştır. Bu kodlayıcı, 8 kHz örnekleme oranı için 22.5 ms (130 örnek) çerçeve uzunluğu kullanmaktadır. Her çerçeve için (sesli ya da sessiz) 54 bit kullanılmaktadır (Kondo, 2007). Bu kodlayıcı için bit ataması, Çizelge 4.2’de verilmiştir.

Çizelge 4.2. FS1015 kodlayıcı için bit ataması

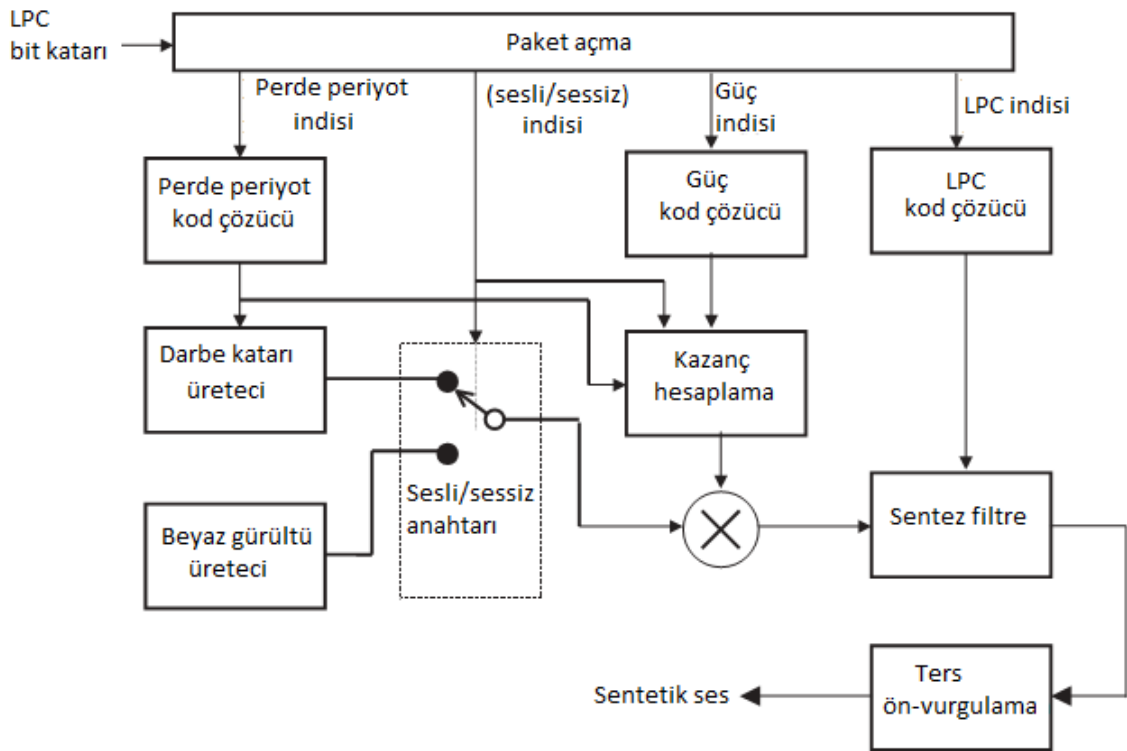
Parametreler	Sesli	Sessiz
Perde periyodu/Sesli-sessiz	7	7
Güç	5	5
LPC	41	20
Senkronizasyon	1	1
Hata koruma	-	21
Toplam	54	54

Şekil 4.8’de LPC için kullanılan kodlayıcı şeması verilmiştir (Kondo, 2007). Bu şemada giriş ses sinyali, çerçeveler üst üste bindirilmeden çerçevelenir. Giriş sinyalinin spektrumunu ayarlamak için ön-vurgu filtresi uygulanır. Ses tespitçisi, sesin sesli-sessiz ayırımını yapar. Ön-vurgulanmış sinyal, doğrusal öngörü analizi için kullanılır ve bu kısımda 10 adet doğrusal öngörü katsayısı elde edilir. Bu katsayılar nicemlenerek indisleri kod çözücüye gönderilir. Nicemlenmiş bu katsayılar, öngörü-hata filtresini oluşturmada kullanılır. Bu filtre, ön-vurgulanmış sinyale uygulanarak filtre çıkışında öngörü-hata sinyali elde edilir. Eğer çerçeve sinyali sesli ise perde periyodu, öngörü-hata sinyali kullanılarak tespit edilir. Ayrıca, sesli çerçeve için öngörü-hata sinyalinin ($e[n]$) gücü, perde periyodunun tam sayı değeri kullanılarak bulunur. Sesli/sessiz biti ile beraber, LPC indislerine, perde periyot indislerine ve güç indislerine ait bit katarı paketlenerek kod çözücüye gönderilir.



Şekil 4.8. LPC kodlayıcı şeması

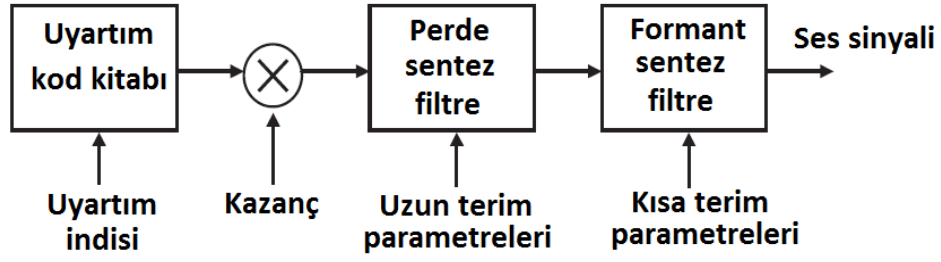
Kod çözücü kısımda, alınan bit katarı içerisinde perde periyodu indisi, LPC indisleri, sesli/sessiz biti ve güç indisleri ayrıştırılarak ilgili değerler tespit edilir. Eğer çerçeve sinyali sesli ise, perde periyodu büyüklüğünde darbe katarı oluşturulur. Bu darbe katarı, kazanç değeri ile çarpılarak doğrusal öngörü katsayılarından oluşturulan ses filtreden geçirilir ve son olarak ters ön-vurgulamadan sonra ses sentezlenir. Sessiz çerçeve içinde benzer adımlar uygulanır ancak darbe katarı üretmek yerine beyaz gürültü üretilir. Şekil 4.9’da LPC için kullanılan kod çözücü şeması verilmiştir.



Şekil 4.9. LPC kod çözücü şeması

4.2.5 Kod uyarımlı doğrusal öngörü kodlayıcı (CELP)

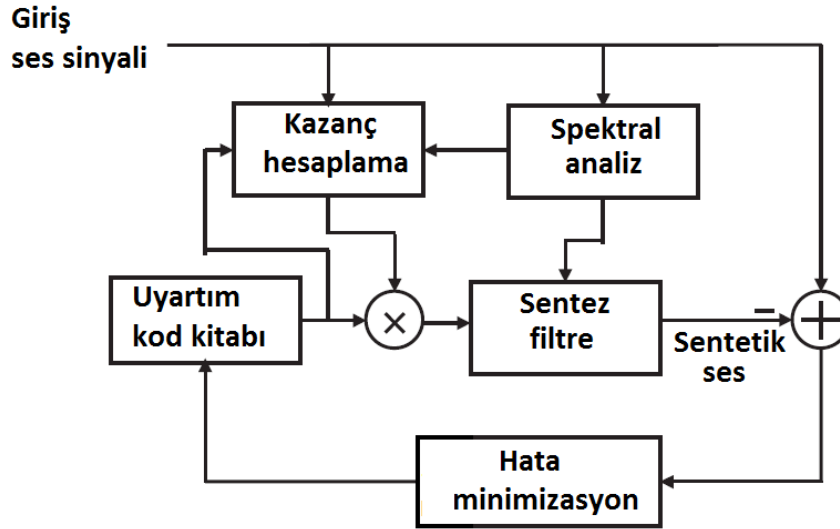
CELP, LPC kodlayıcının iyileştirilmiş hali olarak dizayn edilmiştir (Devalapalli vd., 2003). Temel düşünce olarak, ses sinyalinin (kısa süreli) formant yapısı ve (uzun süreli) perde periyodu bilgisine dayanarak ses sentezleme gerçekleştirmektedir (Schroeder ve Atal,1985). Sesin kısa süreli formant yapısı için 10. dereceden bir LP sentez filtresi kullanır. Ses sinyalinin periyodikliği ise uyarlamalı bir kod kitabı ile modellenir. Böylece, LPC'deki sesli-sessiz sınıflama işlemi olmaksızın bir uyarım kod kitabı yardımı ile sesler sentezlenebilmektedir. Bu kodlayıcıya, uyarım kod kitabı isminden kaynaklı olarak, 'code' ve 'excited' kelimeleri birleştirilerek bu isim verilmiştir (Kondoz, 2007). CELP, ses kodlama için en etkili fikirlerden biri olarak bilinir. Bu kodlayıcının prensipleri, pek çok standartlaştırılmış ses kodlayıcının temelini oluşturur. CELP ses üretim modeli, CELP kısa-süre ve uzun-süre doğrusal öngörü modelleri ile ilişkilidir. Şekil 4.10'daki blok diyagramda ses üretim modeli gösterilmiştir.



Şekil 4.10. CELP için ses üretim modeli

Şekil 4.10'da, uyartım kod kitabından seçilen kod vektör ile kazanç katsayısının çarpılıp, seri bağlı perde ve formant sentez filtrelerinden geçirilerek sentetik ses sinyalinin elde edilmesi gösterilmiştir. Burada, perde sentez filtre, bir perde frekansına göre sinyalde periyodiklik sağlar. Formant sentez filtre ise sinyalin spektral zarfını üretir. Uyartım kod kitabı sabit (fixed) veya uyarlamalı (adaptive) olabilir ve rastgele gürültü ya da darbe katarları içerebilir. Bir ses kodlayıcıda ses sinyali; kazanç, filtre katsayıları, ses uzunlukları gibi parametreler ile temsil edilir. Sistem açık döngü (open loop) ise parametreler giriş sinyalinden elde edilir, nicemleme yapılır ve sonra sentezleme için bu değerler kullanılır. Örnek olarak LPC, MELP, RPE-LTP böyle çalışır. Sinyali sentezlemek için parametreleri kullanmanın daha etkili bir yolu, parametreleri kodlama devam ederken bulmaktır. Bu işlemler kapalı- döngü (closed-loop) olarak bilinir. Amaç, sentezlenmiş sinyal ile orijinal sinyal arasındaki hatayı en küçük yapacak en iyi parametrelerin seçilmesidir.

Sinyal, analiz amacı ile kodlama boyunca sentezlendiği için bu prensip, analiz-sentez olarak bilinir. Şekil 4.11'de bir uyartım sinyali, kapalı-döngü işlemlerine göre seçilir ve sentez filtrelere uygulanır. Sentezlenmiş sinyal ile orijinal sinyal arasındaki hata, hataların karelerinin toplamı şeklinde bulunur ve bu işlem kod kitabındaki uyartım kod vektörlerinin hepsi için tekrar edilir.



Şekil 4.11. CELP kodlayıcının temel bileşenleri

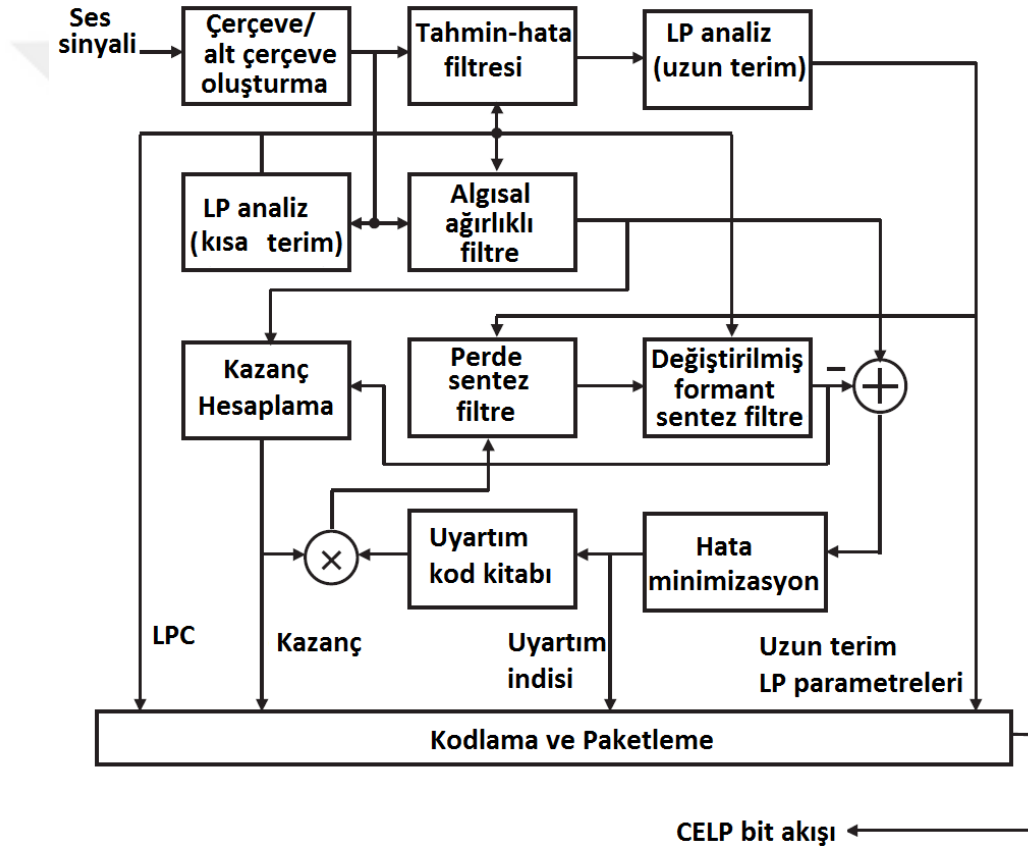
Şekil 4.11'deki formant sentez filtrenin sistem fonksiyonu;

$$H_f(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{i=1}^M a_i z^{-i}} \quad (4.28)$$

Burada $A(z)$, formant analiz filtrenin sistem fonksiyonudur. Kod kitabındaki kod vektörlerinin boyutu, ses çerçevelerinin uzunluğuna bağlıdır. Örneğin FS1016 kodlayıcı için bir alt çerçeve 60 örnekten oluşmaktadır. Bu yüzden uyartım kod kitabı 512x60 boyutlu bir matristen oluşmaktadır. Ses sinyalindeki yüksek enerjili bölgelerdeki gürültü bileşenleri, düşük enerjili bölgelerdeki gürültülere göre daha yüksek enerjiye sahiptir. Diğer ifade ile ses sinyalinin yüksek enerjiye sahip frekans bölgeleri maskelenebilir ve bu bölgeler daha yüksek gürültü oranına sahiptir. Buna karşın düşük enerjili frekans bölgelerdeki gürültü miktarı oransal olarak daha düşüktür. Bu yüzden tipik bir spektrumda, tepe değerlerdeki gürültü miktarı vadi bölgelerdeki gürültü miktarından daha yüksektir. Gürültü spektrumunu kontrol etmenin basit bir yolu hata sinyalini en küçükmeden (minimize etmeden) önce aşağıdaki transfer fonksiyonuna sahip ağırlıklı bir filtreden geçirmektir:

$$W(z) = \frac{A(z)}{A(z/\gamma)} = \frac{1 + \sum_i^M a_i z^{-i}}{1 + \sum_i^M a_i \gamma z^{-i}} \quad (4.29)$$

Burada γ , bir sabittir ve $[0-1]$ aralığındadır. Filtre, ses spektrumunun formant olmayan bölgelerdeki hata sinyal spektrumunu yükseltirken, formant bölgelerdeki hata sinyalinin spektrumunu sönümlendirir. Bir CELP kodlayıcının genel yapısı basitleştirilmiş bir biçimde Şekil 4.12’de verilmiştir.



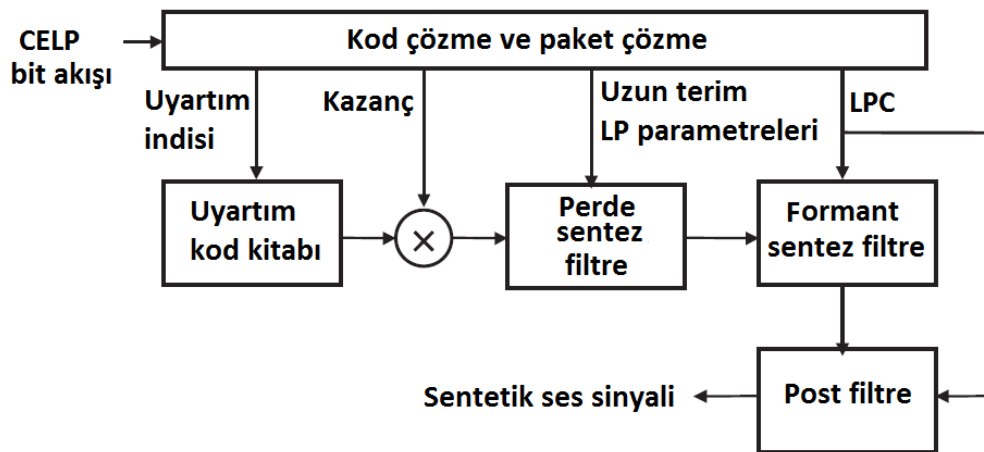
Şekil 4.12. CELP kodlayıcı yapısı

Bu kodlayıcı aşağıdaki gibi çalışır.

- Giriş sinyali çerçevelere bölünür. Ardından bu çerçeve, alt çerçevelere bölünür. Genel olarak çerçeve 4 alt çerçeveye bölünür. Çerçeve uzunluğu 20 ya da 30 ms olarak alınır. Bu durumda her alt çerçeve 5 ya da 7.5 ms uzunluğundadır.

- LPC katsayılarını elde etmek için her bir çerçeveye, kısa-sürelî LP analizi uygulanır. Kısa-sürelî LP analizi; formant frekansların bulunmasını sağlar. Diğer deyişle, ile çerçevenin sinyal spektrumunu oluşturan LPC katsayıları bulunur. Burada, kısa-sürelî LP analizinin girişi, orijinal çerçeve sinyalidir. Sonra her bir alt çerçeveye, uzun- süreli LP analizi uygulanarak perde frekansları bulunur. Uzun-sürelî LP analizi için giriş, tahmini hatadır. Bulunan parametreler ile, algısal ağırlıklı filtre, perde sentez filtre ve deęiştirilmiş formant sentez filtre oluşturularak gerekli sentezleme işlemi gerçekleştirilir.
- Bu aşamadan sonra sentezleme işlemi için ilk önce bir uyarım vektörü bulunmalıdır. Her bir alt çerçeve için bulunan uyarım vektörü hatayı en az veren kazanç deęerine göre seçilir.
- Uyarım vektörüne karşılık gelen indis, kazanç, uzun-terim LP katsayıları ve LPC katsayıları kodlanır, paketlenir ve CELP bit akışı olarak kod çözücüye gönderilir.

Kod çözücü işlemleri için kullanılan CELP kod çözücünün akış diyagramı Şekil 4.13'te verilmiştir. Burada temel olarak, bit paketi çözülür ve bit akışındaki çeşitli parametreler kullanılarak ses sinyali sentezlenir. Sentezlenen ses sinyalinin kalitesini artırmak için çıkışa bir (son) filtre konulur.

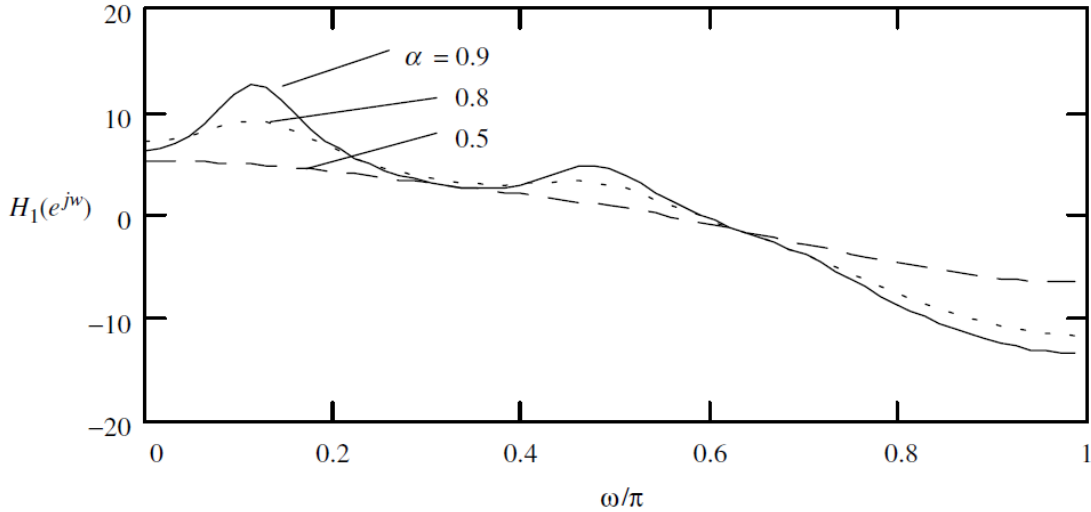


Şekil 4.13. CELP kod çözücünün akış diyagramı

Ağırlıklandırılmış filtre $w(z)$ ile sentetik ses sinyallerinin spektral vadi kısımlarındaki gürültü bileşenleri azaltılır. Ses kalitesini daha da artırmak için spektral vadi kısımlarını sönümlendiren bir filtre (son filtre) kullanılır. Böylece sentezlenen sinyalin tüm spektrumu iyileştirilir. Kısa-sürelili son filtre aşağıdaki gibi bir (LP sentez filtre) frekans cevabına sahiptir.

$$H(z) = \frac{1}{1 + \sum_i^M a_i \alpha z^{-i}} \quad (4.30)$$

Burada α , $0 < \alpha < 1$ değer aralığındaki bir sabittir. Farklı α değerleri için bir son filtrenin genlik cevapları Şekil 4.14'te verilmiştir.



Şekil 4.14. Farklı α değerleri için bir son filtrenin genlik cevapları

4.2.6 Cebirsel CELP kodlayıcı (Algebraic-CELP)

Bu kodlayıcı çeşidi, standart CELP kodlayıcının hesaplama gecikmesini azaltmak için geliştirilmiştir. Burada cebirsel tabiri, uyartım kod vektörlerini oluşturmak için kullanılan basit matematiksel işlemleri belirtmek için kullanılmıştır. Bu cebirsel CELP yaklaşımının temeli bize, tüm kod kitabını fiziksel olarak depolamamıza gerek olmadığını göstermiştir. Aşağıdaki 5 adet standartlaştırılmış kodlayıcı ACELP temellidir.

- ITU-T G723.1 Multiple Prediction Maximum Likelihood Quantization (MP-MLQ) (1995)
- ITU-T G729 Conjugate Structure CELP (CS-CELP) (1995)
- TIA-IS641 ACELP (1996)
- ETSI GSM Enhanced Full Rate (EFR) ACELP (1996)
- ETSI Adaptive Multirate (AMR) ACELP (1999)

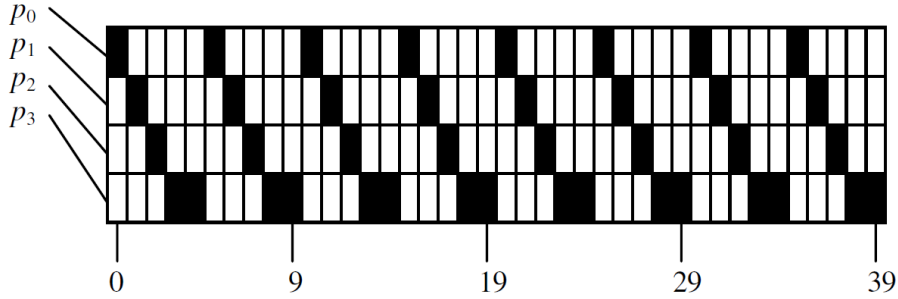
4.2.7 Eşlenik yapılı CELP kodlayıcı (Conjugate structured - CELP)

Bu ses kodlama algoritması, ITU-T tarafından G729 olarak adlandırılır. Bu kodlayıcı, kişisel iletişim sistemlerinde ve dijital uydu haberleşme alanlarında sıklıkla kullanılır. Kodlayıcı girişine, 8kHz'de örneklenmiş 16 bit PCM ses sinyali uygulanır. Her çerçeve, 10ms'lik toplam 80 örnekten oluşur. Bu kodlayıcının CELP parametreleri, sabit (fixed), adaptif kod kitabı indisleri ve kazançları ile LPC filtre katsayılarını içerir. Kod çözücü, bu parametrelere ait değerleri kullanarak sentezlenmiş ses sinyalini üretir (Kondoz, 2007). Ses sinyaline önce 140 Hz'lik bir yüksek geçiren filtre uygulanır. 10 ms'lik çerçeveler için LP katsayıları bulunur. Bu LP katsayıları, LSP (Line Spectral Pair) değerlerine çevrilir. Daha sonra bu değerler nicemlenir ve nicemlenen değerlere toplam 18 bit atama yapılır. Her çerçeve 2 alt çerçeveye bölünür. Bu durumda 5ms'lik (40 örnekli) iki alt çerçeve oluşur. İlk alt çerçevenin LP katsayıları sentez filtrede kullanılır. İkinci alt çerçeve için sentez filtre, nicemlenmiş ve orijinal LP katsayılarından oluşturulur. Her bir alt çerçeve için en iyi CELP parametrelerini sağlayacak şekilde analiz-sentez gerçekleştirilir. B kodlayıcı adaptif ve sabit kod kitaplarına sahiptir. Adaptif kod kitabı, uygun perde (pitch) periyodunun bulunması için kullanılır (sesli, ötümlü sinyaller). Sabit kod kitabı ise sessiz (ötümsüz) sinyallerin sentezlenebilmesi için oluşturulmuştur.

Standart CELP kodlayıcı, sabit bir uyarım kod kitabı kullanır. Kod kitabı, kodlama süresince taranarak en uygun kod vektörü her alt çerçeve için seçilir. G729 kodlayıcı için sabit (fixed) kod kitabı, cebirsel yapıda oluşturulmuştur. Sabit kod kitabındaki bir kod vektörü, her biri 4 adet darbe sinyali vektörünün toplamı olarak aşağıdaki gibi ifade edilebilir:

$$v[n] = \sum_{i=0}^3 s_i \delta[n - m_i]; \quad n = 0, \dots, 39. \quad (3.31)$$

Burada bir kod vektörü ($v[n]$), 4 adet darbe sinyalinin oluşan 40 boyutlu bir vektördür. Cebirsel (sabit) kod kitabının grafiksel gösterimi Şekil 4.15'te verilmiştir.



Şekil 4.15. Cebirsel (sabit) kod kitabının grafiksel gösterimi

Her uyarım kod vektörü, bir alt çerçeve uzunluğu olan 40 örneğe sahiptir. Uyarım kod vektörü ($v[n]$) için $s_i = \pm 1$, darbe işaretini ve m_i , darbe işaretinin konumunu gösterir. Çizelge 4.3'de 4 adet uyarım vektörü için darbe konumları verilmiştir.

Çizelge 4.3. Uyarım vektörü için darbe konumları

Darbe No	İşaretler	Pozisyonlar	Bitler
i_0	$s_0 : \pm 1$	0, 5, 10,15,20,25, 30,35	1+3
i_1	$s_1 : \pm 1$	1, 6, 11,16,21,26, 31,36	1+3
i_2	$s_2 : \pm 1$	2, 7, 12,17,22,27, 32,37	1+3
i_3	$s_3 : \pm 1$	3, 8, 13,18,23,28, 33,38	1+4
		4,9,14,19,24,29,34,39	

Her darbe sinyalinin işareti için 1 bit gerekir. Darbe sinyalinin pozisyonu için (s_0, s_1, s_2) 3 bit gerekir. s_3 için ise 4 bit gereklidir. Böylece tüm kod kitabını indirmek için 17 bit gerekir. Adaptif kod kitabı gerçekte, adaptif perde filtresidir. Adaptif perde filtre denklem 4.32'deki gibi ifade edilir.

$$P(z) = \frac{1}{1 - \beta z^{-T}}, \quad (4.32)$$

burada T , perde periyodu ve β filtre kazancıdır. G729 kodlayıcı, FS1016 veya IS54 gibi CELP kodlayıcılar ile karşılaştırıldığında adaptif (uyarlamalı) kod kitabına daha karmaşık metotların uygulandığı görülmektedir. Ancak bu metotlar ile daha kaliteli ses sinyalleri sentezlenir. DSP teknolojisinin gelişmesinden faydalanarak, kullandığı bu karmaşık metotların hesaplama sürelerini telafi edebilmektedir. İlk alt çerçeve için perde periyoduna 8 bit ve ikinci alt çerçeve için 5 bit atanır. G729 kodlayıcı için bit ataması aşağıdaki Çizelge 4.4'te verilmiştir.

Çizelge 4.4. G729 kodlayıcı için bit ataması

Parametreler	Kod kelimeleri	Alt çerçeve	Alt çerçeve	Bit/çerçeve
LSF	L0,L1,L2,L3	1	1	18
Uyarlamalı kod kitabı	P1,P2	8	5	13
Perde parite biti	P0	1		1
Sabit kod kitabı indisi	C1,C2	13	13	26
Sabit kod kitabı işareti	S1,S2	4	4	8
Kod kitabı kazançları (1. Etap)	GA1,GA2	3	3	6
Kod kitabı kazançları (2. Etap)	GB1,GB2	4	4	8
Toplam				80

4.2.8 Düşük gecikmeli CELP kodlayıcı (LD-CELP)

Düşük bit oranında (16kbps) düşük bir gecikme için kullanılmaktadır. LD-CELP'in oluşturulması için pek çok strateji geliştirilmiştir. Bunlardan ilki çerçeve uzunluğunun 20 örneğe indirilmesidir. Kullanılan çerçevenin uzunluğundan dolayı oluşan kodlama gecikmeleri, daha çok kodlayıcı da gerçekleştirilen tamponlamadan kaynaklanmaktadır. Eğer çerçeve uzunluğu büyükse kodlama gecikmeside o kadar artmaktadır. Bu yüzden bir

çözüm olarak, çerçeve uzunluğunun azaltılması kararlaştırılmıştır. Ayrıca, geleneksel CELP gibi her çerçeve 4 alt çerçeveye bölünmüş ve her alt çerçeve sadece 5 örnekten oluşturulmuştur. İlk alt çerçevenin 5 örneği alınarak tamponlanır. Böylece tamponlama için sadece 0,625 ms gecikmesi ve 1,25 ms-1,875 ms arasında kodlama gecikmesi oluşmaktadır. Bu değerler, geleneksel CELP ile karşılaştırıldığında oldukça düşüktür. CELP için tamponlama gecikmesi 20 ms-30 ms arasındadır (Chen vd., 1992).

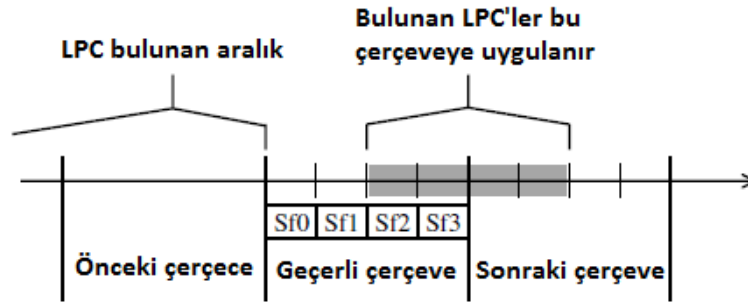
İkinci strateji ise, doğrusal öngörü katsayılarını bulmak için kullanılan otokorelasyon değerlerinin hesaplanması ile ilgilidir. Genelde kısa çerçeve uzunlukları için Hamming pencereleme metodu kullanıldığında hesaplama maliyeti yüksek olmaktadır. Bu yüzden doğrusal öngörü katsayılarının bulunması için Chen pencereleme kullanılır. Chen pencereleme, öz yinelemeli (recursive) ve öz yinelemesiz yaklaşımların birleşimi (kombinasyonu) şeklinde oluşturulmuş bir tekniktir. Düşük bit oranında (16kbps) düşük bir gecikme oluşturmak amacı ile geliştirilmiştir. Kodlama gecikmesi, kodlayıcı kısımdaki çerçevelerin uzunluğunun artması ile arttığından, çerçeve boyutunun azaltılması gecikmenin de azaltılmasını sağlamıştır. Geleneksel CELP kodlayıcı gibi LD-CELP’te giriş ses sinyallerini 20 örneklilik çerçevelere böler. Her çerçeveyi de 4 ayrı alt çerçeveye böler. Bu 0.625 ms’lik bir gecikmeye yol açar. Genel olarak kodlama gecikmesi 1.25 ms ile 1.875 ms arasındadır. Bu gecikme değerleri geleneksel CELP kodlayıcılar için 20 ms - 30 ms aralığındadır. LD-CELP’in kullandığı çerçeve uzunluğu çok kısa olduğu için, çerçevenin istatistiksel özellikleri yakın geçmiş ve gelecek değerleri ile yakından ilişkilidir. Geçerli çerçeve için LPC katsayılarının tamamı geçmiş değerlerden tahmin edilebilir. Böylece, kodlayıcı kısım 20 örneklilik çerçeveyi tamponlamaya (işlemeye) almadan, bir alt çerçeve kullanılarak kodlamaya başlayabilir.

Üçüncü strateji ise, geriye doğru uyarlamalı doğrusal öngörüdür. Geleneksel ses kodlayıcılar, ileri doğru uyarlamalı doğrusal öngörü gerçekleştirirler. LD-CELP ise sentetik (sentezlenmiş) ses sinyalinden LPC elde eder. Bundan dolayı, bu katsayıları nicemleyip kod çözücüye göndermesine gerek yoktur. Böylece, oldukça fazla sayıda bit göndermeye gerek yoktur. Ancak bu yöntemin bir dezavantajı kanal hatalarına açık olmasıdır. Eğer bir çerçevede hata oluşursa bu gelecek çerçevelere de etki etmektedir.

Dördüncü strateji, yüksek dereceden öngörü sentez filtre kullanmaktır. 50. dereceden bir öngörü sentez filtre ile sesli çerçevelerin periyodikliği bulunmaktadır. Filtrenin katsayıları, sentetik ses sinyali kullanılarak, geriye doğru uyarlamalı analiz sonucu elde edilir. Aynı işlemler kodlayıcı ve kod çözücünde gerçekleştirilir. Filtrenin sistem fonksiyonu;

$$H(z) = \frac{1}{1 + \sum_{i=1}^{50} a_i z^{-i}} \quad (4.33)$$

Burada doğrusal öngörü katsayıları (a_i), $i=1, \dots, 50$ geriye doğru uyarlama ile sentetik sessinyalinden bulunur. LPC'ler her 20 örnekli çerçevede güncellenmektedir. Bu güncelleme işlemi Şekil 4.16'da verilmiştir.



Şekil 4.16. LPC'lerin belli çerçeveler ile güncellenmesi

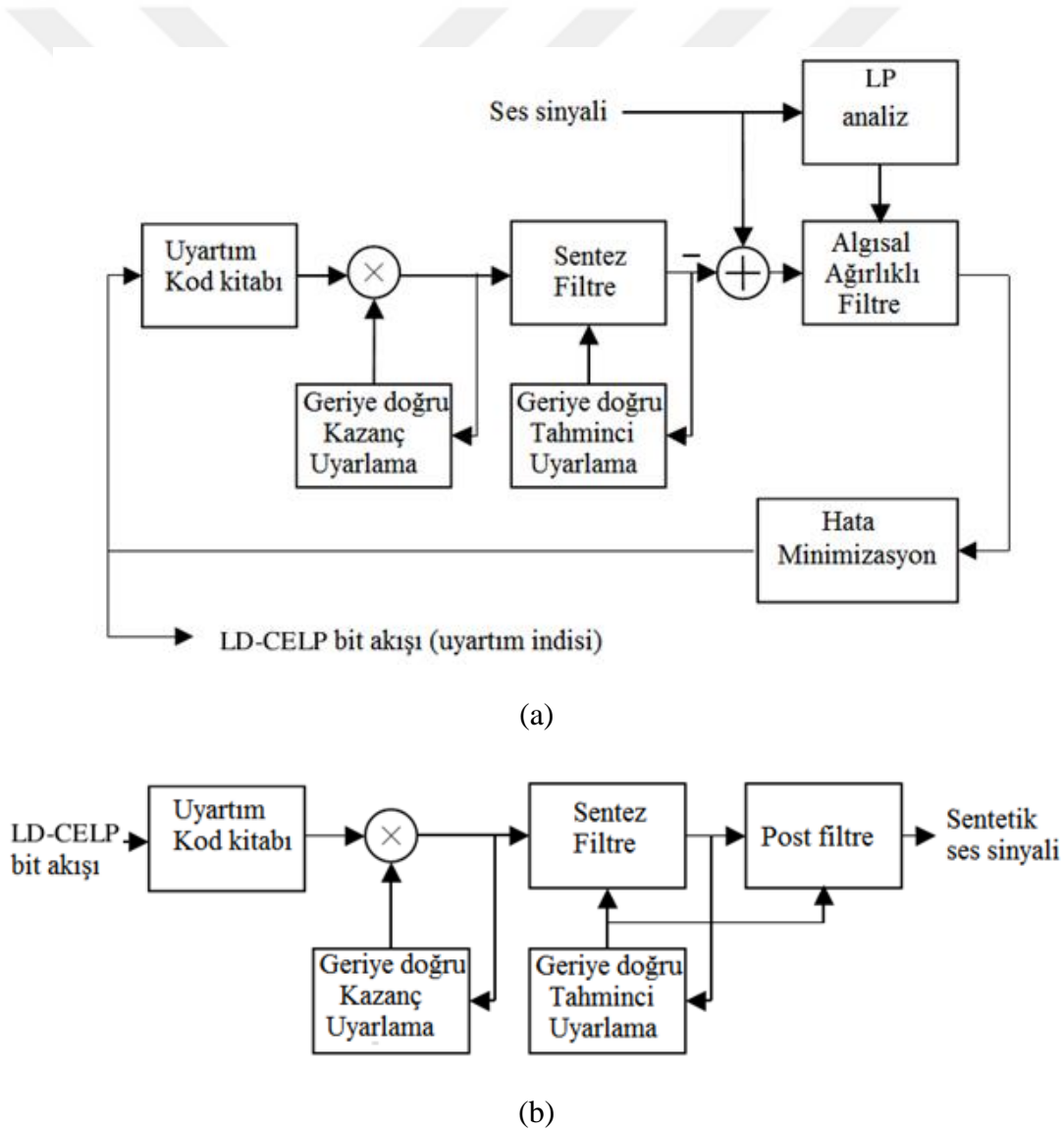
Doğrusal öngörü katsayıları Sf0 alt çerçeveden önceki çerçeveler kullanılarak bulunur. Bulunan katsayılar geçerli çerçevenin Sf2 ve Sf3 alt çerçeveleri ile sonraki çerçevenin Sf0 ve Sf1 alt çerçeveleri için kullanılır.

Beşinci stratejik yaklaşım ise, geriye doğru uyarım kazancının uyarlamasıdır. Uyarım kazancı, her alt çerçevede 10. dereceden uyarlamalı doğrusal öngörü filtresi kullanılarak, logaritmik kazanç düzleminde güncellenmektedir. Bu işlemin yapılması ile geçerli kazanç değeri, nicemleme yapılmış uyarım değerlerinden türetilmektedir. Böylece, uyarım kazancını belirtmek için kod çözücü kısma herhangi bir bit göndermeden, aynı kazanç değeri kod çözücünde elde edilmektedir. Bu işlemler kodlayıcı ve kod çözücünde aynı biçimde yapılmaktadır. Algısal ağırlıklı filtre 10 doğrusal öngörü katsayısına sahiptir. Bu

katsayılar orijinal ses verilerinden üretilir. Filtre katsayıları her çerçevede güncellenir. Geçerli çerçevenin katsayıları, önceki çerçevenin örneklerinden elde edilir. Algısal ağırlıklı filtrenin sistem fonksiyonu;

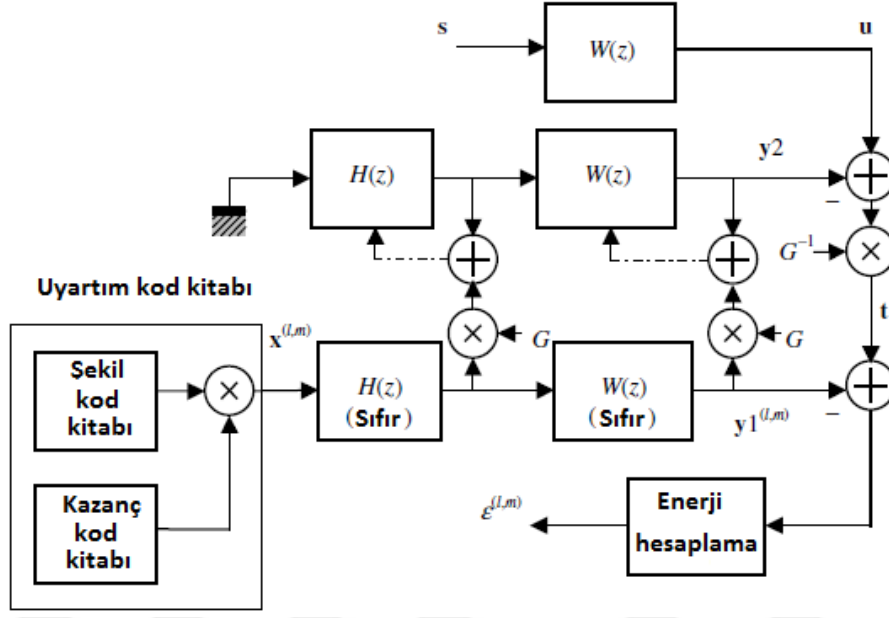
$$w(z) = \frac{A\left(\frac{z}{\gamma_1}\right)}{A\left(\frac{z}{\gamma_2}\right)} = \frac{1 + \sum_{i=1}^{10} b_i \gamma_1^i z^{-i}}{1 + \sum_{i=1}^{10} b_i \gamma_2^i z^{-i}}, \quad (4.34)$$

olarak ifade edilir. $\gamma_1 = 0.9$ ve $\gamma_2 = 0.6$ olarak alınır. Şekil 4.17'de LD-CELP (G728) için kodlayıcı ve kod çözücü yapısı verilmiştir.



Şekil 4.17. LD-CELP (G728) kodlayıcı (a) ve kod çözücü (b) yapısı

Şekil 4.18’da geleneksel CELP kodlayıcının aksine geçerli alt çerçevenin kodlanması süresince geçmiş değerlerden tahmin ile elde edilen uyartım kazancı bilinmektedir. Bu yüzden bazı hesaplama kazançları elde edilmiştir.



Şekil 4.18. LD-CELP kodlama döngüsü

LD-CELP için uyartım kod kitabı, şekil ve kazanç kod kitaplarından oluşmaktadır. Şekil kod kitabı için 7 bit ve kazanç kod kitabı için 3 bit olmak üzere toplam 10 bit kullanılmaktadır. Oluşturulan bu yapı sayesinde hesaplama karmaşıklığı azaltılmıştır. Kazanç kod kitabı toplam sekiz farklı değer alır. Bu 8 değer için 3 bitin atanması gerekir. Bu 3 bitin ilk 2 biti farklı genlikleri, son bit ise kazanç değerinin işaretini belirler. Bu nedenle, sekiz kazanç değerinin dört farklı büyüklüğü vardır ve bunların yarısı pozitif ve diğer yarısı negatiftir. Kazanç değerlerini $g^{(m)}$ ile gösterirsek, $m=0, \dots, 7$ için aşağıdaki gibi oluşturulur;

$$g^{(0)} = 33/64 \quad (4.35)$$

$$g^{(m)} = g^{(m-1)}7/4, \quad m = 1, 2, 3 \quad (4.36)$$

$$g^{(m)} = -g^{(m-4)}, \quad m = 4, 5, 6, 7. \quad (4.37)$$

Uyartım kod kitabı, şekil kod kitabı ve kazanç kod kitabı ile oluşturulabilecek 1024 farklı diziden oluşur. Böylece, şekil kod kitabı $\mathbf{v}^{(l)}$ ($0 < l < 127$) ile gösterilirse, her bir uyartım dizisi $g^{(m)}\mathbf{v}^{(l)}$ ile gösterilir. Bu diziler, formant sentez filtresi ve algısal ağırlıklandırma filtresinin seri bağlanması ile oluşturulmuş filtreden geçirilerek sıfır durum cevabı $\mathbf{y}_1^{(l,m)}$ elde edilir:

$$\mathbf{y}_1^{(l,m)} = g^{(m)}\mathbf{H}\mathbf{v}^{(l)}. \quad (4.38)$$

Burada \mathbf{H} , kaskad bağlı filtrenin darbe cevabı matrisidir. l ve m 'yi bulmak için aşağıdaki hata değerini minimum yapan l ve m araştırılır.

$$\varepsilon^{(l,m)} = \|\mathbf{t} - \mathbf{y}_1^{(l,m)}\|^2, \quad (4.39)$$

burada \mathbf{t} hedef vektördür ve aşağıdaki denklemde eşdeğeri bulunur:

$$\mathbf{t} = \frac{1}{G}(\mathbf{u} - \mathbf{y}_2), \quad (4.40)$$

burada \mathbf{u} , algısal ağırlıklı giriş ses vektörü, G kazanç değeri ve \mathbf{y}_2 , seri filtrelerin ($H(z)$ ve $W(z)$) sıfır giriş cevabı vektörüdür. Denklem 4.39'un daha genel ifadesi aşağıdaki gibi yazılabilir;

$$\varepsilon^{(l,m)} = \mathbf{t}^T\mathbf{t} - 2\mathbf{t}^T\mathbf{y}_1^{(l,m)} + \left(\mathbf{y}_1^{(l,m)}\right)^T\mathbf{y}_1^{(l,m)}, \quad (4.41)$$

burada $\mathbf{t}^T\mathbf{t}$ ifadesi, l ve m 'ye göre sabittir. Bu yüzden ifade aşağıdaki gibi sadeleştirilebilir,

$$\varepsilon^{(l,m)} = -2\mathbf{t}^T\mathbf{y}_1^{(l,m)} + \left(\mathbf{y}_1^{(l,m)}\right)^T\mathbf{y}_1^{(l,m)} = -2g^{(m)}\mathbf{t}^T\mathbf{H}\mathbf{v}^{(l)} + \left(g^{(m)}\right)^2\|\mathbf{H}\mathbf{v}^{(l)}\|^2. \quad (4.42)$$

Bu son denklemde, hatayı en küçük (minimize) edecek l ve m değerleri bulunur.

4.2.9 Uyarlamalı Çoklu Oran Kodlayıcı (AMR)

AMR, GSM için geliştirilmiş yeni bir konuşma kodlayıcı bileşenidir (Holma ve Toskala, 2011). AMR'nin iki farklı sürümü vardır. Bunlar AMR-WB ve AMR-NB olarak da bilinen temel sürümlerdir. Geniş bantlı AMR, 50 - 7000 Hz'lik bir konuşma bant genişliği kullanırken, dar bantlı AMR'nin bant genişliği 300 - 3400 Hz'dir. Bu geniş bantlı AMR'ye daha doğal bir konuşma kalitesi verir. Dar bant AMR ses kodlayıcısı, 12.2, 10.2, 7.95, 7.4, 6.7, 5.9, 5.15 ve 4.75 kbps bit hızlarına sahip sekiz ses kodlayıcı bileşiminin bir kombinasyonudur (Holma and Toskala, 2011). Dar bantlı AMR ses kodlayıcı bileşeni, ACELP algoritmasını kullanmaktadır. En düşük altı bit oranı, yarı oranlı kanal modunda kullanılabilir. Yarı oranlı kanal modu, iletişim kanalındaki trafiğin çok yoğun olduğu durumlarda kullanılmaktadır. Tam oranlı kanal modu ise iletişim trafiğinin az olduğu durumlarda kullanılır. Bu, ses kodlayıcı modlarında hata düzeltme ve kötü çerçeve algılamada gerçekleştirilmektedir. Bu, hata düzeltme için konvolüsyonlu kodlamanın ve kötü çerçeve tespiti için bir çevrimsel artıklık kontrolünün (CRC) kombinasyonu ile yapılır. AMR-geniş bant, 6.60 - 23.85 kbps arasında değişen dokuz farklı kodlayıcı oranı sunar (Holma and Toskala, 2011). 6.60 ile 19.85 kbps arasındaki kodlayıcı oranı GSM tarafından da desteklenebilir. AMR-Geniş Bant, AMR-Dar Bant ile aynı algoritmayı kullanır. AMR ses kodlayıcı, her telefon çağrısında doğru modu seçmek için tam oran veya yarı oran kanal modunu dinamik olarak seçer. Seçim, mobil istasyon ve baz istasyonu tarafından alınan kalitenin ve şebeke yükünün ölçülmesine dayanır. Yüksek trafik yüklerinde AMR, yarı oranlı kanal yoğun bir şekilde kullanır. Ağ daha az meşgul olduğunda, AMR en düşük iletişim koşullarını yaşayanlardan başlayarak mümkün olduğunca çok sayıda aramaya tam oran kanal kodlaması atar. Ağ aynı zamanda, en iyi çağrı kalitesini elde etmek için, AMR tam oran ve AMR yarı oranı içinde en iyi hata düzeltme seviyesini seçer. Kodlayıcı mod adaptasyonu olarak bilinen bu süreç, hücre boyunca gelişmiş ses kalitesiyle sonuçlanır. AMR ayrıca, ses etkinlik algılama ve süresiz iletimde içerir. Bu yapılar, sessizlik dönemlerinde kodlamayı ve iletimini durdurmak, böylece iletişim parazitini azaltmak ve pil ömrünü uzatmak için kullanılır.

4.2.10 Geliştirilmiş Ses Hizmetleri Kodlayıcı (The Enhanced Voice Services (EVS))

EVS, süper geniş bant ses kodlama standartıdır . 20 kHz'e kadar ses bant genişliği sunar. Geliştirilmiş, paket kaybı gizlemesine sahip olması nedeni ile yüksek güvenilirliğe sahiptir (Atti vd., 2015). Üçüncü Nesil Ortaklık Projesi (3GPP) tarafından geliştirilen EVS kodlayıcı bileşeni, gerçek zamanlı iletişim için gereken düşük gecikmeyi desteklemektedir. EVS'nin uygulama alanları, iyileştirilmiş telefon ve telekonferans, görsel-ışitsel konferans hizmetleri ve ses yayını içerir. EVS, geriye dönük uyumluluğun korunduğu AMR-WB gibi önekilere benzer kavramlar kullanmaktadır. EVS, ACELP ve Modifiye ayırık kosinüs dönüşümü (MDCT) kullanarak, içeriğe bağlı olarak ses sıkıştırma modları arasında geçiş yapar. EVS kodlayıcı bileşeni, HD ses için kullanılabilir.

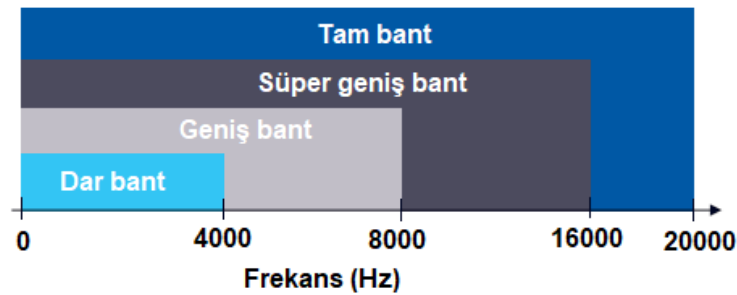
EVS'de aşağıdaki özellikler mevcuttur:

- Kaynak kontrollü değişken bit hızı (SC-VBR)
- Ses etkinliği detektörü (VAD)
- Uygun gürültü üretimi (CNG)
- Ağlarda paket kaybı için hata gizleme (EC)
- Çerçeve / paket hata tespitini artırmak için kanal modu
- Titreşim tampon (jitter buffer) yönetimi (JBM)

EVS için giriş örnekleme oranları 8, 16, 32 ve 48 kHz olabilir. Farklı bant genişlikleri için aşağıdaki bit hızlarını (kbps'de) destekler:

- Dar bant (NB): 5.9, 7.2, 8, 9.6, 13.2, 16.4, 24.4
- Geniş bant (WB): 5.9, 7.2, 8.96, 13.2, 13.2 kanal farkında, 16.4, 24.4, 32, 48, 64, 96, 128 (AMR-WB 10 için 6.6 ~ 23.85)
- Süper geniş bant (SWB): 9.6, 13.2, 13.2 kanal farkında, 16.4, 24.4, 32, 48, 64, 96, 128
- Tam Bant (FB): 16.4, 24.4, 32, 48, 64, 96, 128

Bit oranları her 20 ms'de değiştirilebilir. Şekil 4.19'da, iletişimde kullanılan bant frekans aralıkları verilmiştir.



Şekil 4.19. EVS için kullanılan çeşitli bant frekans aralıkları

EVS için yapılan öznel dinleme testleri, EVS'nin tüm çalışma noktalarında, AMR ve AMR-WB üzerinde önemli ölçüde geliştirilmiş ses kalitesi sunduğunu göstermektedir (Bruhn, 2015). EVS, müzik ve ses ile müzik karışımı için eşit derecede yüksek kalite sağlayan uygun bir kodlayıcıdır. Çizelge 4.5'te, 3GPP mobil kodlayıcıların karşılaştırması verilmiştir.

Çizelge 4.5. 3GPP mobil kodlayıcılar

	GSM EFR	AMR-NB	AMR-WB	AMR-WB+	EVS
Yıl	1996	1999	2001	2004	2014
Bant genişliği	Dar bant	Dar bant	Geniş bant	Tam bant	Dar bant Tam bant Geniş bant Süper geniş bant
Bit oranı	12.2kbps	4.75-12.2kbps	4.75-23.85kbps	6-48kbps	5.9-128 kbps

2G GSM sisteminde, en yüksek kalite oranını destekleyen Gelişmiş Tam Oran (EFR) kodlayıcı kullanılmıştır. 3G ve sonrası tüm 3GPP ses hizmetleri için varsayılan kod çözücü olan AMR dar bant (AMR-NB) kodlayıcı kullanılmıştır. AMR Geniş Bant (AMR-WB) kodlayıcı, yüksek çözünürlüklü ses için oluşturan kodlayıcıdır. Genişletilmiş AMR-WB (AMR-WB +) kodlayıcısı, çift kanal (stereo) sinyalleri kodlama imkanı sağlar ve müzik akışı gibi konuşma dışı uygulamalar içinde kullanılabilir. EVS kodlayıcı ise 3GPP kodlayıcı yapısındaki ses kalitesi açısından en iyi kodlayıcıdır.

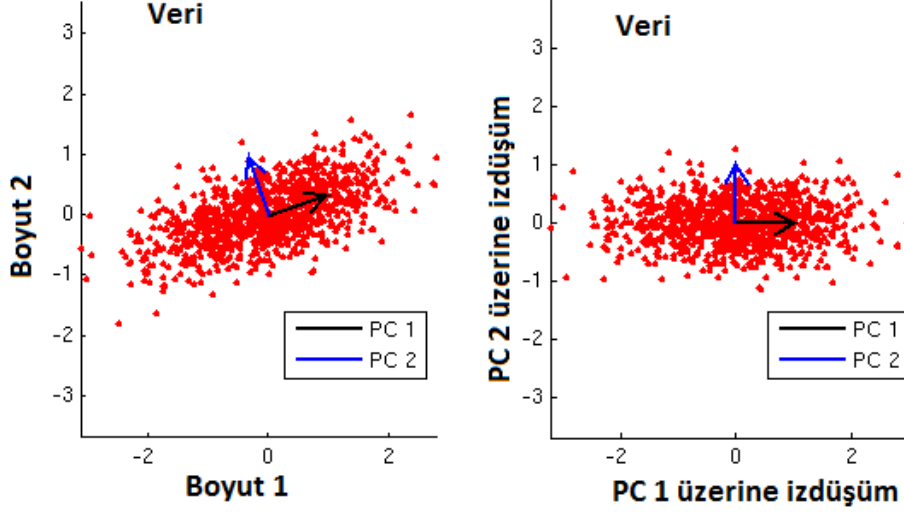
5. YÖNTEM

Büyük boyutlu sinyalleri daha küçük boyutlara bir dönüşüm ile haritalayan, yani original sinyali daha küçük boyutlu uzayda tanımlayan alt uzay yöntemlerine Ayrık Kosinüs Dönüşümü (DCT), Karhunen Loeve dönüşümü (KLT), bağımsız bileşen analizi (ICA), dalgacık dönüşümü (WT) örnek verilebilir. İçlerindeki en iyi dönüşüm ise KLT'dir. KLT'de, sinyal yapısı değiştikçe kullanılması gereken taban vektörleri de değişir. Bu yüzden yüksek bir işlem gücü ile taban vektörlerinin hesaplanması gerekir. Dolayısıyla, görüntü ve ses sinyalleri için kodlama işlemlerinde genellikle kullanılmaz. Buna karşın KLT, uygun taban vektörlerinden oluşturulan kod kitapları ve hesaplama gecikmesini kısaltacak nicemleme yöntemleri ile kodlamada kullanılabilir. Ayrıca bağımsız bileşen analizi ile de kodlama için benzer çalışmalar yapılabilir. Aşağıdaki bölümlerde KLT ve ICA dönüşümleri detayları ile açıklanmıştır.

5.1 Karhunen-Loeve Dönüşümü (KLT)

Karhunen-Loeve Dönüşümü ilk olarak Karhunen ve Loeve tarafından sürekli rasgele süreçler için seri genişletme olarak tanıtıldı. KLT, bir rasgele dizinin kovaryans veya korelasyon matrisini köşegenleştiren bir dönüşümdür (Goyal, 2001). Bu sayede veriye uygulanan dönüşüm sonucu korelasyonsuz dönüşüm katsayıları elde edilir. KLT ile dönüşüm sonucu elde edilen katsayılar için en iyi enerji yoğunluğu elde edilir. Bu katsayıların küçük değerli olanları atılarak ve kalan katsayı değerlerine ters KLT uygulanarak, veri sinyalleri yaklaşık olarak elde edilebilir. Bu özelliğinden dolayı KLT sinyal kodlama, sıkıştırma, imge tanıma gibi alanlarda kullanılabilir. Ancak KLT'nin bir dezavantajı sinyal bağımlı olmasıdır, yani dönüşüm uygulanan sinyal değiştikçe KLT'nin birbirine dik olan taban vektörleri de değişir. Sinyal bağımlılığı sebebi ile değişen bu taban vektörlerinin her defasında hesaplanması gerekeceğinden hızlı bir algoritmaya sahip değildir. Şekil 5.1'de, 2 boyutlu bir veri matrisinin dağılımı ve bu veri matrisine KLT uygulanması sonucu oluşan yeni dağılım gösterilmiştir. Şekilden de görülebileceği gibi, KLT ile veri dağılımının 2 eksene göre en yüksek varyanslarının olduğu yönlerde taban vektörleri elde edilir. Veriye bu taban vektörleri kullanılarak dönüşüm uygulandığında

Şekil 5.1'deki gibi yeni veri dağılımını istatistiksel olarak tamamen ilişkisiz (korelasyonsuz) olur. Burada PC, temel bileşeni (Principal component) gösterir.



Şekil 5.1. KLT için 2 boyutta veri dağılımı

Genel olarak, veri içerisindeki değerlerin ilişkisi ne kadar yüksek ise, KLT o kadar yüksek enerji birikimi sağlar. Bu ilişkiler korelasyon yada kovaryans matrisleri ile belirlenir. KL dönüşümünü anlamak için özdeğer ve özvektör kavramlarına değinmek gereklidir. Eğer C , $n \times n$ boyutlu bir matris, λ bir skaler değer olmak üzere bu matrisin bir özdeğeri ve \mathbf{u} vektörü, C 'nin λ özdeğerine karşılık gelen özvektör ise aşağıdaki eşitlik yazılabilir;

$$C\mathbf{u} = \lambda\mathbf{u} . \quad (5.1)$$

Eğer \mathbf{x} rastgele sütun vektörü olarak aşağıdaki gibi gösterilir ise;

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad (5.2)$$

ortalama vektör;

$$\mathbf{m}_x = \frac{1}{M} \sum_{k=1}^M \mathbf{x}_k, \quad (5.3)$$

kovaryans matrisi ise;

$$\mathbf{C}_x = E\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^T\}, \quad (5.4)$$

olarak tanımlanır. Operatör E ile \mathbf{C}_x , x_i ve x_j elemanları arasındaki ortak olasılık yoğunluk fonksiyonlarını kullanarak teorik olarak hesaplanmaktadır. Burada \mathbf{x} $n \times 1$ boyutlu bir vektör ve \mathbf{C}_x , $n \times n$ boyutlu bir matristir. \mathbf{C}_x 'nin c_{ii} elemanı, x_i 'nin varyansı ve \mathbf{C}_x 'nin c_{ij} elemanları x_i ile x_j arasındaki kovaryansı oluşturur. Eğer x_i ile x_j ilişkisiz ise kovaryans sıfıra eşittir ($c_{ij} = c_{ji} = 0$). Kovaryans matrisi aşağıdaki gibi yazılır.

$$\mathbf{C}_x = E\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^T\} = E\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x}^T - \mathbf{m}_x^T)\} = E\{\mathbf{xx}^T - \mathbf{x}\mathbf{m}_x^T - \mathbf{m}_x\mathbf{x}^T + \mathbf{m}_x\mathbf{m}_x^T\} \quad (5.5)$$

Buradan kolayca görülebileceği gibi;

$$\mathbf{x}\mathbf{m}_x^T = \mathbf{m}_x\mathbf{x}^T, \quad (5.6)$$

olur. Böylece

$$\begin{aligned} E\{\mathbf{xx}^T - \mathbf{x}\mathbf{m}_x^T - \mathbf{m}_x\mathbf{x}^T + \mathbf{m}_x\mathbf{m}_x^T\} &= E\{\mathbf{xx}^T - 2\mathbf{m}_x\mathbf{x}^T + \mathbf{m}_x\mathbf{m}_x^T\} \\ &= E\{\mathbf{xx}^T\} - E\{2\mathbf{m}_x\mathbf{x}^T\} + E\{\mathbf{m}_x\mathbf{m}_x^T\} \end{aligned}, \quad (5.7)$$

olur. Ortalama \mathbf{m}_x vektörü ve $\mathbf{m}_x\mathbf{m}_x^T$ matrisi sabit değerler içerdiği için aşağıdaki gibi yazabiliriz;

$$E\{\mathbf{xx}^T - 2\mathbf{m}_x\mathbf{x}^T + \mathbf{m}_x\mathbf{m}_x^T\} = E\{\mathbf{xx}^T\} - 2\mathbf{m}_xE\{\mathbf{x}^T\} + \mathbf{m}_x\mathbf{m}_x^T. \quad (5.8)$$

Ayrıca $E\{\mathbf{x}^T\} = \mathbf{m}_x^T$ olduğu için;

$$\mathbf{C}_x = E\{\mathbf{xx}^T\} - 2\mathbf{m}_x E\{\mathbf{x}^T\} + \mathbf{m}_x \mathbf{m}_x^T = E\{\mathbf{xx}^T\} - 2\mathbf{m}_x \mathbf{m}_x^T + \mathbf{m}_x \mathbf{m}_x^T. \quad (5.9)$$

$$\mathbf{C}_x = E\{\mathbf{xx}^T\} - \mathbf{m}_x \mathbf{m}_x^T, \quad (5.10)$$

bulunur. Burada kovaryans matrisi \mathbf{C}_x , aşağıda belirtilen şekilde de ifade edilebilir;

$$\mathbf{C}_x = \frac{1}{M} \sum_{k=1}^M \mathbf{x}_k \mathbf{x}_k^T - \mathbf{m}_x \mathbf{m}_x^T. \quad (5.11)$$

Burada \mathbf{C}_x gerçel ve simetriktir. \mathbf{C}_x 'in özdeğerleri λ_i ($i=1,2,\dots,n$)'lerin azalan sırada ($\lambda_i \geq \lambda_{i+1}$; $i=1,2,\dots,n-1$) sıralandıklarını ve herbirine karşılık gelen özvektörlerin sütun şeklinde \mathbf{u}_i ile gösterildiğini varsayalım. \mathbf{C}_x 'in özvektörlerinin satır halinde yazılması ile oluşturulan \mathbf{A} özvektör matrisi;

$$\mathbf{A} = \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix}, \quad (5.12)$$

ve $\mathbf{A}^T = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n]$ ile gösterilir. Burada \mathbf{A} 'nın ilk satırı, en büyük özdeğere karşılık gelen özvektörü temsil eder. Burada, \mathbf{A} matrisi dönüşüm matrisi olarak bilinir ve \mathbf{x} vektörünü \mathbf{y} vektörüne haritalar. Dönüşüm aşağıdaki gibi yazılır;

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x). \quad (5.13)$$

Yukarıdaki denklemde verilen dönüşüm Karhunen-Loeve veya Hotelling dönüşüm olarak adlandırılır. Denklem 5.13'teki \mathbf{y} vektörlerinin ortalaması aşağıdaki gibi sıfır bulunur;

$$E\{\mathbf{y}\} = E\{\mathbf{A}(\mathbf{x} - \mathbf{m}_x)\} = \mathbf{A}E\{\mathbf{x} - \mathbf{m}_x\} = \mathbf{A}(E\{\mathbf{x}\} - \mathbf{m}_x) = \mathbf{A}(\mathbf{m}_x - \mathbf{m}_x) = \mathbf{0}. \quad (5.14)$$

$\mathbf{m}_y = \mathbf{0}$ kabul edilirse, \mathbf{y} vektörlerinin kovaryans matrisleri;

$$\mathbf{C}_y = E\{(\mathbf{y} - \mathbf{m}_y)(\mathbf{y} - \mathbf{m}_y)^T\} = E\{\mathbf{y}\mathbf{y}^T\}, \quad (5.15)$$

olarak gösterilir ve;

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m}_x), \quad (5.16)$$

$$\mathbf{y}^T = [\mathbf{A}(\mathbf{x} - \mathbf{m}_x)]^T = (\mathbf{x} - \mathbf{m}_x)^T \mathbf{A}^T, \quad (5.17)$$

olur. Buradan kovaryans matrisi \mathbf{C}_y , aşağıdaki adımlardan sonra bulunur.

$$\mathbf{y}\mathbf{y}^T = \mathbf{A}(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^T \mathbf{A}^T, \quad (5.18)$$

$$E\{\mathbf{y}\mathbf{y}^T\} = E\{\mathbf{A}(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^T \mathbf{A}^T\} = \mathbf{A}E\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^T\} \mathbf{A}^T, \quad (5.19)$$

$$\mathbf{C}_y = \mathbf{A}\mathbf{C}_x\mathbf{A}^T, \quad (5.20)$$

$$\mathbf{C}_x\mathbf{A}^T = \mathbf{C}_x[\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_n] = [\lambda_1\mathbf{u}_1 \quad \lambda_2\mathbf{u}_2 \quad \dots \quad \lambda_n\mathbf{u}_n], \quad (5.21)$$

$$\mathbf{C}_y = \mathbf{A}\mathbf{C}_x\mathbf{A}^T = \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_n^T \end{bmatrix} [\lambda_1\mathbf{u}_1 \quad \lambda_2\mathbf{u}_2 \quad \dots \quad \lambda_n\mathbf{u}_n]. \quad (5.22)$$

Burada \mathbf{u}_i dik bir özvektördür ve aşağıdaki özelliklere sahiptir;

$$\mathbf{u}_i^T \mathbf{u}_i = 1, \quad i = 1, \dots, n \quad (5.23)$$

$$\mathbf{u}_i^T \mathbf{u}_j = 0, \quad i, j = 1, \dots, n, \quad (5.24)$$

ve bu yüzden \mathbf{C}_y kovaryans matrisinin köşegen değerleri \mathbf{C}_x 'in özdeğerlerinden oluşur.

$$\mathbf{C}_y = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}, \quad (5.25)$$

\mathbf{y} vektörlerinden oluşturulan kovaryans matrislerinin köşegen olmayan elemanları sıfıra eşittir. Bu yüzden \mathbf{y} vektörlerinin elemanları ilişkisiz olur. Orijinal \mathbf{x} vektörünü ona karşılık gelen \mathbf{y} vektöründen yeniden oluşturmak isteyelim. Öncelikle dik özvektörlere sahip bir \mathbf{A} matrisi için;

$$\mathbf{A}\mathbf{A}^T = \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \vdots \\ \mathbf{e}_n^T \end{bmatrix} [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \dots \quad \mathbf{e}_n] = \mathbf{I}, \quad (5.26)$$

olur. Burada \mathbf{I} birim matristir. Bundan dolayı, $\mathbf{A}^{-1} = \mathbf{A}^T$ olur, ve her hangi bir \mathbf{x} vektörü ona karşılık gelen \mathbf{y} vektörü ile aşağıdaki gibi bulunur;

$$\mathbf{x} = \mathbf{A}^T \mathbf{y} + \mathbf{m}_x. \quad (5.27)$$

\mathbf{C}_x 'in tüm özvektörleri yerine, sadece en büyük K adet özdeğere karşılık gelen özvektörleri

kullandığımızı varsayarsak kullanılan matris $\mathbf{A}_K = \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_K^T \end{bmatrix}$ ile gösterilebilir. Burada \mathbf{A}_K 'nın

boyutu $K \times n$ 'dir. Bu durumda \mathbf{y} vektörlerinin boyutu K olacaktır ve yaklaşık olarak elde edilecek sinyal aşağıdaki gibi bulunur;

$$\hat{\mathbf{x}} = \mathbf{A}_K^T \mathbf{y} + \mathbf{m}_x. \quad (5.28)$$

Orijinal \mathbf{x} sinyali ile yaklaşık elde edilen sinyal $\hat{\mathbf{x}}$ arasındaki ortalama kare hata aşağıdaki denklemde verilmiştir.

$$\mathbf{e}_{ms} = \sum_{j=1}^n \lambda_j - \sum_{j=1}^K \lambda_j = \sum_{j=K+1}^n \lambda_j . \quad (5.29)$$

5.2 Bağımsız Bileşen Analizi (ICA)

Bağımsız Bileşen Analizi (ICA), çok değişkenli verilerden istatistiksel olarak dikleştirilmiş ve aynı zamanda bağımsız olan bileşenler üretmeyi amaçlar (Comon,1994). ICA modelini kestirmedeki anahtar kavram, bağımsız bileşenlerin normalden farklı bir dağılım sergilemesidir. Bu kavram, genellikle normal-dışılık (non-gaussianity) olarak bilinir. Normal-dışılığın ikinci önemli ölçütü negentropidir. Terim olarak negentropi, entropinin ters işaretlisi yani negatif entropidir. Entropi, bir sistemin düzensizliğini ölçerken negentropi düzenliliğini ölçer. Bağımsız bileşenler analizi ilk olarak, kas büzülmesindeki hareketin basitleştirilmiş bir modelinin geliştirilmesi amacıyla yapılan bir çalışmada ortaya atılmıştır (Herault,1991). Bağımsız bileşenler analizi günümüzde görüntü işleme, beyin tomografisi, iletişim, finans, sismoloji gibi değişik disiplinlerde geniş bir uygulama alanına sahiptir (Switzer ve Green,1984). Bağımsız bileşenler analizinde çok değişkenli verilerin bir dizi bağımsız bileşenin (faktörün) doğrusal birleşiminden ibaret olduğu varsayılır. Rassal vektör $\mathbf{x} = (x_1, x_2, \dots, x_m)^T$ 'in i 'inci bileşeni, x_i bağımsız bileşenlerin (s_k) toplamı olarak aşağıdaki gibi gösterilebilir:

$$\mathbf{x}_i = \mathbf{a}_{i,1}\mathbf{s}_1, \dots, \mathbf{a}_{i,k}\mathbf{s}_k, \dots, \mathbf{a}_{i,n}\mathbf{s}_n , \quad k = 1, 2, \dots, n . \quad (5.30)$$

Burada $a_{i,k}$ karışım ağırlıklarıdır. \mathbf{x} vektörünün taban vektörleri (basis vectors) $\mathbf{a}_k = (a_{1,k}, \dots, a_{m,k})^T$ ile gösterilir ise taban vektörü \mathbf{a}_k , \mathbf{A} karışım matrisinin sütununu oluşturur. $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]$ matrisi ile birlikte \mathbf{x} vektörü aşağıdaki gibi ifade edilir.

$$\mathbf{x} = \mathbf{A}\mathbf{s} . \quad (5.31)$$

Burada $\mathbf{s} = (s_1, \dots, s_n)^T$ olur. Bağımsız bileşen analizindeki amaç, karışım matrisi \mathbf{A} ve bağımsız bileşenler \mathbf{s} 'i bulmaktır. Eğer taban vektörlerinin sayısı, \mathbf{x} 'sin boyutuna eşitse ($n=m$), \mathbf{A} matrisinin tersi $\mathbf{W} = \mathbf{A}^{-1}$ olmaktadır. Burada \mathbf{W} ayırıştırma (unmixing)

matrisidir ve $s=W \cdot x$ ile bağımsız bileşenler bulunmaktadır. Eğer $n>m$ ise, matris tersi (pseudo inverse) ile yine bağımsız bileşenler bulunabilir (Comon, 1994).

5.3. Tez Çalışmasında Gerçekleştirilen Kodlayıcılar

Bu çalışmada, ses işaretlerinin kodlanması için KLT ve ICA temelli iki yeni yöntem önerilmiştir. Çalışmada, KLT ve ICA tabanlı yöntemler için sırasıyla, kümelenecek az sayıdaki özvektörler ve bağımsız bileşenler kullanılmıştır. KLT ve ICA için oluşturulan kod kitapları kullanılarak ses sıkıştırması gerçekleştirilmiştir. KLT için daha fazla sıkıştırma işlemi, küçük özdeğerlere karşılık gelen oto-kovaryans özvektörlerini atarak ve kalan özvektörlere de vektör niceme uygulayarak elde edilmiştir. Buna ek olarak, daha iyi SNR için, kullanılabilir bant genişliğinin geri kalan kısmını kullanan tekrar eden bir hata düzeltme süreci önerilmektedir. Geliştirilen yaklaşımların tümü için etkili bir ses kodlama biçimi oluşturulmaktadır. ICA, alt uzay temelli ses kodlamada literatürde ilk kez kullanılmaktadır. Önerilen yöntemlerle, CBR ve VBR kullanan kodlama algoritmaları geliştirilmiştir. Deneysel çalışmalarda TIMIT ses veri tabanı kullanılmıştır. Ses sinyalleri KLT ve ICA için çeşitli çerçeve uzunluklarını kullanarak 2.4 kbps, 8 kbps, 12.2 kbps, 16 kbps, 16.4kbps ve 19.85 kbps oranlarında sentezlenmiştir. Sentezlenen ses sinyallerinin kaliteleri LPC (2.4 kbps), G728 (LD-CELP, 16 kbps), G729A (CS-CELP, 8 kbps), EVS (16.4 Kbps), AMR-NB (12,2 kbps) ve AMR-WB (19,85 kbps) gibi kodlayıcılarla karşılaştırılmıştır. Karşılaştırma için kişisel kalite ölçütü olan, Ortalama Görüş Puanı (MOS), Ses Kalitesinin Algısal Değerlendirmesi (PESQ) ve Birleşik Ölçüt olan, “Cov” kullanılmıştır.

Literatürdeki alt uzay tabanlı yöntemleri kullanan çoğu ses kodlama ile ilgili çalışma, dönüşüm düzlemindeki bir kaç katsayının kullanılması ile sinyal üretmeyi içermektedir. Geleneksel olarak uygulanan dönüşümler, Karhunen Loeve Dönüşümü (KLT) (Goyal, 2001), Bağımsız Bileşen Analizi (ICA) (Ferreira ve Figueiredo, 2003), Ayırık Kosinüs Dönüşümü (DCT) (Ahmed vd., 1974), Hızlı Fourier Dönüşümü (FFT) (Kumar ve Kumar, 2012) ve Dalgacık Dönüşümü (WT) (Skodras vd., 2001) olarak bilinmektedir. DCT ve WT, görüntü kodlama için popüler olmasına rağmen, KLT ve ICA gibi sinyalin içeriğine bağımlı olan yöntemler için, pratik ses kodlama uygulamalarında daha fazla araştırma yapmaya gerek vardır.

KLT, enerji sıkıştırma ve ilişkisizlik açısından istatistiksel olarak en etkili dönüşümdür. Bir sinyal belirli bir zamansal ilişki ile (yani, $R(\tau) \neq \delta$) Gauss dağılımına sahipse, sinyalin KL dönüşümünü kullanmak, orijinal sinyali kullanmaktan daha etkilidir (Kim ve Kleijn, 2004; Lee ve Kim, 2010; Ozerov ve Kleijn, 2011). Buna ek olarak, bir Gauss sinyali, Gauss olmayan dağılımlara sahip birçok bağımsız bileşenin ağırlıklı toplamı kullanılarak elde edilebilir. Bu nedenle, bir sinyali yeniden oluşturmak için KLT ve ICA tabanlı yöntemler tercih edilebilir.

Önerilen KLT ve ICA tabanlı yöntemlerin performansları, farklı bit oranları için değerlendirilmiştir. Benzer alt uzay tabanlı çalışmalar, standart kodlayıcıların (örneğin CELP, LPC, vb.) temel yapılarını göz önüne alarak ses kodlamayı gerçekleştirirken (Kim ve Kleijn, 2004; Lee ve Kim, 2010; Ozerov ve Kleijn, 2011; Ju vd., 2014; Oger vd., 2006), bu tez çalışmasında önerilen ses kodlayıcı biçimi, standart kodlayıcıların yapılarını kullanmaz. Standart kodlayıcıların yapılarına dayanan çalışmaların aksine, bu tez çalışması; KLT'yi kullanarak elde edilen özvektör matrislerinin ana bileşen vektörlerini kullanarak vektör nicemlemesini ve tekrarlanan bir hata azaltma sürecini içermektedir. Hata azaltma süreci, istenilen bit oranı elde edilinceye kadar SNR'i sürekli iyileştirmek için, hata sinyalini girişe geri besleyerek oluşturulur.

Önerilen KLT tabanlı kodlayıcı, bir eğitim ses veri kümesinden elde edilen otokorelasyon matrislerinin özdeğerlerini ve özvektörlerini kullanır. Özvektör ve dönüşüm katsayılarına ait kod kitapları oluşturulduktan sonra kodlayıcı oluşturulur ve performanslar test kümesindeki ses sinyalleri üzerinde ölçülür. Burada ses sinyalleri, kod kitapları kullanılarak elde edilen en yakın komşu değerlerine göre sentezlenir. Eğitim ve test aşamalarında çeşitli parametrelerin test edilmesi gerekir. Sonuçlar geleneksel ses kodlama yöntemlerinin performanslarıyla karşılaştırılmıştır. Özellikle, kod kitabı ve kod vektör boyutları, vektör niceme (VQ), çerçeve boyutu ve son olarak veri oranının etkileri derinlemesine incelenmiştir. Ek olarak, eğitim kümesindeki ses verileri kullanılarak ICA tabanlı kodlayıcılar için bağımsız bileşen ve karışım matrisine ait kod kitapları oluşturulmuştur. Yine, test kümesindeki ses sinyalleri, önceden eğitilmiş kod kitapları kullanılarak sentezlenir. Önerilen yöntemlerle, Sabit Bit Oranlı (CBR) ve Değişken Bit Oranlı (VBR) kodlama yaklaşımları benimsenmiştir. Önerilen algoritma, küçük kovaryans

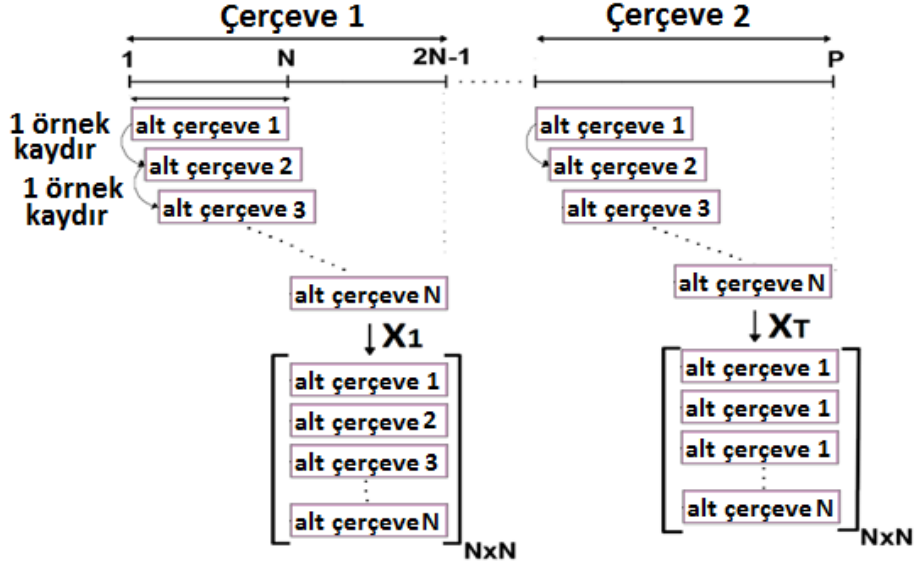
özdeğerlerinin ortadan kaldırılması için klasik bir alt uzay tabanlı bir yöntem geliştirmek için iki ilave yaklaşımdan oluşur:

- eğitim kümesinden bulunmuş özvektör matrislerinin nicemlenmesi ve
- hedef bit oranında SNR'yi iyileştirmek için hata sinyalinin geri beslenmesi.

5.3.1. KLT temelli kodlayıcı

İlk alt uzay stratejisi, sinyal örneklerini ilişkisizleştirmeyi sağlayan KLT'ye bağlıdır. KLT'nin taban fonksiyonları, giriş sinyalinin otokovaryans matrisinin özvektörleri olup, özvektörler, azalan bir sıraya göre dizilen özdeğerlere karşılık gelmektedir. Dönüşüm katsayıları, özvektör ile ilişkili olarak giriş sinyalinin o özvektör üzerine izdüşümü olarak tanımlanır. Eğer kod çözücü kısım, özvektörler ve dönüşüm katsayıları hakkında bilgiye sahip ise, alıcı kısımda sinyal, özvektörlerin doğrusal bir birleşimi ile yaklaşık olarak elde edilebilir. Kod kitapları, TIMIT veri tabanının eğitim kümesindeki ses sinyali çerçevelerini kullanarak oluşturulmuştur (Zue ve ark., 1990). Test kümesindeki ses sinyalleri bu kod kitapları kullanılarak sentezlenmiştir. Deneysel çalışmalarda, kod kitabı büyüklükleri, 2^{10} ile 2^{16} arasındaki değerleri almıştır.

Önerilen yöntem, bir ses biriminin çerçevelerinden bir kovaryans matrisi elde ederek başlar. N uzunluklu alt çerçeveler, M ($M = 2N-1$) uzunluğundaki her çerçeve üzerinde 1 örnek kaydırılarak $N \times N$ boyutlu bir veri matrisi oluşturulur. Kaydırılan her alt çerçeve, veri matrisindeki bir satırı oluşturur. Şekil 5.2'de bir ses birim için veri matrislerinin nasıl oluşturulduğu gösterilmektedir.



Şekil 5.2. Bir ses birim sinyalinin $N \times N$ lik veri matrislerinin oluşturulması

Eğitim veri kümesindeki p 'inci veri matrisinin (\mathbf{X}_p) kovaryans matrisi aşağıdaki gibi elde edilir:

$$\mathbf{C}_p = \frac{1}{N-1} \sum_{i=1}^N [(\mathbf{x}_i - \mathbf{m}_x)(\mathbf{x}_i - \mathbf{m}_x)^T], \quad p=1,2,\dots,r. \quad (5.32)$$

Burada \mathbf{x}_i ($\mathbf{x}_i \in \mathbf{R}^N$), p 'inci veri matrisindeki i 'inci satır vektörüne karşılık gelen i 'inci sütun vektörüdür. r ise, deneylerde 1024, 2048, 4096, 16384, 32768 ve 65536 gibi değerler alan kovaryans matrislerinin sayısıdır. Denklem 5.32'deki N , bir veri matrisindeki satır ve sütunların sayısıdır. \mathbf{m}_x ise sütun vektörlerinin ortalamasıdır. Burada özvektör ve katsayı kod kitapları kovaryans matrisinin özvektörlerini kullanarak oluşturulur. Kovaryans matrisinin özdeğerleri azalan sırada dizildiğinde $\{\lambda_1 > \lambda_2 > \dots > \lambda_N\}$, özvektör matrisi Φ_p , denklem 5.33'de gösterildiği gibi, \mathbf{C}_p 'nin en büyük K özdeğerlerine karşılık gelen özvektörleri kümeleyerek oluşturulur.

$$\Phi_p = \{\phi_1, \phi_2, \dots, \phi_K\}. \quad (5.33)$$

Burada ϕ 'ler, N -boyutlu özvektörlerdir ($\Phi_p \in \mathbf{R}^{N \times K}$). Özvektör kod kitabı Φ_p matrislerinin kümeler halinde yazılması ile aşağıdaki gibi gösterilir:

$$\Phi^{cb} = \{\Phi_1, \Phi_2, \dots, \Phi_r\}. \quad (5.34)$$

p 'inci katsayı matrisi \mathbf{Y}_p aşağıdaki şekilde yazılır:

$$\mathbf{Y}_p = \Phi_p^T \mathbf{X}_p. \quad (5.35)$$

Burada \mathbf{X}_p , boyutu $N \times N$ olan p 'inci veri matrisidir ve $\mathbf{Y}_p \in \mathbf{R}^{K \times N}$. $K \times M$ ($M=N.r$) boyutuna sahip bir katsayı kod kitabıdır (\mathbf{Y}^{cb}). Katsayı kod kitabı, katsayı matrislerinin tüm sütunlarının birleştirilmesi ile oluşturulur. Katsayı vektörlerinin sayısı (M) 1024, 2048, 4096, 16384, 32768 ve 65536'dan birine eşittir. Bu sayılar farklı çerçeve uzunluklarına ve farklı bit oranlarına göre seçilmiştir.

Test ses çerçeve sinyalleri ($\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^t$) üst üste bindirilmeden test aşamasında oluşturulur. q 'inci test çerçevesi (\mathbf{x}^q), özvektör matrisleri tarafından gerilen alt uzay üzerine izdüşümü alınarak sıkıştırılır. k 'inci iterasyon için, bir özvektör matrisi ve katsayı vektörü kullanılarak matematiksel olarak bulunan Öklid uzaklıkları, q 'inci test çerçevesi (\mathbf{x}_k^q) için aşağıdaki gibi ifade edilir:

$$f_p^q = \left\| \mathbf{x}_k^q - (\Phi_p \mathbf{y}_p^q) \right\|, \quad p=1,2,\dots,r, \quad (5.36)$$

Burada Φ_p , kod kitabındaki q 'inci özvektör matrisi ve $\mathbf{y}_p^q = (\Phi_p^T \mathbf{x}_k^q)$ 'dir. k 'inci iterasyon için en küçük Öklid uzaklık, a_k olarak gösterilir;

$$a_k = \operatorname{argmin} (f_p^q), \quad p=1,2,\dots,r. \quad (5.37)$$

Eğer en küçük Öklit uzaklığı veren özvektör matrisi Φ_{a_k} olarak belirtilirse, katsayı vektörü (\mathbf{y}_k^q), k 'inci iterasyon için aşağıdaki gibi ifade edilir:

$$\mathbf{y}_k^q = \Phi_{a_k}^T \mathbf{x}_k^q. \quad (5.38)$$

Öklid uzaklıkları (A_k^q), \mathbf{y}_k^q kullanarak ;

$$A_l^q = \|\mathbf{y}_k^q - \mathbf{Y}_l\|, \quad l=1,2,\dots,M, \quad (5.39)$$

ifade edilir. Burada \mathbf{Y}_l , kod kitabındaki l 'inci sütun vektörüdür. En küçük Öklit uzaklığı veren katsayı vektörünün indisi, k 'inci iterasyon için h_k ile aşağıdaki gibi belirtilir;

$$h_k = \text{argmin}(A_l^q), \quad l=1,2,\dots,M. \quad (5.40)$$

En küçük Öklit uzaklığı veren katsayı vektörü, kodlayıcı tarafında $\hat{\mathbf{y}}_k^q = \mathbf{Y}_{h_k}$ ile gösterilir. Özvektör matrisi (Φ_{a_k}) ve katsayı vektörü ($\hat{\mathbf{y}}_k^q$) kullanılarak, test çerçevesi ($\hat{\mathbf{x}}_k^q$) yaklaşık olarak aşağıdaki şekilde bulunur;

$$\hat{\mathbf{x}}_k^q = \Phi_{a_k} \hat{\mathbf{y}}_k^q. \quad (5.41)$$

Bir hata sinyali ise;

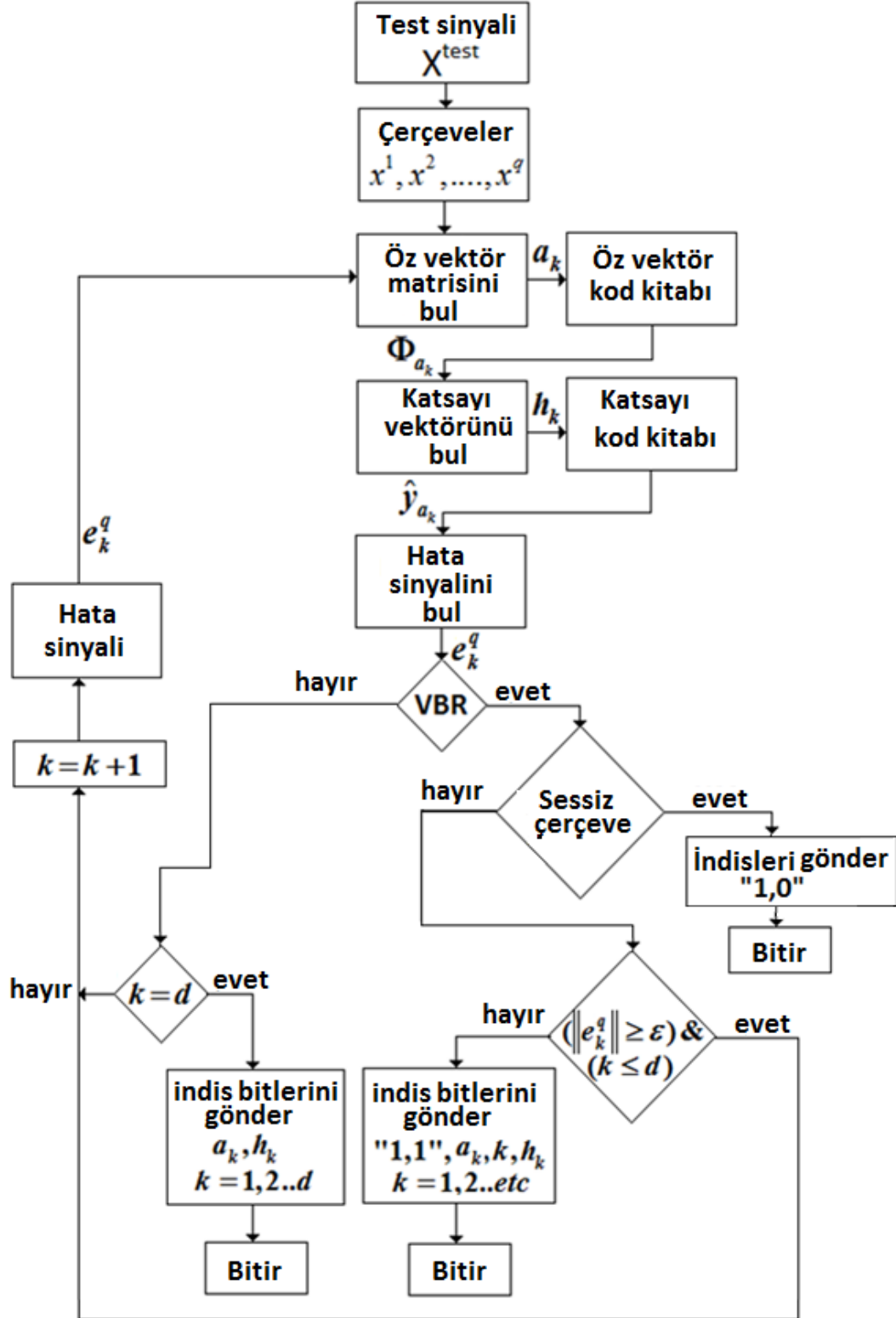
$$\mathbf{e}_k^q = \mathbf{x}_k^q - \hat{\mathbf{x}}_k^q. \quad (5.42)$$

olarak ifade edilir. Eğer istenilen bit oranı için hata azaltma işlemi kullanılıyorsa, denklem 5.36'daki \mathbf{x}_k^q yerine hata sinyali (\mathbf{e}_k^q) kullanılır. Başka bir deyişle, \mathbf{e}_k^q ikinci iterasyon için test çerçevesi olarak kullanılır ve $\hat{\mathbf{x}}_k^q$ 'i bulmak için kullanılan aynı işlemler izlenerek $\hat{\mathbf{e}}_k^q$ bulunur. $\hat{\mathbf{x}}_k^q$ ve $\hat{\mathbf{e}}_k^q$ 'i bulmak için kullanılan özvektör matrisine ve katsayı vektörüne ait olan indisler kod çözücüye gönderilir. Bu indisler kullanılarak, kod çözücünde sentezlenen sinyal aşağıdaki şekilde yazılır;

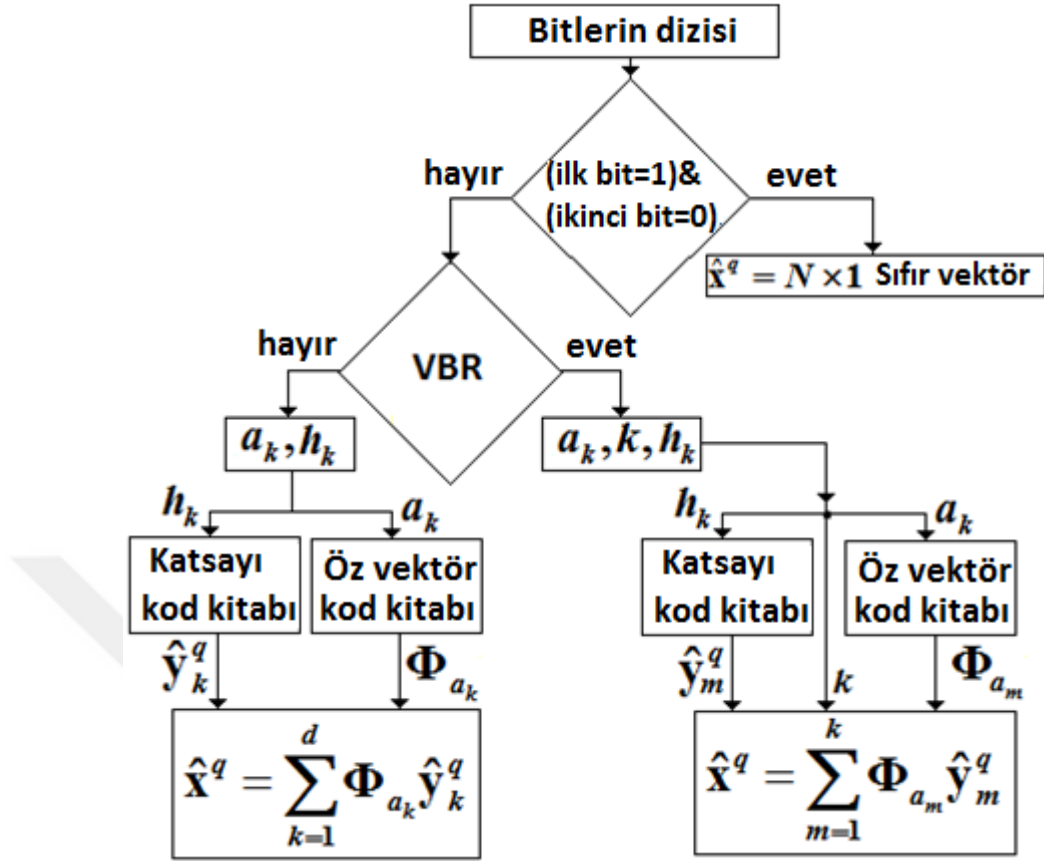
$$\hat{\mathbf{x}}^q = \hat{\mathbf{x}}_1^q + \sum_{k=2}^d \hat{\mathbf{e}}_k^q. \quad (5.43)$$

Hata sinyalinin iteratif olarak SNR düzeltme için kullanılması ile sentezlenmiş sinyal ($\hat{\mathbf{x}}^q$) ve \mathbf{x}^q arasındaki hata azalır. Şekil 5.3'de, önerilen KLT tabanlı alt uzay yönteminin kodlayıcı kısmı ve Şekil 5.4'te ise kod çözücü kısımları gösterilmiştir. Şekil 5.3'de k , başlangıçta 1 olarak ayarlanan geçerli iterasyon değeridir. Burada d , CBR ve

VBR kodlaması için kullanılan en yüksek iterasyon sayısıdır. Bu işlem, test sinyalinin tüm çerçeveleri için gerçekleştirilir.



Şekil 5.3. KLT temelli alt uzay yöntem için kodlayıcı blok şeması.



Şekil 5.4. KLT temelli yöntem için kod çözücü blok şeması.

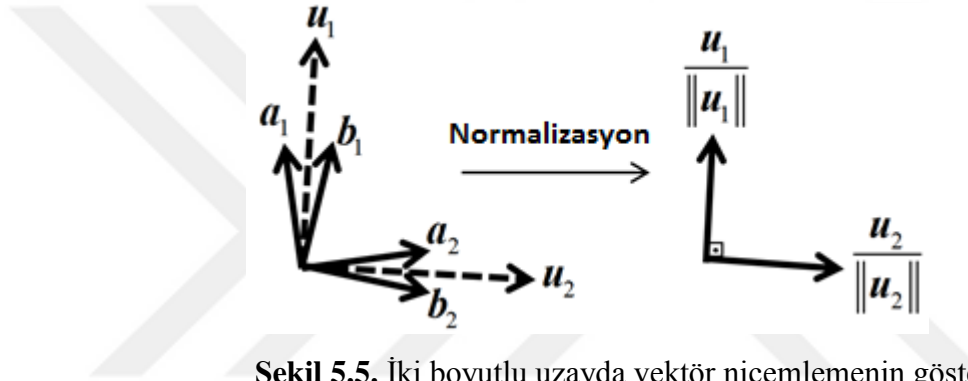
5.3.2. Özvektör nicemleme

Vektör nicemleme, hesaplama gecikmesini kısaltmak için özvektörler üzerinde gerçekleştirilir. Her bir özvektör matrisinin temel bileşen özvektörünü kullanan K-ortalama (K-means) yöntemi (Jain, 2010), özvektör nicemleme için kullanılmıştır. Böylece, benzer yöndeki temel özvektörleri içeren özvektör matrislerinden kümeler elde edilir. Yeni özvektör kod kitabı, nicemlemeden sonra $\theta^{cb}=\{\theta_1,\theta_2,\dots,\theta_m\}$ ile gösterilir, $m < r$. Yeni özvektör matrisi j 'nci küme için bulunur ve aşağıdaki şekilde tanımlanır;

$$\theta_j = \sum_{i=1}^p \Phi_i, \quad j=1,2,\dots,m, \quad (5.44)$$

burada p , j 'inci kümedeki özvektör matrislerinin sayısıdır. Özvektör matrislerinin sütunlarının her biri, sütunların normuna bölünerek normalize edilir. Normalize edilmiş özvektör matrislerinin boyutu $N \times K$ olur.

Örnek olarak, $\mathbf{A}=[\mathbf{a}_1 \ \mathbf{a}_2]$ ve $\mathbf{B}=[\mathbf{b}_1 \ \mathbf{b}_2]$ temel bileşen vektörleri \mathbf{a}_i ve \mathbf{b}_i olan iki matris ise, ve \mathbf{u}_1 ile \mathbf{u}_2 , \mathbf{A} ve \mathbf{B} matrislerinin ilk $(\mathbf{a}_1, \mathbf{b}_1)$ ve ikinci $(\mathbf{a}_2, \mathbf{b}_2)$ temel bileşenleri ise $\mathbf{u}_1 = (\mathbf{a}_1 + \mathbf{b}_1)$ ve $\mathbf{u}_2 = (\mathbf{a}_2 + \mathbf{b}_2)$ olur, böylece matris biçiminde $\mathbf{U}=[\mathbf{u}_1 \ \mathbf{u}_2]$ olur. Normalize edilmiş matris ise $\mathbf{U}_{norm} = \left[\frac{\mathbf{u}_1}{\|\mathbf{u}_1\|} \ \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|} \right]$ olur. İki boyutlu uzayda, aynı kümede bulunan \mathbf{A} ve \mathbf{B} matrislerinin normalizasyon işlemi Şekil 5.5'te gösterilmiştir.



Şekil 5.5. İki boyutlu uzayda vektör nicemlemenin gösterimi.

Özvektör kod kitabında optimal özvektör matrisinin araştırılması için 3 farklı yaklaşım kullanılmıştır.

i) Birinci yaklaşımda (**A1**), ses sinyalleri özvektör kod kitabındaki tüm özvektör matrislerinin teker teker kullanılması ile sentezlenir. (Hesaplama gecikmesi yüksektir).

ii) İkinci yaklaşımda (**A2**), sentezleme işlemi yalnızca M normalize edilmiş özvektör matrislerinin (\mathbf{U}_{norm}) kullanılması ile yapılır. (Hesaplama gecikmesi düşüktür). Bu yaklaşımda K- ortalama yöntemi kullanılmıştır.

iii) Üçüncü yaklaşımda (**A3**), hesaplama gecikmesini azaltmak ve bit atama verimliliğini arttırmak için, özvektör matrislerinin nicemleme bölgeleri içinde eşit sayıda özvektör matrisi içeren kümelere nicemlenmesini sağlayan yeni bir nicemleme tekniği geliştirilmiştir. Bu, 'dengeli K-ortalama (means)' olarak adlandırıldı. (Hesaplama gecikmesi düşüktür). Üçüncü yaklaşım için yeni kod kitabı yapısı, aşağıdaki algoritma ile

gerçekleştirilmiştir. Bu algoritmanın sonunda, M adet küme içeren ve her bir küme içinde L adet özvektör matrisi olan bir kod kitabı yapısı oluşturulmuştur.

Adım 1) $t = 1$ olarak ayarlanır.

Adım 2) Birinci özvektör matrisi ile kalan özvektör matrislerinin temel vektörleri arasındaki Öklid uzaklıkları bulunur;

$$\mathbf{D}_i = \|\Phi_1 - \Phi_i\| \quad i=2,3,\dots,N$$

Burada N , özvektör kod kitabındaki özvektör matrislerinin sayısıdır.

Adım 3) $N-1$ adet özvektör matrisi arasında, en küçük $L-1$ adet Öklid uzaklıklarını veren $L-1$ adet özvektör matrisi bulunur. Birinci özvektör matrisi ile $L-1$ adet özvektör matrisinin birleştirilmesi sonucu L adet özvektör matrislerinden oluşan t 'inci küme oluşturulur (ψ_t).

Adım 4) Şekil 5.5'te açıklandığı gibi t 'inci kümedeki özvektör matrislerini (ψ_t) kullanarak t 'inci normalize edilmiş özvektör matrisi (\mathbf{U}_{norm}^t) bulunur. t 'inci kümedeki özvektör matrisleri, özvektör kod kitabından çıkarılır.

Adım 5) t değeri 1 arttırılır. $t=M$ ise algoritma sonlandırılır. Aksi takdirde Adım 2'ye gidilir. Burada M , kümelerin sayısıdır.

Bir test çerçevesi için, en uygun küme, denklem 5.36'daki en küçük Öklid uzaklığı veren normalize edilmiş bir özvektör matrisi kullanılarak seçilir. Daha sonra, en küçük Öklid uzaklığına sahip en uygun özvektör matrisi, seçilen kümeye ait L adet özvektör matrisleri içerisinde bulunur. $(L + M) \ll N$ olduğundan, hesaplama gecikmesi önemli ölçüde azaltılır. Buna ek olarak, yeni kod kitabı için kullanılan bit sayısı $(=\log_2 M + \log_2 L)$ ile nicemlenmemiş olan kod kitabına atanan bit sayısı $(=\log_2 N)$ eşittir. Katsayı kod kitapları için de benzer algoritma uygulanmakta ve katsayı vektörlerinin hesaplama gecikmesi üçüncü yaklaşım için kısaltılmaktadır. İkinci yaklaşım için kümelerdeki katsayılar ve özvektörler üçüncü yaklaşımdaki gibi aranmaz. İkinci ve üçüncü yaklaşımlar için kod kitaplarına atanan bit sayıları eşittir. Ancak üçüncü yaklaşımın hesaplama gecikmesi ikinci yaklaşımdan daha kısadır.

Sayısal bir örnek verilecek olursa; diyelim ki 64 özvektör matrisimiz var ve 16 küme oluşturulmak isteniyor. Bunun sonucunda üçüncü yaklaşım ile 16 kümeli, her biri 4 özvektör matrisine sahip bir kod kitabı üretilir ve toplam bit sayısı ($\log_2 4 + \log_2 16 = 6$) olur. Bit sayısı doğal olarak $\log_2 64 = 6$ 'ya eşittir. Üçüncü yaklaşımın (A3) işlem stratejisine göre, her bir kümedeki bu 4 özvektör matrisi kullanılarak, Şekil 5.5'deki gibi bir adet normalize edilmiş özvektör matrisi oluşturulur. Böylece 16 adet normalize edilmiş özvektör matrisi üretilir. Test çerçevesi için, denklem 5.36'da en küçük Öklid uzaklığı veren, en uygun normalize edilmiş özvektör matrisi seçilir. Ardından seçilen özvektör matrisine karşılık gelen kümeye içerisindeki dört özvektör matrisi arasından, denklem 5.36'ya göre yine en uygun özvektör matrisi seçilir. Sonuç olarak toplam 20 (4 + 16) özvektör matrisi için arama yapılır. Eğer bu şekilde nicemleme yapılmıyaydı toplam 64 özvektör matrisi için arama yapılacaktı.

5.3.3. ICA temelli kodlama

ICA, özelliklerine göre KLT'ye benzer. Bununla birlikte, KLT'nin aksine, ICA'nın taban vektörleri birbirine dik değildir. Bu tez çalışmasında, bu iki alt uzay yönteminin sentezlenen ses kalitesi anlamında performansları incelenmiştir. Rassal değişkenlerin karşılıklı bilgileri, değişkenler arasındaki karşılıklı bağımlılığın bir ölçüsüdür. ICA ile, rastgele bir vektöre (ya da matrise) Bağımsız Bileşen Analizi uygulanması sonucunda bulunan bileşenler arasındaki istatistiksel bağımlılık en aza indirilmektedir (Comon, 1992). ICA uygulamaları; veri sıkıştırma, veri kaynaklarının tesbiti, kör tanıma ve dekonvolüsyonu içerir (Comon, 1992). Bu çalışmada ICA yöntemi, FastICA algoritması (Hyvärinen ve Oja, 2000) kullanılarak MATLAB programı ile gerçekleştirilmiştir. Kod kitaplarını oluşturmak için, başlangıçta eğitim aşamasında elde edilen ses sinyali çerçeveleri, çakışmayan alt çerçevelere bölünmüştür. FastICA algoritması, her çerçevenin alt çerçevelerinden elde edilen veri matrisine uygulanmıştır. Daha sonra her veri matrisi için karışım matrisi (**A**) ve bağımsız bileşen matrisi (**S**) bulunmuştur. Bulunan bu matrisler ile bağımsız bileşen ve karışım kod kitapları oluşturulmuştur. Test aşamasında, bu kod kitapları kullanılarak test sinyalleri sentezlenir.

p 'inci veri matrisi, $\mathbf{X}_p \in \mathbf{R}^{M \times N}$ ile gösterilirse bu matris aşağıdaki gibi ifade edilebilir:

$$\mathbf{X}_p = \mathbf{A}_p \mathbf{S}_p, \quad p=1,2,\dots,r, \quad (5.45)$$

Burada \mathbf{A}_p , $M \times M$ boyutlu bir karışım matrisi; \mathbf{S}_p , $M \times N$ boyutlu bir bağımsız bileşen matrisi ($M < N$) ve r , eğitim kümesinden elde edilen veri matrislerinin sayısıdır. Bağımsız bileşen kod kitabı, bağımsız bileşen matrislerinin kümesi olarak aşağıdaki gibi oluşturulur;

$$\mathbf{S}^{cb} = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_r\}. \quad (5.46)$$

Karışım kod kitabı (\mathbf{U}^{cb}) tüm karışım matrislerinin satırlarının birleştirilmesi ile oluşturulur. Burada $\mathbf{U}^{cb} \in \mathbf{R}^{Z \times M}$ ve Z değeri $M \times r$ büyüklüğündedir. Test aşamasında, test ses sinyalleri ($\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^t$) üst üste örtüşmeksizin test kümesinden elde edilir. k 'inci iterasyon için p 'inci sinyal aşağıdaki gibi hesaplanır;

$$\mathbf{x}_{p_k}^q = \left(\mathbf{x}_k^q \mathbf{S}_p^T (\mathbf{S}_p \mathbf{S}_p^T)^{-1} \right) \mathbf{S}_p, \quad p=1,2,\dots,r, \quad (5.47)$$

Burada \mathbf{x}_k^q , k 'inci iterasyon için q 'inci test çerçevesi ve \mathbf{S}_p , p 'inci bağımsız bileşen kod kitabı içindeki bağımsız bileşen matrisidir. \mathbf{x}_k^q ve $\mathbf{x}_{p_k}^q$ arasındaki minimum Öklit uzaklığını veren k 'inci iterasyon indisi (a_k) aşağıdaki gibi hesaplanır:

$$a_k = \operatorname{argmin} \left(\left\| \mathbf{x}_k^q - \mathbf{x}_{p_k}^q \right\| \right), \quad p=1,2,\dots,r. \quad (5.48)$$

Bağımsız bileşen kod kitabından \mathbf{S}_{a_k} 'nin alınmasıyla, yaklaşık karışım vektörü $\mathbf{u}_{a_k}^q$ aşağıdaki gibi yazılır:

$$\mathbf{u}_{a_k}^q = \mathbf{x}_k^q \mathbf{S}_{a_k}^T (\mathbf{S}_{a_k} \mathbf{S}_{a_k}^T)^{-1}. \quad (5.49)$$

Daha sonra, karışım kod kitabının tüm satırlarından $\mathbf{u}_{a_k}^q$ aranarak en küçük Öklit uzaklığını veren k 'inci iterasyon indisi (h_k) bulunur.

$$h_k = \operatorname{argmin} \|\mathbf{u}_{a_k}^q - \mathbf{U}_l^{cb}\|, \quad l=1,2,\dots,Z, \quad (5.50)$$

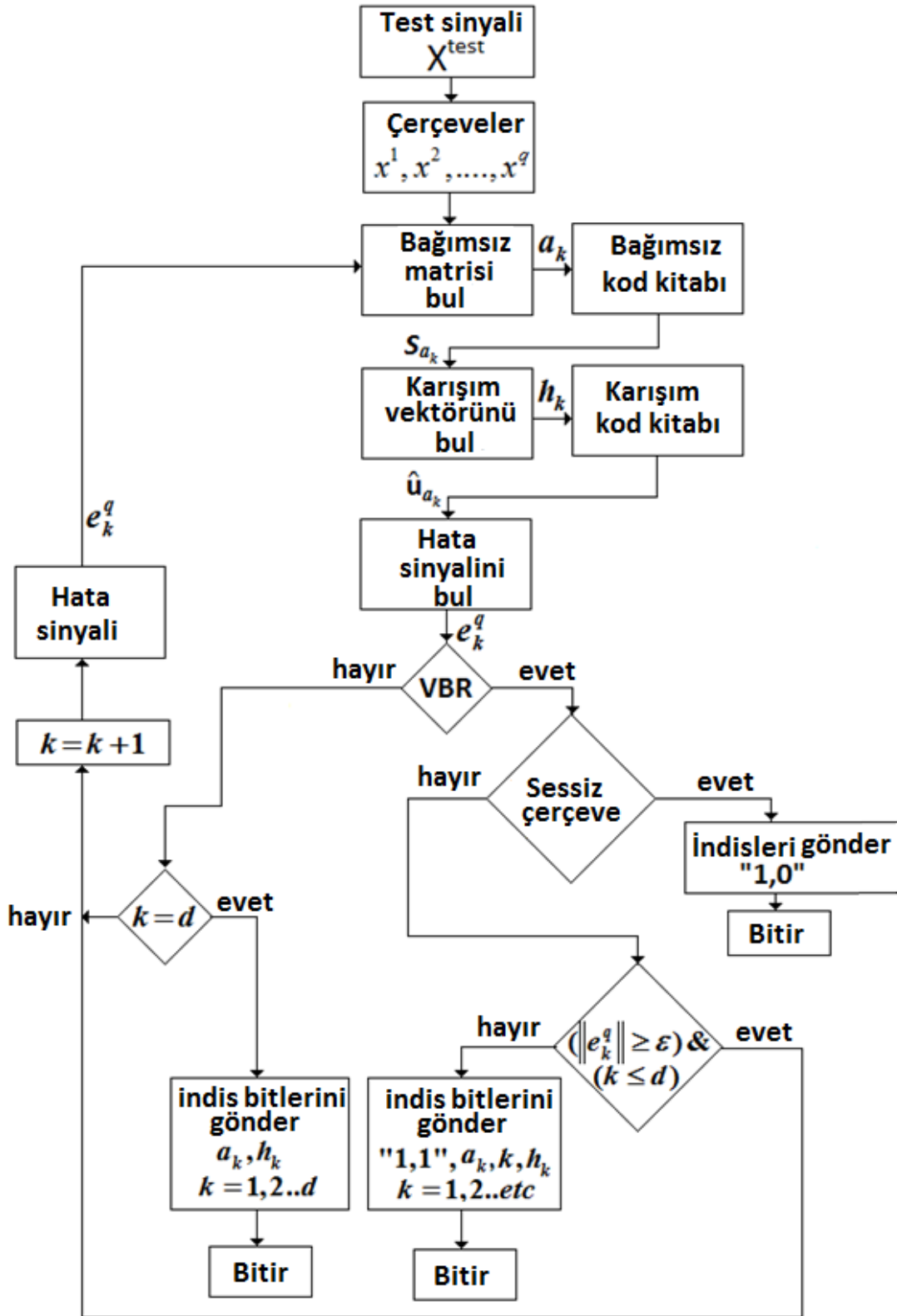
Burada \mathbf{U}_l^{cb} karışım kod kitabındaki l 'inci satır vektörüdür. \mathbf{U}^{cb} 'den $\mathbf{u}_{h_k} = \hat{\mathbf{u}}_k^q$ olarak seçildikten sonra, $\hat{\mathbf{x}}_k^q$ aşağıdaki gibi hesaplanır,

$$\hat{\mathbf{x}}_k^q = \mathbf{u}_{h_k} \mathbf{S}_{a_k}, \quad (5.51)$$

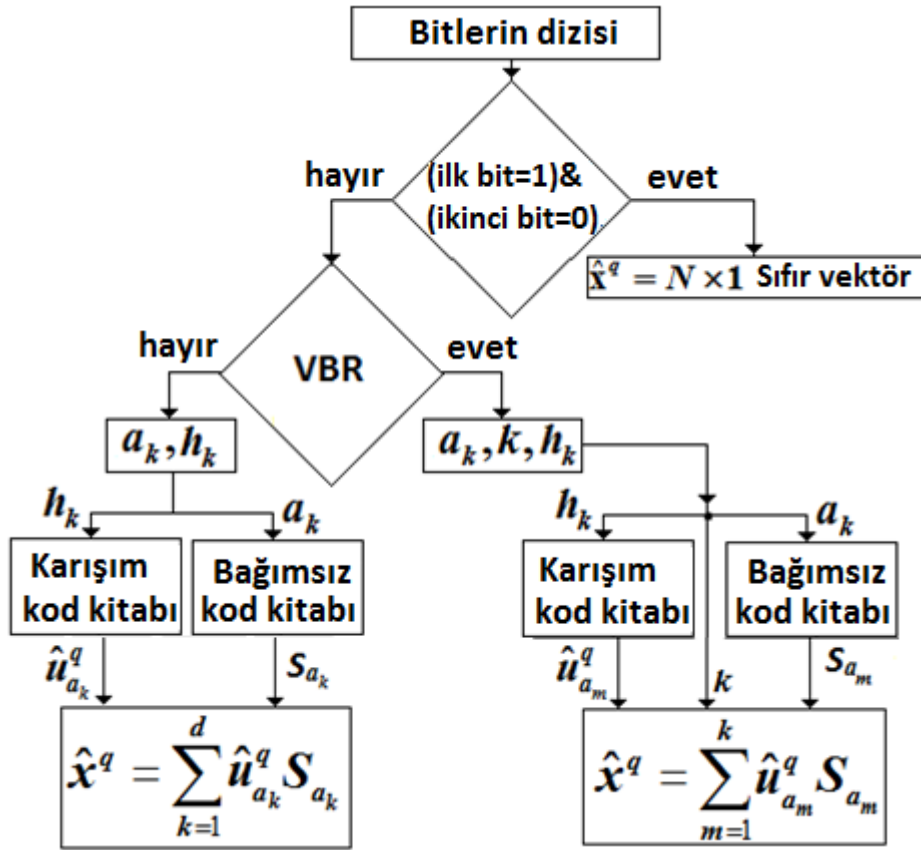
ve hata sinyali aşağıdaki gibi olur;

$$\mathbf{e}_k^q = \mathbf{x}_k^q - \hat{\mathbf{x}}_k^q. \quad (5.52)$$

Bu çalışmada, KLT- ve ICA- temelli yöntemler için aynı hata azaltma işlemleri gerçekleştirilmiştir. Önerilen ICA- temelli yöntem için kodlayıcı ve kod çözücü bölümler sırası ile Şekil 5.6'da ve Şekil 5.7'de verilmiştir.



Şekil 5.6. ICA temelli alt uzay yöntem için kodlayıcı blok şeması.



Şekil 5.7. ICA temelli yöntem için kod çözücü blok şeması.

5.3.4 Sabit ve değişken bit oranlı kodlama

Bu çalışmada, hem CBR hem de VBR kullanan kodlamalar, KLT ve ICA tabanlı alt uzay yöntemlerle gerçekleştirilmiştir. CBR kullanan kodlama, ses sinyalinin sesli ve sessiz çerçeveleri için aynı sayıda bit ataması yapmaktadır. CBR'nin aksine, VBR kullanan kodlamada ise her çerçeveye atanan bit sayısının değişmesine izin verilir. VBR ile, sesli olmayan çerçeveler için daha az bit ve sesli çerçeveler için daha fazla bit atayarak CBR'den daha yüksek kalitede ses sinyali sentezlemek mümkündür. Bu çalışmanın ana fikri, otokovaryans özvektörlerini kullanarak, yaklaşık olarak bir ses sinyalini sentezlemek ve ardından gerçek ve sentezlenmiş ses sinyalleri arasındaki hatayı sürekli azaltmaktır. İterasyon sayısı, hedef bit oranlarına göre ayarlanır. CBR ve VBR kullanan yöntemler için farklı hata azaltma algoritmaları kullanılmıştır. CBR kodlamada, kodlayıcı doğal olarak çıktı örneklerinin bit oranını hedeflemektedir. Eğer M^e (veya M^i) ve M^c (veya M^m) sırasıyla, özvektör (veya bağımsız bileşen) ve katsayı (veya karıştırma) kod kitaplarının boyutu ise $\log_2 M^e + \log_2 M^c$ (veya $\log_2 M^i + \log_2 M^m$) adet bitlere ihtiyacımız olacaktır.

Daha yüksek M^e ve M^c değerleri, daha düşük bir sıkıştırma oranı ile daha iyi kaliteye karşılık gelir. FL , çerçeve uzunluğu, d en fazla iterasyon sayısı ve FS , örnekleme frekansı ise, o zaman CBR kullanan kodlama için istenen bit oranı (kbps) şu şekilde tanımlanır:

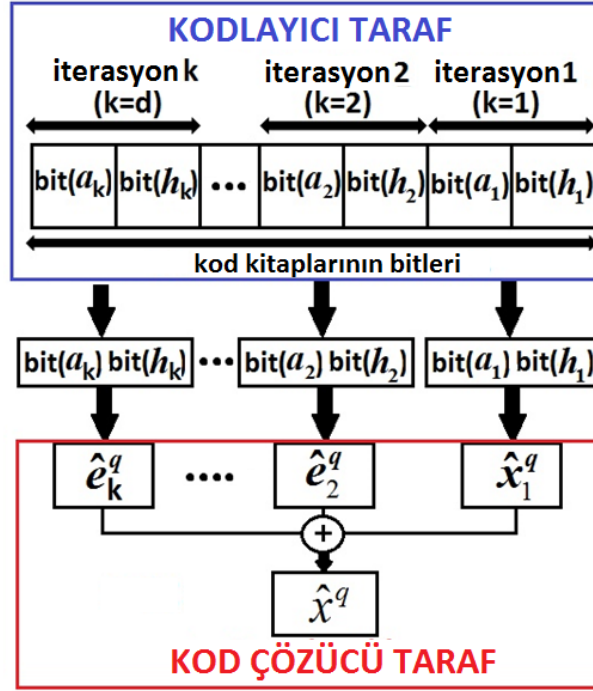
$$B_{CBR} = \frac{[tb \times FS] \times d}{FL}. \quad (5.53)$$

Burada tb , kod kitaplarının indislerini göndermek için gereken bit sayısıdır ve $\log_2 M^e + \log_2 M^c$ veya $\log_2 M^i + \log_2 M^m$ 'ye eşittir. Denklem 5.53'de d parametresi, en yüksek iterasyon sayısını temsil eden bir sabittir ve bu parametre hem kodlayıcı hem de kod çözücü tarafında bilinir. KLT_CBR ve ICA_CBR yöntemleri için bit atamaları Çizelge 5.1'de verilmiştir. NI , bu tabloda iterasyon sayısına karşılık gelir.

Çizelge 5.1. KLT_CBR ve ICA_CBR yöntemleri için bit atamaları

Çerçeve uzunluğu (Örnekler)	NI	Özvektör kod kitabı bitleri	Katsayı kod kitabı bitleri	Toplam bitler	Kbps
80	1	12 bit	12 bit	1x(12+12)=24	2.4
32	1	16 bit	16 bit	1x(16+16)=32	8
64	3	16 bit	17 bit	3x(16+17)=99	12.2
32	2	16 bit	16 bit	2x(16+16)=64	16
48	3	16 bit	17 bit	3x(16+17)=99	16.4
96	4	15 bit	15 bit	4x(15+15)=120	19.85

Kodlayıcı tarafta, birinci iterasyonda ($k = 1$), test sesine ait çerçeveler ile sentezlenmiş çerçeveler arasında bir hata sinyali bulunur ($\mathbf{e}_1^q = \mathbf{x}^q - \hat{\mathbf{x}}_1^q$). İkinci iterasyonda ($k = 2$), hata çerçevesi (\mathbf{e}_1^q) test çerçevesi olarak kullanılır ve $\hat{\mathbf{x}}_1^q$ bulmada kullanılan aynı adımları izleyerek $\hat{\mathbf{e}}_2^q$ bulunur. Üçüncü iterasyonda ($k = 3$), hata sinyali ($\mathbf{e}_2^q = \mathbf{x}^q - \hat{\mathbf{x}}_1^q - \hat{\mathbf{e}}_2^q$) test çerçevesi olarak kullanılır ve $\hat{\mathbf{e}}_3^q$ bulunur. İstenen bit oranına ulaşıldığında işlem durdurulur ($k = d$). Önerilen KLT_CBR ve ICA_CBR yöntemleri için kod çözücü taraftaki bir test çerçevesinin sentezi ve bit atamaları Şekil 5.8'de gösterilmektedir.



Şekil 5.8. Önerilen KLT_CBR ve ICA_CBR için kodlayıcı-kod çözücü kısımlardaki bit atama yapısı

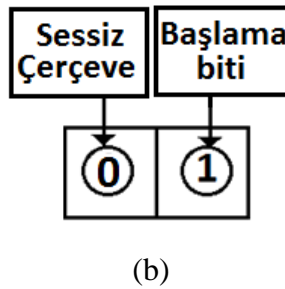
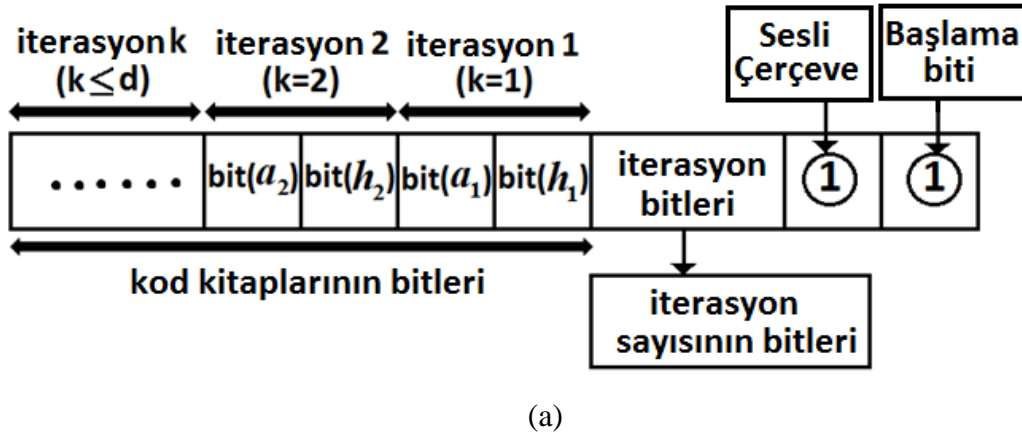
Şekil 5.8'de, $\text{bit}(a_1), \dots, \text{bit}(a_k)$ ve $\text{bit}(h_1), \dots, \text{bit}(h_k)$ sırasıyla özvektör (veya bağımsız bileşen) kod kitabı ve katsayı (veya karışım) kod kitabına karşılık gelen bit atamalarını göstermektedir. CBR kullanan kodlama için, mevcut iterasyonun (k) ve kod kitabının indisleri, test sinyalinin tüm çerçeveleri için kod çözücü tarafa iletilir ve test sinyalleri, kod çözücü taraftaki aynı kod kitapları kullanılarak sentezlenir.

VBR kodlamada, önceden belirlenmiş hata eşiği (ε) ve izin verilen en yüksek iterasyon sayısı (d) istenilen bit oranlarını elde etmek için dinamik olarak belirlenir. Şekil 5.8'de verilen işlem, aynı zamanda VBR kodlamasında da kullanılır, ancak VBR kodlaması için bit ataması, bir çerçevenin sesli veya sessiz olmasına göre farklılık gösterir. Ardından algoritma, hata sinyalinin norm değeri önceden tanımlanmış bir eşik değerinden (ε) daha düşük olana kadar ya da geçerli iterasyon indisi (k), en yüksek iterasyon değeri (d) 'ye eşit olana kadar devam eder:

$$\|\mathbf{x}_k^{test} - \mathbf{x}_k^{syn}\| < \varepsilon \text{ veya } k=d, \quad (5.54)$$

burada \mathbf{x}_k^{test} ve \mathbf{x}_k^{syn} sırası ile, k 'inci iterasyon için test and sentezlenmiş sinyal çerçeveleridir.

VBR kullanan kodlamada, mevcut çerçevenin sesli veya sessiz olup olmadığının kod çözücüde belirlenebilmesi için ek bir bit kullanılır. Şekil 5.9'da, toplam bit dizisinin ilk biti daima başlangıç bitidir. Çerçeve sesli ise, dizinin ikinci biti '1' olur, aksi halde '0' olur. Bu iki bit, çerçevenin sessiz ya da sesli olup olmadığını belirtir. q 'inci çerçevenin hata sinyalinin norm değeri, birinci iterasyon için ε_{um} 'den düşükse, bu çerçeve sessiz bir çerçeve olarak değerlendirilir. Aksi halde, q 'inci çerçeve sesli olarak değerlendirilir. ε_{um} parametresi, sesli olmayan çerçeveler için önceden tanımlanmış bir eşik değeridir ($\varepsilon_{um} < \varepsilon < 1$). Sesli olmayan çerçeveler için, toplam bit dizisi yalnızca "0 1" bitlerini içerir ve kod çözücü tarafta çerçevenin boyutunda olan bir sıfır vektörü oluşturulur. Kodlayıcı tarafta, her sesli çerçeve için farklı iterasyon sayısı kullanılabilir. Bu nedenle, iterasyonların sayısı kod çözücü tarafta da bilinmelidir. Şekil 5.9'da, sesli çerçevelerin bit yapısı için iterasyon sayısına karşılık gelen bitler gösterilmektedir.



Şekil 5.9. Önerilen KLT_VBR ve ICA_VBR için (a) sesli ve (b) sessiz çerçevelere ait bit dizisi yapıları.

VBR kullanan kodlama için istenilen bit oranı (kbps) aşağıdaki şekilde tanımlanır:

$$B_{VBR} = \frac{2 \cdot f_{uv} + \sum_{y=1}^{f_v} k_y \cdot (tb+2) + f_v \cdot in}{(f_{uv} + f_v)} \times \frac{FS}{FL}, \quad (5.55)$$

Burada k_y , y 'inci sesli çerçeve için kullanılan iterasyon sayısıdır ($1 \leq k_y \leq d, y=1, 2, \dots, f_v$). f_v ve f_{uv} sırasıyla sesli ve sessiz çerçevelerin sayısıdır. Denklem 5.55'de, y ise test sinyali içerisindeki sesli çerçeve sayısıdır ve in , bir sesli çerçeve için kullanılan iterasyon sayısına karşılık gelen bitlerin sayısıdır. VBR kullanan kodlamada ε , ε_{min} ve d parametreleri için bit ataması yapılmamaktadır. Çünkü bu değerler önceden deneysel olarak bulunmuş olup, kodlayıcı ve kod çözücü kısımlarında bilinmektedir. Çizelge 5.2'de, önerilen yöntemler KLT_VBR ve KLT_CBR için farklı bit oranlarındaki bit ataması verilmiştir.

Çizelge 5.2. KLT_VBR ve KLT_CBR için farklı bit oranlarındaki bit ataması

Çerçeve örnek sayısı	Özvektör kod kitabı bitleri	Katsayı kod kitabı bitleri	İterasyon sayıları	İterasyon bitleri	Sesli çerçeve bitleri	Sessiz çerçeve bitleri	Çerçeve başına düşen bitler	Kbps
80	12	12	$1 < K < 2$	1	$K \times (12+12)$	2	24	2.4
32	16	16	$1 < K < 2$	1	$K \times (16+16)$	2	32	8
64	16	16	$1 < K < 4$	2	$K \times (16+16)$	2	99	12.2
32	16	16	$1 < K < 3$	2	$K \times (16+16)$	2	64	16
48	16	16	$1 < K < 4$	2	$K \times (16+16)$	2	99	16.4
96	15	15	$1 < K < 5$	3	$K \times (15+15)$	2	120	19.85

ICA tabanlı yöntem, Çizelge 5.2'de verilen bit atama yapısına sahiptir. TIMIT veri tabanında kullanılan eğitim kümesi, özvektör kod kitaplarının boyutlarını belirlemek için kullanılır. Aynı kod kitapları, CBR ve VBR kodlaması için kullanılır. VBR ve CBR arasındaki fark, VBR'nin sesli çerçeveler için CBR'den daha fazla iterasyon içermesi ve sessiz çerçeveler için sadece bir kez iterasyon yapmasıdır.

5.4 Ses Kodlayıcıların Kalite Ölçütleri

Ses kodlayıcıların nesnel kalitesi ayrı ayrı değerlendirilir. Bu amaçla, PESQ (Ses Kalitesinin Algısal Değerlendirmesi), WSS (Ağırlıklandırılmış Spektral Eğim) ve LLR (Logaritmik Tahmin Oranı) gibi nesnel ölçütler kullanılır (Hu ve Loizou, 2008;

Krishnamoorthy, 2011). PESQ, algılanan ses kalitesinin nesnel olarak öngörülmesi için bir test metodu olup, telekomünikasyon ve IP ağlarında yaygın olarak kullanılmaktadır. PESQ yöntemini geliştiren araştırmacılar, kişisel ölçümlerle bulunan sonuçlar ile PESQ'nun en yüksek ilintiye sahip olduğunu iddia etmektedir (Goudarzi ve Sun, 2009). Ortalama görüş puanı (MOS), kişisel bir değerlendirme ölçütü olarak telefon ağlarındaki ses kalitesinin sayısal bir ölçümünü sağlar. MOS değerleri, insan duyu sistemini kullanan kişisel testlerden elde edilir.

5.4.1 Öznel kalite ölçütü

Öznel ölçümler, kişilere, orijinal ses sinyalleri ile sentezlenmiş ses sinyalleri dinletilerek gerçekleştirilir (Osahenvemwen, 2015). Bu metot “MOS” olarak adlandırılır ve Mutlak Kategori Derecelendirme (ACR) metodudur. Bu test, Uluslararası Telekomünikasyon Birliği (ITU-T P.800) tarafından standartlaştırılmıştır. ITU-T P.800 standardı, özellikle dar bantlı ses kodlama uygulamaları için ses kalitesini değerlendirme, veri tabanı toplamak ve oluşturmak için geliştirilmiştir. ACR testinde, dinleyiciler, konuşma kalitesini mükemmel (5), iyi (4), orta (3), düşük (2), kötü (1) olmak üzere beş puanlık ölçeği kullanarak değerlendirir. Değerlendirmelerin ortalaması alınarak bir MOS değeri bulunur (Osahenvemwen, 2015). Bir MOS testi, genellikle birkaç saniyelik bir ses akışını tek tek dinleyen 12 ile 24 arası katılımcı içermelidir. 4,5-5 arası bir MOS değeri, mükemmel bir ses kalitesine karşılık gelmektedir; 4 civarı bir değer ise, makul derecede iyi ve kabul edilebilirdir. Çizelge 5.3’de, MOS değerlerine karşılık gelen ses kalitesi dereceleri verilmiştir.

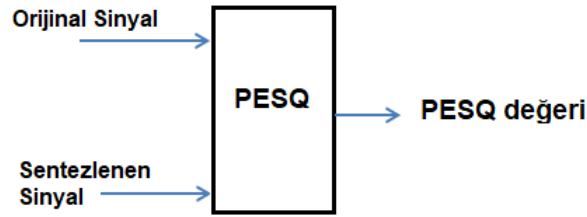
Çizelge 5.3. MOS değerleri ve karşılık gelen ses kalitesi dereceleri

MOS DEĞERLERİ	DERECELENDİRME
5	Mükemmel
4	İyi
3	Orta
2	Düşük
1	Kötü

MOS testleri, ses kalitesini ölçmek için en güvenilir yöntem olmasına rağmen bunu yapmak zaman alıcı, zahmetli ve pahalıdır. Bu yüzden MOS testlerini gerçekleştirmek zordur.

5.4.2 Ses kalitesinin algısal değerlendirilmesi (PESQ)

PESQ, nesnel bir ITU-T standartıdır. MOS testlerinin yapılması pahalı ve çok zaman gerektirdiği için MOS testi yerine kullanabilmek için geliştirilmiştir. PESQ testi ile bulunan PESQ değeri öznel MOS değerine çevrilir. PESQ testi için orijinal ve sentezlenmiş ses sinyalleri kullanılır. Bu sinyallere PESQ algoritması uygulanır ve çıkışta bir PESQ değeri elde edilir. Şekil 5.10'da, PESQ değerinin bulunması gösterilmiştir.



Şekil 5.10. PESQ algoritması uygulanarak PESQ değerinin elde edilmesi

PESQ değeri, 1 ile 4.5 aralığında tek bir sayı olan MOS benzeri bir ölçekle eşleştirilir (Kumar ve ark., 2014). PESQ değeri ile MOS değeri arasındaki korelasyonun hesaplanması için korelasyon katsayısı aşağıda verilen Pearson formülü ile hesaplanır;

$$r = \frac{(x_i - \hat{x})(y_i - \hat{y})}{\sqrt{\sum (x_i - \hat{x})^2 \sum (y_i - \hat{y})^2}}, \quad (5.56)$$

buradaki PESQ değerleri, MOS değerlerine haritalanmış haldeki değerlerdir. Denklem 5.56'da, x_i , i 'inci MOS değeri; \hat{x} , ortalama MOS değerini; y_i i 'inci PESQ değeri ve \hat{y} ortalama PESQ değeridir. PESQ değerlerinin MOS değerine haritalanması aşağıdaki denklem ile gerçekleştirilir;

$$y = 0.999 + \frac{4.999 - 0.999}{1 + e^{-1.4945x + 4.6607}}, \quad (5.57)$$

burada x PESQ değeridir. Ayrıca MOS değeri (y) kullanılarak, PESQ değeri aşağıdaki denklem ile elde edilir;

$$x = \frac{4.6607 - \ln\left(\frac{4.999 - y}{y - 0.999}\right)}{1.4945}. \quad (5.58)$$

5.4.3 Ağırlıklandırılmış spektral eğim (WSS)

WSS ölçüsü, işitsel bir modele dayalı bir frekans düzlemi değeridir. WSS ölçüsü aşağıdaki gibi (Krishnamoorthy, 2011) tanımlanmıştır;

$$WSS = \frac{1}{M} \sum_{m=0}^{M-1} \frac{\sum_{j=1}^K WSS(j, m) (S_o(j, m) - S_s(j, m))^2}{\sum_{j=1}^K WSS(j, m)}, \quad (5.59)$$

burada $WSS(j, m)$, j 'inci frekans bandı için hesaplanan ağırlıklardır (Krishnamoorthy, 2011); K değeri, 25 olarak alınır ve M , veri parçalarının sayısıdır. Denklem 5.59'da $S_o(j, m)$ ve $S_s(j, m)$, j 'inci frekans bandı için sırasıyla orijinal ve işlenmiş ses sinyallerinin spektral eğimleridir.

5.4.4 Logaritmik tahmin oranı (LLR)

LLR ölçüsü, LPC temelli nesnel kalite ölçütlerinden biridir ve spektral zarfların benzerliği ile ilgilidir. Her 20 ms'lik çerçeve için LLR, (Hu ve Loizou, 2008) aşağıdaki gibi tanımlanmıştır;

$$LLR(\mathbf{a}_o, \mathbf{a}_s) = \log\left(\frac{\mathbf{a}_s \mathbf{R}_o \mathbf{a}_s}{\mathbf{a}_o \mathbf{R}_o \mathbf{a}_o}\right), \quad (5.60)$$

burada \mathbf{a}_o ve \mathbf{a}_s , sırasıyla orijinal ve işlenmiş ses çerçevelerinin doğrusal öngörü katsayı vektörleri ve \mathbf{R}_o , orijinal ses çerçevesinin oto-korelasyon matrisidir.

5.4.5 Birleşik ölçüt

Geleneksel nesnel yöntemler, Sinyal Gürültü Oranı (SNR) ve genel ses kalitesi açısından yüksek korelasyon sağlayamayabilirler. Bu nedenle farklı nesnel kalite ölçütlerini birleştirerek, birleşik bir nesnel ölçüm gerçekleştirilebilir (Hu and Loizou, 2008). Bu birleşik ölçüt ile daha doğru bir ölçüm yapabilmek için, parçalı (segmental) SNR, Ağırlıklandırılmış Spektral Eğim (WSS), LLR ve PESQ gibi dört temel ölçütün birleşimleri kullanılır (Hu and Loizou, 2008). Üç temel ölçüm çeşidi vardır.

- 1) Sinyaldeki bozulmayı ölçmek için **Csig**,
- 2) Gürültü bozulmasını ölçmek için **Cbak**,
- 3) Genel ses bozulmasını ölçmek için **Cov**.

Bu üç ölçüm değeri, nesnel ölçütlerin doğrusal birleşimleri ile elde edilir.

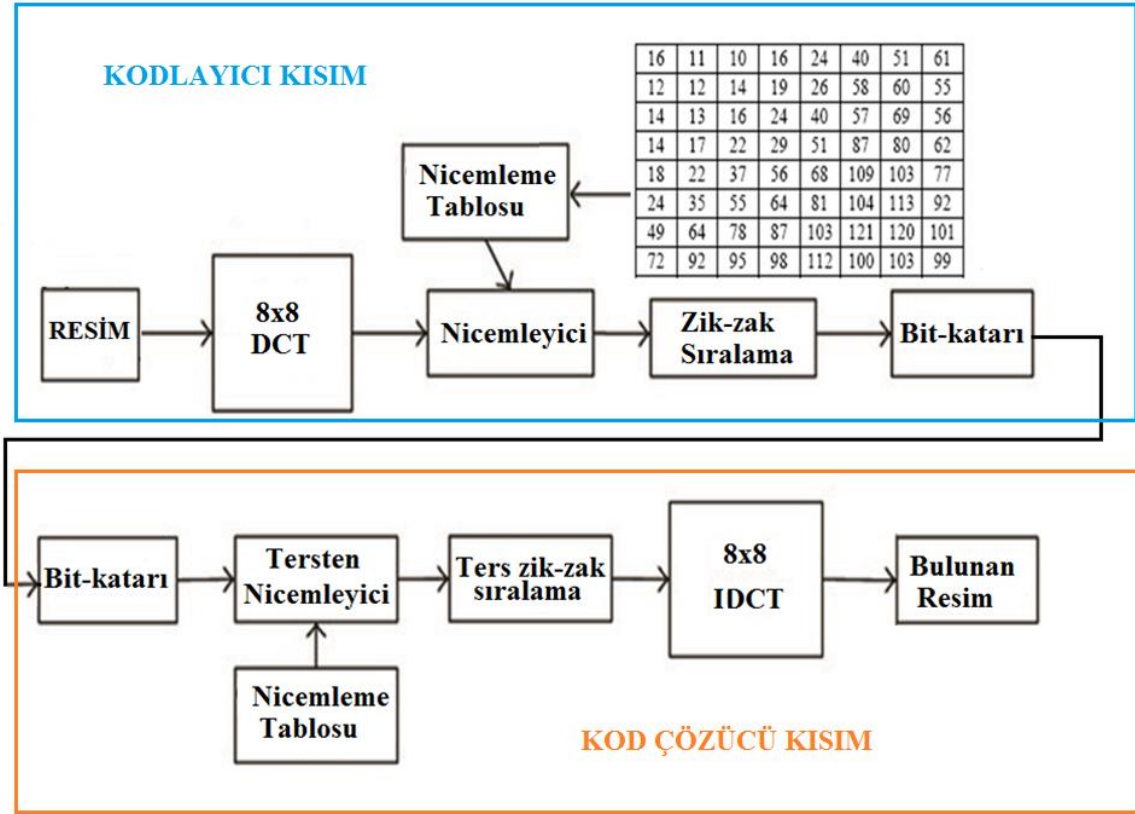
$$C_{sig} = 3.093 - 1.029 \cdot LLR + 0.603 \cdot PESQ - 0.009 \cdot WSS \quad (5.61)$$

$$C_{bak} = 1.634 + 0.478 \cdot PESQ - 0.007 \cdot WSS + 0.063 \cdot segSNR \quad (5.62)$$

$$C_{ov} = 1.594 + 0.805 \cdot PESQ - 0.512 \cdot LLR - 0.007 \cdot WSS \quad (5.63)$$

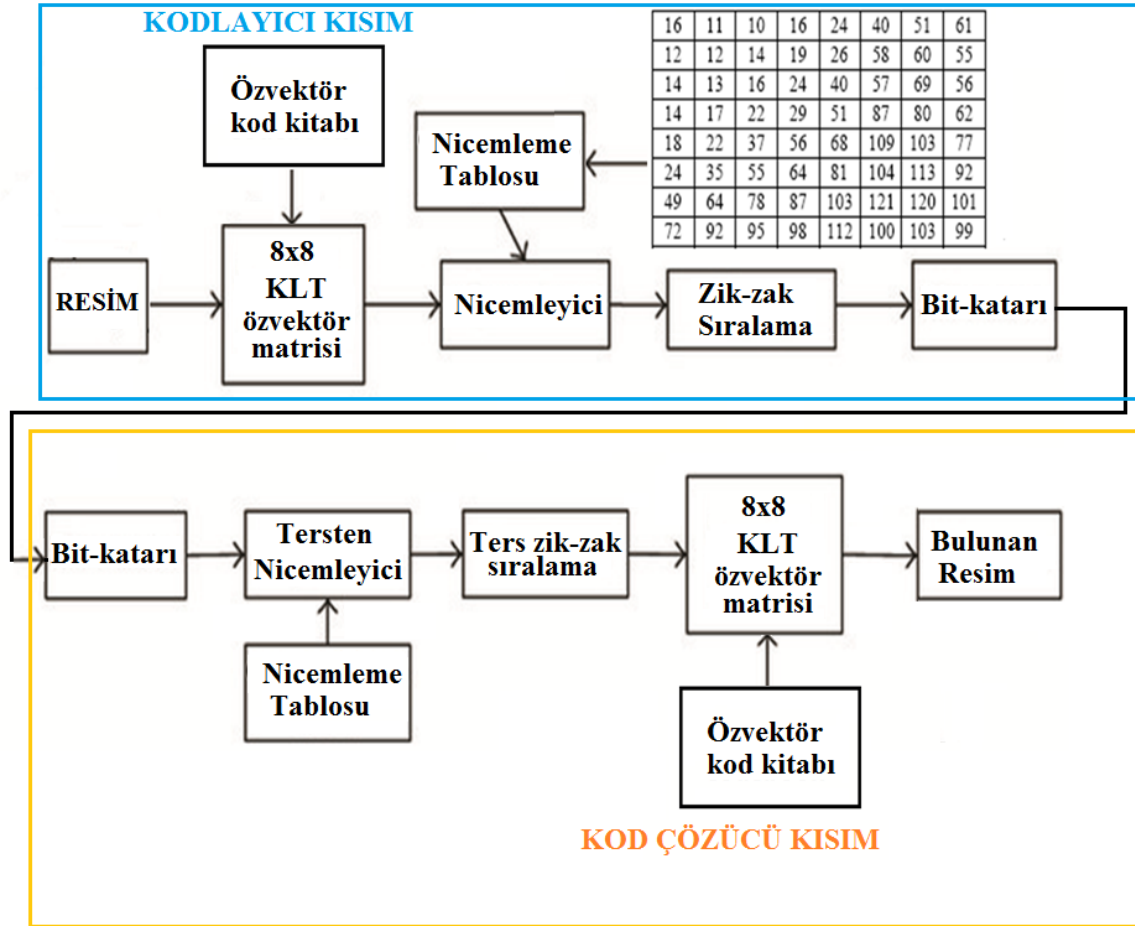
5.5 KLT Temelli Resim Sıkıştırma

Bu tez çalışmasında, ses sinyallerinin sıkıştırılmasının dışında resim kodlamada gerçekleştirilmiştir. KLT temelli resim kodlama için, bölüm 5.3.2'de anlatılan özvektör nicemleme yöntemi kullanılmıştır. Böylece resmi kodlamak için gerekli süre azaltılmıştır. Ayrıca, eğitim kümesinden elde edilen farklı sayıda nicemlenmiş özvektör matrisi için PSNR değeri bulunmuştur. KLT temelli resim kodlama için oluşturulan algoritma JPEG algoritmasının benzerini kullanmaktadır. Şekil 5.11'de resim kodlama için kullanılan JPEG blok diyagram gösterilmiştir.



Şekil 5.11. JPEG için kodlayıcı ve kod çözücü blok diyagram

KLT temelli algoritmanın JPEG algoritmasından tek farkı, 8×8 'lik sabit DCT matrisi yerine, 8×8 'lik N adet ($N=1,2,4,\dots,32$) KLT özvektör matrisi kullanmasıdır. Bu özvektör matrislerinden elde edilen taban resimlerinin (basis images) en uygunları kullanılarak test resimleri yaklaşık olarak elde edilir. Şekil 5.12'de, KLT temelli resim kodlama için, kodlayıcı ve kod çözücü blok diyagramı gösterilmiştir.



Şekil 5.12. KLT ile resim kodlama için kodlayıcı ve kod çözücü blok diyagramı

Toplam 40 adet resimden, 32 tanesi eğitim ve 8 tanesi test için kullanılmıştır. Üç farklı özvektör nicemleme yöntemi denenmiştir. Bu yöntemlerden ilki (Y1), 32 adet eğitim resminden elde edilen tek adet özvektör matrisi ile oluşturulmuştur. Bu yöntemde nicemleme yapılmayıp, 32 resmin her 8×8 bloğu için, tek adet özvektör matrisi bulunmuştur. Bu yöntem ile (Y1), 8 adet test resmi için bulunan PSNR değerleri Çizelge 5.4'te verilmiştir. Çizelge 5.4'de KLT-CB, kod kitabındaki özvektör sayısını göstermektedir.

Çizelge 5.4. Birinci yöntem (Y1) ile bulunan PSNR değerleri

RESİM	KLT-CB	KLT-PSNR	JPEG-PSNR
Test Resmi-1	1	25,32	25,23
Test Resmi-2	1	21,36	21,33
Test Resmi-3	1	31,43	31,55
Test Resmi-4	1	25,42	25,38
Test Resmi-5	1	27,14	27,12
Test Resmi-6	1	30,30	30,40
Test Resmi-7	1	31,69	31,84
Test Resmi-8	1	30,52	30,54
Ortalama		27,52	27,55

Kullanılan ikinci yöntem (Y2), bölüm 5.3.2’de bahsedilen nicemleme yöntemi ile, 32 adet normalize edilmiş KLT özvektör matrisi elde edilerek oluşturulmuştur. Test resimleri için, bu özvektör matrisleri kullanılarak, yaklaşık resimler ikinci yöntem ile elde edilmiş ve bulunan KLT ve JPEG- PSNR değerleri Çizelge 5.5’de verilmiştir.

Çizelge 5.5. İkinci yöntem (Y2) ile bulunan PSNR değerleri

RESİM	KLT-CB	KLT-PSNR	JPEG-PSNR
Test Resmi-1	32	31,34	32,95
Test Resmi-2	32	21,12	21,45
Test Resmi-3	32	29,98	31,23
Test Resmi-4	32	26,92	27,59
Test Resmi-5	32	25,75	26,1
Test Resmi-6	32	33,01	34,7
Test Resmi-7	32	26,11	30,15
Test Resmi-8	32	28,34	29,52
Ortalama		27,82	29,21

Üçüncü yöntemde ise (Y3), bölüm 5.3.2’de bahsedilen üçüncü yaklaşımdaki (A3) algoritma kullanılarak, 32 normalize edilmiş özvektör matrisine ait küme ve her kümeye ait 32 adet özvektör matristen oluşan kod kitabı kullanılmıştır. Bu kod kitabı ile bulunan sonuçlar Çizelge 5.6’da verilmiştir.

Çizelge 5.6. Üçüncü yöntem (Y3) ile bulunan PSNR değerleri

RESİM	KLT-CB	KLT-PSNR	JPEG-PSNR
Test Resmi-1	32x32	31,01	33,88
Test Resmi-2	32x32	21,58	21,70
Test Resmi-3	32x32	30,23	31,92
Test Resmi-4	32x32	27,21	27,86
Test Resmi-5	32x32	25,58	26,04
Test Resmi-6	32x32	31,24	33,47
Test Resmi-7	32x32	25,72	27,44
Test Resmi-8	32x32	26,55	27,78
Ortalama		27,51	28,90

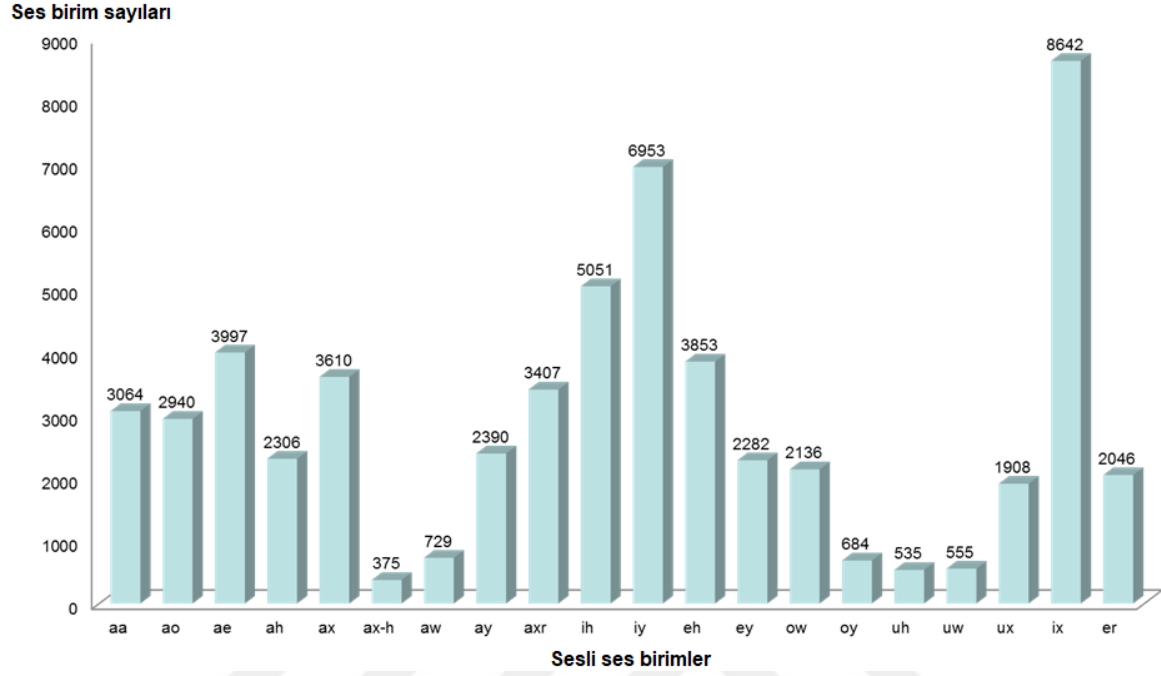
6. BULGULAR VE TARTIŞMA

Bu tez çalışmasında, TIMIT (Texas Instruments (TI) and Massachusetts Institute of Technology (MIT)) ses veri tabanı kullanılmıştır. TIMIT ses veri tabanı, 1990 yılında oluşturulmuştur. TIMIT, farklı şive ve cinsiyetlerdeki seslendiricilerin söylediği fonetik ve sözlüksel bir ses veri tabanıdır (Garofolo vd., 1990). TIMIT veri tabanının ses materyalleri, eğitim ve test kümelerine bölünmüştür. Eğitim kümesinde, 462 seslendiricinin söylediği 10 adet cümle birlikte toplam 4620 cümle bulunmaktadır. Test kümesi ise, 168 seslendiricinin söylediği 8 adet cümle ile birlikte toplam 1344 cümleden oluşmaktadır. Veri tabanındaki ses sinyalinin örnekleme frekansı 16 kHz'dir (16 bit, PCM). TIMIT ses veri tabanı 61 adet ses birimi (fonem) içermektedir (Zue vd., 1990). TIMIT, daha çok fonetik bilgi ve otomatik konuşma tanıma için tasarlanmıştır. Genel olarak; otomatik ses tanıma sistemlerinde (Ney ve Ortmanns, 2000), çeviri, dikte etme, komut ile kontrol etme gibi alanlarda kullanılır. Rakam tanıma ve ses birimi (fonem) tanıma alanlarında da pek çok çalışma yapılmıştır. Şekil 6.1'de TIMIT veri tabanında bulunan 61 adet ses birimi listelenmiştir.

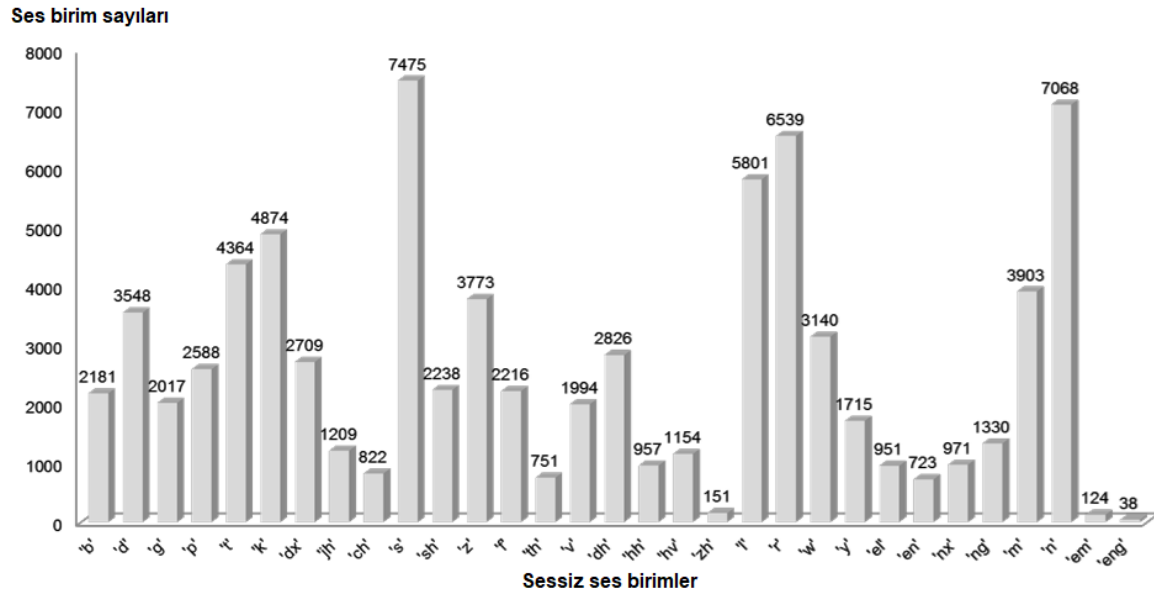
Numara	Ses birim	Numara	Ses birim	Numara	Ses birim
1	iy	22	ch	43	en
2	ih	23	b	44	eng
3	eh	24	d	45	l
4	ey	25	g	46	r
5	ae	26	p	47	w
6	aa	27	t	48	y
7	aw	28	k	49	hh
8	ay	29	dx	50	hv
9	ah	30	s	51	el
10	ao	31	sh	52	bcl
11	oy	32	z	53	dcl
12	ow	33	zh	54	gcl
13	uh	34	f	55	pcl
14	uw	35	th	56	tcl
15	ux	36	v	57	kcl
16	er	37	dh	58	q
17	ax	38	m	59	pau
18	ix	39	n	60	epi
19	axr	40	ng	61	h#
20	ax-h	41	em		
21	jh	42	nx		

Şekil 6.1. TIMIT veri tabanında bulunan 61 adet ses birimi

Şekil 6.2 ve Şekil 6.3'te sırasıyla, TIMIT veri tabanına ait sesli ve sessiz ses birimlerinin sayıları gösterilmektedir.



Şekil 6.2. TIMIT veri tabanına ait sesli ses birimlerinin sayıları



Şekil 6.3. TIMIT veri tabanına ait sessiz ses birimlerinin sayıları

TIMIT ses veri tabanındaki 61 adet ses biriminden söyleyişleri birbirine yakın olan ses birimleri birleştirilerek, 45 adet ses birim oluşturulmuştur. Bu ses birimleri aşağıdaki listede belirtilmiştir.

{sh, ix, hv, eh, jh, ih, dcl, d, ah, kcl, k, s, ux, q, en, g, r, w, ao, dx, axr, l, y, uh, n, ae, m, oy, ax, dh, ey, v, hv, iy, aa, t, hh, ch, el, z, pcl, ng, er, uw}.

Birleştirme işlemi, TIMIT veri tabanındaki ses birimi sayıları dikkate alınarak gerçekleştirilmiştir. Eğitim aşamasında, en az sayıya sahip ses birimleri dikkate alınarak, kullanılacak her ses birimine ait bir sayı sınırı seçilmiştir.

Deneysel çalışmalarda örnekleme frekansı, aşağı örnekleme işlemi ile 8 kHz'e dönüştürülmüştür. Sadece 19.85 kbps'de, 16 kHz'lik örnekleme frekansı kullanılmıştır. Eğitim aşamasında, her ses birimi sınıfı için (15 ila 100 arasında değişen) eşit sayıda ses birim kullanılmıştır. Eğitim aşamasında kullanılan her ses birimi, 8 kHz ve 16 kHz'de sırasıyla 960 ve 1920 örnek uzunluğuna sahiptir. Ardından, 960 ve 1920 örneklilik ses birimi uzunlukları çerçevelere bölünmüştür. Eğitim aşamasında, ICA ve KLT tabanlı yöntemler için, çeşitli çerçeve uzunlukları kullanılarak veri matrisleri oluşturulmuştur. Test sinyalleri için, TIMIT veri tabanının test kümesinden rastgele seçilen 30 söyleyiş kullanılmıştır. Her söyleyiş için MOS, PESQ ve Cov değerleri bulunmuştur. MOS gibi kişisel dinleme testlerinin gerçekleştirilmesi zor ve zaman alıcı bir süreçtir. Bu yüzden kişisel dinleme testleri, yaşları 20-26 arasında değişen 10 adet öğrenci ile yaşları 30-60 arasında değişen 10 adet akademisyen dinleyiciler ile gerçekleştirilmiştir. Ardından, 30 adet söyleyiş için, bu MOS, PESQ ve Cov değerlerinin ortalamaları hesaplanmıştır.

Önerilen KLT ve ICA tabanlı yöntemlerin etkinliğini analiz etmek için çoklu testler yapılmıştır. Deneysel çalışmalarda, KLT_CBR ve KLT_VBR sırasıyla, KLT için CBR ve VBR kodlamasına karşılık gelir. Benzer şekilde, ICA_CBR ve ICA_VBR sırasıyla, ICA için CBR ve VBR kodlamasına karşılık gelir. İlk sentezleme yaklaşımı için (A1), PESQ ve Cov'un ortalama değerleri Çizelge 6.1 ile Çizelge 6.8 arasında gösterilmiştir. 2 ms'lik çerçeve uzunluğu, 16 kbps'lik KLT_VBR için ve 8 kbps'lik KLT_CBR için kullanılmaz. Çünkü, iterasyon (yineleme) sayısı birden fazla olduğunda, 8 kbps ve 16 kbps değerleri aşılmaktadır. Buna ek olarak, 2 ms ve 4 ms'lik çerçeve uzunlukları da, 8 kbps'lik

KLT_VBR yöntemi için benzer sebepten kullanılmamıştır. Ses sinyalleri, tüm farklı çerçeve uzunlukları için en büyük K adet özdeğerlere karşılık gelen özvektörler kullanılarak sentezlenir ($K = 1,2,3,\dots, 10$).

Çizelge 6.1. 16 kbps için PESQ değerleri (KLT_VBR (A1))

FL (ms)	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
4	3,07	3,32	3,52	3,68	3,74	3,79	3,74	3,71	3,66	3,61
6	3,11	3,37	3,53	3,65	3,70	3,88	3,97	3,75	3,73	3,65
8	2,96	3,21	3,38	3,44	3,52	3,73	3,62	3,57	3,53	3,51
10	2,83	3,00	3,14	3,27	3,30	3,34	3,35	3,30	3,28	3,26

Çizelge 6.2. 16 kbps için Cov değerleri (KLT_VBR (A1))

FL (ms)	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
4	3.80	4.06	4.26	4.41	4.47	4.51	4.46	4.44	4.39	4.35
6	3.82	4.13	4.28	4.35	4.46	4.58	4.62	4.47	4.45	4.39
8	3.69	3.93	4.11	4.17	4.29	4.46	4.40	4.37	4.32	4.30
10	3.57	3.74	3.87	4.01	4.09	4.12	4.15	4.05	4.02	4.00

Çizelge 6.3. 16 kbps için PESQ değerleri (KLT_CBR (A1))

FL (ms)	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
2	2,90	3,22	3,53	3,57	3,37	3,11	2,89	2,7	2,62	2,48
4	2,85	3,08	3,26	3,52	3,50	3,60	3,65	3,48	3,41	3,27
6	2,84	3,03	3,21	3,59	3,66	3,76	3,82	3,70	3,63	3,61
8	2,77	2,98	3,16	3,23	3,49	3,32	3,27	3,25	3,23	3,19
10	2,74	2,90	3,00	3,13	3,27	3,22	3,16	3,14	3,12	3,08

Çizelge 6.4. 16 kbps için Cov değerleri (KLT_CBR (A1))

FL (ms)	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
2	3,65	3,98	4,27	4,29	4,14	3,90	3,70	3,51	3,41	3,26
4	3,59	3,81	4,02	4,25	4,24	4,33	4,37	4,21	4,15	4,02
6	3,57	3,78	3,96	4,27	4,33	4,41	4,49	4,38	4,29	4,26
8	3,52	3,76	3,93	4,01	4,19	4,09	4,03	4,01	3,98	3,93
10	3,50	3,67	3,76	3,91	3,98	3,99	3,94	3,92	3,90	3,80

Çizelge 6.5. 8 kbps için PESQ değerleri (KLT_VBR (A1))

FL (ms)	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
6	2,66	2,86	2,95	3,11	3,23	3,37	3,50	3,39	3,27	3,26
8	2,65	2,84	3,00	3,05	3,08	3,13	3,13	3,15	3,13	3,12
10	2,68	2,81	2,89	2,98	2,99	3,04	3,01	3,03	3,01	2,98

Çizelge 6.6. 8 kbps için Cov değerleri (KLT_VBR (A1))

FL (ms)	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
6	3,22	3,55	3,68	3,79	3,92	4,02	4,10	4,03	3,87	3,86
8	3,32	3,52	3,67	3,74	3,78	3,83	3,83	3,85	3,85	3,83
10	3,34	3,49	3,58	3,67	3,69	3,74	3,71	3,72	3,71	3,70

Çizelge 6.7. 8 kbps için PESQ değerleri (KLT_CBR (A1))

FL (ms)	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
4	2,64	2,82	2,98	3,12	3,20	3,09	2,99	2,87	2,78	2,73
6	2,6	2,78	2,92	3,01	3,03	3,08	3,10	3,10	3,08	3,06
8	2,51	2,67	2,82	2,9	2,88	2,86	2,79	2,68	2,66	2,61
10	2,22	2,32	2,41	2,39	2,39	2,36	2,38	2,26	2,26	2,21

Çizelge 6.8. 8 kbps için Cov değerleri (KLT_CBR (A1))

FL (ms)	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
4	3,21	3,41	3,6	3,82	3,87	3,78	3,68	3,59	3,5	3,46
6	3,37	3,58	3,66	3,72	3,75	3,78	3,80	3,81	3,8	3,78
8	3,15	3,36	3,51	3,6	3,61	3,59	3,47	3,39	3,31	3,27
10	2,88	3,02	3,12	3,11	3,11	3,1	3,09	2,98	2,95	2,89

30 test ses sinyalinin, ortalama MOS, PESQ ve Cov değerleri, Çizelge 6.9 ile Çizelge 6.15 arasında verilmektedir. Bu tablolarda A1, A2 ve A3 sırasıyla birinci, ikinci ve üçüncü arama yaklaşımlarını, ER (+) ise hata azaltma işleminin uygulandığını göstermektedir.

Çizelge 6.9. 2.4 Kbps için ortalama MOS, PESQ ve Cov değerleri

Yöntemler	ER	MOS	PESQ	Cov
ICA_VBR	ER(+)	2,50	2,38	2,94
ICA_CBR	ER(+)	2,47	2,32	2,85
KLT_VBR (A1)	ER(+)	2,52	2,55	3,02
KLT_CBR (A1)	ER(-)	2,47	2,48	2,98
KLT_VBR (A3)	ER(+)	2,42	2,43	2,81
KLT_CBR (A3)	ER(+)	2,33	2,29	2,58
LPC	-	2,31	2,35	2,61

Farklı kod kitabı büyüklükleri için deney sonuçları, Çizelge 6.10'da gösterilmektedir. Yüksek bit oranlarındaki büyük boyutlu kod kitapları için hesaplama gecikmesinin arttığı gözlemlenmiştir. Bu nedenle, alt bölüm 5.3.2'de anlatıldığı gibi, ikinci yaklaşım (A2) için kod kitabına vektör nicemleme uygulanarak, özvektör kod kitabının boyutu (ya da özvektör matrislerinin sayısı) 65536'dan 16384'e ve 1024'e düşürülmüştür. Çizelge 6.10'daki K , kullanılan katsayıların uzunluğunu veya en iyi sonucu veren en büyük özdeğer sayısını belirtir.

Çizelge 6.10. Nicemleme yöntemi ile 16 Kbps ve 19.85 Kbps için bulunan ortalama PESQ ve Cov değerleri

Arama yaklaşımları	Kod kitabı boyutu	Kbps	KLT_CBR			KLT_VBR		
			K	PESQ	Cov	K	PESQ	Cov
A2	1024	16	5	2,86	3,54	6	3,01	3,70
A2	16384	16	3	3,41	4,01	6	3,54	4,12
A1	65536	16	7	3,82	4,49	7	3,97	4,62
A1	65536	19.85	7	4,01	4,68	7	4,11	4,74
A3	1024+64	19.85	5	3,85	4,53	5	4,01	4,67
A3	1024+64	16	7	3,65	4,30	5	3,81	4,47

Çizelge 6.10'da, üçüncü yaklaşım (A3) için, normalize edilmiş özvektör matrislerinden oluşan kümeler kullanılmıştır ve sayısı 1024'tür ($M = 1024$). Aynı zamanda her bir kümedeki özvektör matrislerin sayısı 64'tür ($L=64$). En uygun özvektör matrisini bulmak için ise toplam $M + L$ ($1024 + 64$) özvektör matrisi araştırılmıştır. Çizelge

6.10'dan görüleceği gibi, ikinci yaklaşım (A2) için nicelme yöntemi kullanıldığında kalite değerleri düşmektedir. Ancak üçüncü yaklaşımda kalitenin, 16kbps ve 19.85kbps için tam arama yapan birinci yaklaşıma (A1) yakın olduğu görülmektedir.

Çizelge 6.11 ile Çizelge 6.15 arasında, iyi bilinen G729A, G728, EVS, AMR-NB ve AMR-WB ses kodlayıcıları, önerilen KLT ve ICA tabanlı yöntemlerle karşılaştırılmıştır. G729A, AMR-NB, G728, EVS (Narrow-Band) ve AMR-WB için standart ses kodlayıcıları sırasıyla 8 kbps, 12.2 kbps, 16 kbps, 16.4 kbps ve 19.85 kbps bit oranlarına sahiptir. Çizelge 6.11 ile Çizelge 6.14 arasında, ICA tabanlı yöntem için bulunan en iyi sonuçlar 4×4 boyutlu karışım matrisleri ve $4 \times N$ boyutlu bağımsız bileşen matrisleri kullanılarak elde edilmiştir. Burada N çerçevelerin uzunluğuna karşılık gelir. Ek olarak, Çizelge 6.9'da ve Çizelge 6.15'deki ICA tabanlı yöntem için 2×2 boyutlu karışım matrisleri ve $2 \times N$ boyutlu bağımsız bileşen matrisleri kullanılmıştır.

Çizelge 6.11. 8 Kbps için ortalama MOS, PESQ ve Cov değerleri

Yöntemler	ER	MOS	PESQ	Cov
ICA_VBR	ER(+)	3,38	2.98	3.77
ICA_CBR	ER(-)	3,28	2.83	3.56
KLT_VBR (A1)	ER(+)	3,61	3.50	4.10
KLT_CBR (A1)	ER(-)	3,42	3,20	3,82
KLT_VBR (A3)	ER(+)	3,40	3.25	3.85
KLT_CBR (A3)	ER(-)	3,30	3,03	3,68
G729A	-	3,58	3,38	3,91

Çizelge 6.12. 12.2 Kbps için ortalama MOS, PESQ ve Cov değerleri

Yöntemler	ER	MOS	PESQ	Cov
ICA_VBR	ER (+)	3,94	3,48	4,11
ICA_CBR	ER (+)	3,80	3,36	3,95
KLT_VBR(A1)	ER (+)	4,05	3,78	4,38
KLT_CBR (A1)	ER (+)	4,00	3,65	4,22
KLT_VBR (A3)	ER (+)	3,95	3,64	4,23
KLT_CBR (A3)	ER (+)	3,82	3,38	4,04
AMR-NB	-	4,11	3,74	4,32

Çizelge 6.13. 16 Kbps için ortalama MOS, PESQ ve Cov değerleri

Yöntemler	ER	MOS	PESQ	Cov
ICA_VBR	ER (+)	4,05	3,63	4,29
ICA_CBR	ER (-)	3,98	3,50	4,05
KLT_VBR (A1)	ER (+)	4,15	3,97	4,62
KLT_CBR (A1)	ER (+)	4,08	3,82	4,49
KLT_VBR (A3)	ER (+)	4,06	3,81	4,47
KLT_CBR (A3)	ER (+)	4,00	3,65	4,30
G728	-	4,11	3,68	4,43

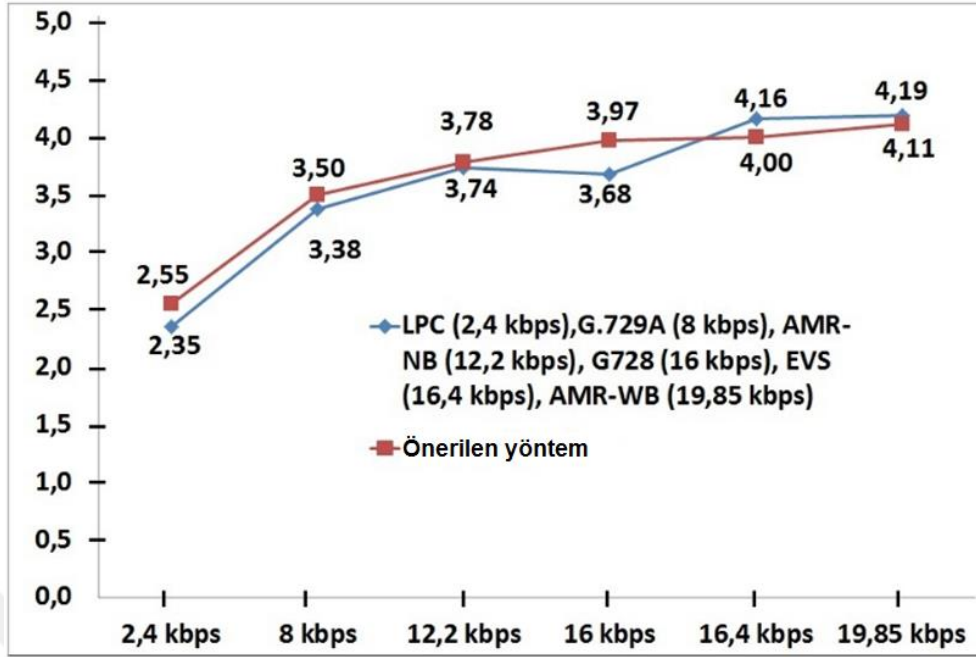
Çizelge 6.14. 16.4 Kbps için ortalama MOS, PESQ ve Cov değerleri

Yöntemler	ER	MOS	PESQ	Cov
ICA_VBR	ER (+)	4,13	3,67	4,32
ICA_CBR	ER (+)	4,03	3,56	4,11
KLT_VBR (A1)	ER (+)	4,19	4,00	4,65
KLT_CBR (A1)	ER (+)	4,10	3,84	4,51
KLT_VBR (A3)	ER (+)	4,08	3,83	4,48
KLT_CBR (A3)	ER (+)	4,04	3,68	4,32
EVS (NB)	-	4,43	4,16	4,80

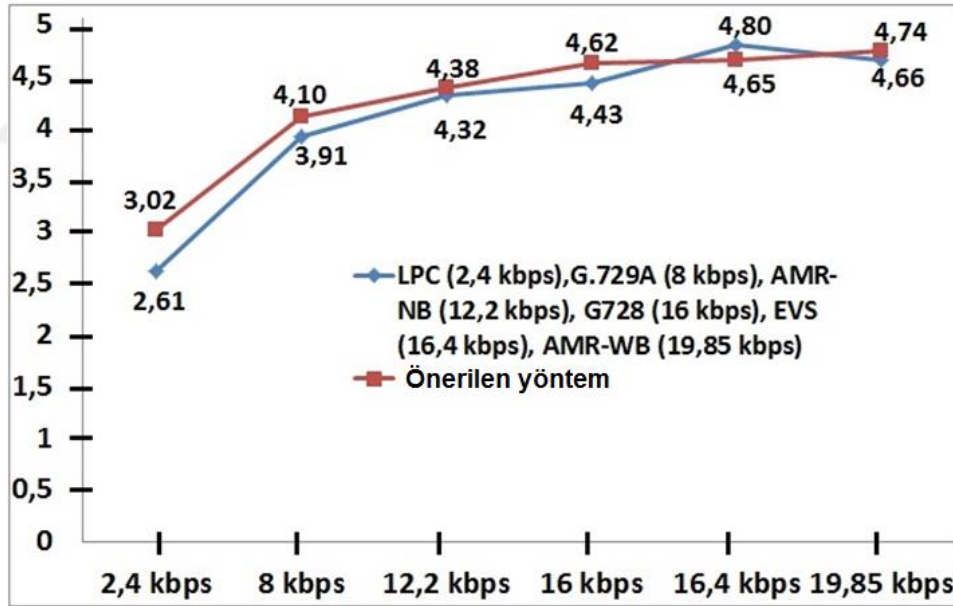
Çizelge 6.15. 19.85 Kbps için ortalama MOS, PESQ ve Cov değerleri

Yöntemler	ER	MOS	PESQ	Cov
ICA_VBR	ER (+)	4,23	3,84	4,60
ICA_CBR	ER (+)	4,17	3,73	4,51
KLT_VBR (A1)	ER (+)	4,36	4,11	4,74
KLT_CBR (A1)	ER (+)	4,29	4,01	4,68
KLT_VBR (A3)	ER (+)	4,25	4,01	4,67
KLT_CBR (A3)	ER (+)	4,18	3,85	4,53
AMR-WB	-	4,50	4,19	4,66

Önerilen yöntem (KLT_VBR (A1)) ve diğer ses kodlayıcıları için elde edilen PESQ ve Cov değerleri, sırasıyla Şekil 6.4 ve Şekil 6.5'te karşılaştırmalı olarak gösterilmektedir.



Şekil 6.4. KLT_VBR (A1) ve diğer kodlayıcılar için elde edilen PESQ değerleri.



Şekil 6.5. KLT_VBR (A1) ve diğer kodlayıcılar için elde edilen Cov değerleri.

KLT ve ICA tabanlı yöntemler için çerçeve başına kodlayıcı ve kod çözücü gecikmeleri Çizelge 6.16 ve Çizelge 6.17’de sırası ile verilmiştir. Buradaki gecikmeler çerçeve başına düşen gecikmelerdir. Bu tablolarda FL, mili saniye cinsinden çerçeve uzunluğudur. ED ve DD sırası ile, kodlayıcı ve kod çözücü gecikmeleridir ve mili saniye cinsinden gösterilir. K_C, K_V sırası ile KLT_CBR ve KLT_VBR’in kısaltması olarak gösterilmiştir.

Çizelge 6.16. KLT ve ICA tabanlı yöntemler için çerçeve başına kodlayıcı gecikmeleri.

	2,4 kbps		8 kbps		12,2 kbps		16 kbps		16,4 kbps		19,85 kbps	
	ED	FL	ED	FL	ED	FL	ED	FL	ED	FL	ED	FL
K_C (A1)	382	10	1260	4	1520	6	2190	6	2210	6	2420	6
K_C (A3)	14	10	29	4	43	6	59	6	63	6	74	6
K_V (A1)	302	10	1182	6	1440	6	2082	6	2098	6	2280	6
K_V (A3)	12	10	26	6	37	6	54	6	59	6	71	6
ICA_CBR	266	20	965	4	1380	4	1918	2	1981	2	2195	6
ICA_VBR	204	20	742	4	1208	4	1650	2	1705	2	2026	6

Çizelge 6.17. KLT ve ICA tabanlı yöntemler için çerçeve başına kod çözücü gecikmeleri.

	2,4 kbps		8 kbps		12,2 kbps		16 kbps		16,4 kbps		19,85	
	DD	F	DD	FL	DD	FL	DD	FL	DD	FL	DD	FL
K_C (A1)	0,6	10	0,8	4	0,9	6	0,98	6	1,01	6	1,1	6
K_C (A3)	0,18	10	0,32	4	0,45	6	0,50	6	0,51	6	0,57	6
K_V (A1)	0,57	10	0,78	6	0,87	6	0,96	6	0,98	6	1,07	6
K_V (A3)	0,12	10	0,28	6	0,4	6	0,48	6	0,50	6	0,55	6
ICA_CBR	0,51	20	0,67	4	0,83	4	0,92	2	0,94	2	1,01	6
ICA_VBR	0,46	20	0,63	4	0,79	4	0,91	2	0,92	2	0,98	6

7. SONUÇ VE ÖNERİLER

KLT'nin, dik dönüşümler arasında en yüksek enerji yoğunluğu sağladığı bilinmektedir. Sinyal iyi derecede korelasyona sahip ise, KLT ile sinyal enerjisinin çoğu bir kaç KLT dönüşüm katsayısına dönüştürülür ve geriye kalan küçük değerli dönüşüm katsayıları kodlanmaksızın atılabilir. KLT, en önemli sinyal bileşenlerini bulmak için sadece ikinci dereceden istatistikler kullanırken, ICA daha yüksek sıralı istatistikleri kullanmaktadır. Bu farklılıktan dolayı, bu dönüşümlerin her ikisi de CBR veya VBR içeren yeni bir ses kodlama yaklaşımında kullanılmıştır. Deneysel çalışmalarda, iletişim sistemleri için ITU-T'nin test sinyalleri olarak üretilen dört farklı İngilizce ses sinyali, önerilen yöntemlerle test edilmiştir. Bu testlerin ortalama PESQ ve Cov değerleri, özellikle TIMIT veri kümesini kullanan 4 ms ve 6 ms'lik çerçeve uzunlukları için bulunan PESQ ve Cov değerlerine yakındır. Ayrıca sonuçlar, en yeni standart kodlayıcılara karşı iyi bir performans sergilemekte, bu da önerilen yöntemlerin ses kodlamada alternatif bir yöntem olarak düşünülmesi gerektiğini göstermektedir.

Algoritmalar, Intel i-5 işlemciye ve 4GB RAM'e sahip masa üstü bilgisayarda "Matlab" ortamında çalıştırılmıştır. Bu çalışmada, hesaplama gecikmesinin istenilen bit hızına ve kod kitabının boyutuna göre değiştiği gözlenmiştir. İlk yaklaşım en iyi kaliteyi verir, ancak hesaplama gecikmesi çok yüksektir. Üçüncü yaklaşım, 16 kbps, 16.4 kbps ve 19.85 kbps için ilk yaklaşıma yakın bir kaliteye sahiptir. Ayrıca üçüncü yaklaşımın hesaplama gecikmesi ilk yaklaşıma göre oldukça düşüktür. İkinci yaklaşım (A2) ile sentezlenen ses sinyallerinin kalitesinin, birinci ve üçüncü (A3) yaklaşımlara göre daha kötü olduğu görülmüştür, yani, daha kötü PESQ ve Cov değerleri vermektedir.

KLT_VBR (A1) yöntemini kullanarak sentezlenen ses sinyallerinin kalite değerleri, LPC, G729A, G728, AMR-NB (12.2 kbps) ve AMR-WB (19.85 kbps) kodlama yöntemlerinin kalite değerlerinden daha yüksektir. KLT_VBR (A1) ve KLT_VBR (A3) yöntemlerinin PESQ ve Cov değerleri, EVS'den (16.4 kbps) daha düşük çıkmaktadır. KLT_VBR (A1) yönteminin PESQ değeri yalnızca AMR-WB (19.85 kbps) değerinden düşüktür. Bununla birlikte, KLT_VBR (A1) 'nin Cov değeri AMR-WB'den daha yüksek

bulunmuştur. KLT_CBR (A1) yöntemi, G728 ve LPC yöntemlerinden daha yüksek PESQ ve Cov değerleri vermiştir. Buna ek olarak, KLT_CBR (A1) yöntemi, AMR-WB'den daha yüksek bir Cov değeri vermiştir. ICA_VBR yöntemi, LPC yönteminden daha yüksek PESQ ve Cov değerleri vermektedir. Buna ek olarak, ICA_CBR yöntemi, LPC yönteminden daha yüksek bir Cov değeri vermiştir. ICA_VBR ve ICA_CBR yöntemlerinin PESQ ve Cov değerleri G729A, AMR-NB ve AMR-WB yöntemlerinden daha düşüktür. Bununla birlikte, KLT_VBR (A3), KLT_VBR'den (A1) biraz daha düşük PESQ ve Cov değerlerine sahiptir. Ayrıca, KLT_VBR (A3), LPC, G728'den daha yüksek PESQ ve Cov değerlerine ve AMR-WB'den daha yüksek Cov değerine sahiptir. KLT_VBR (A3) 'un PESQ ve Cov değeri, AMR-NB ve G729A'dan biraz düşük bulunmuştur.

Bu çalışmanın bir katkısı da, KLT ile elde edilen özvektör matrislerinin temel bileşen vektörlerine, vektör nicemeleme uygulamaktır. Ayrıca, ICA kullanılarak ses kodlama yapılması literatürde ilk kez uygulanmaktadır. Bir diğer katkı ise, tekrar eden bir hata azaltma işlemini gerçekleştirmektir. Bu işlem için hata sinyali kullanılarak, istenen bit oranı elde edilinceye kadar SNR değeri tekrar tekrar iyileştirilir. Bu çalışmada, CBR ve VBR kullanan ses sıkıştırması için, iki popüler altuzay yöntemi olan KLT ve ICA kullanılması önerilmiştir. Kullanılan bütün kombinasyonlarda (KLT_CBR, KLT_VBR), yeni bir kovaryans özvektör kümeleme stratejisi gerçekleştirilmiştir. Bu iki alt uzay yönteminin, hem CBR hem de VBR için çeşitli hedef bit oranlarında karşılaştırılmasının, bu alt uzay yöntemlerin kullanılabilirliği için kapsamlı bir gerekçe oluşturacağı düşünülmüştür. Ayrıca hata sinyalinin kodlayıcıya tekrar giriş sinyali olarak geri beslenmesi de, hedef bit oranında SNR'yi iyileştirmek için geliştirilmiş yeni bir yöntem olarak önerilmiştir. Dolayısı ile birinci bölümde (Giriş ve Amaç) bahsedildiği gibi, savunulan temel düşüncelerin uygulanması ile yapılan ses kodlamanın başarılı sonuçlar verdiği görülmüştür. KLT tabanlı yöntemler, özellikle KLT_VBR, deneysel olarak ICA tabanlı ve diğer bahsedilen yöntemlerden daha iyi sonuçlar vermiştir. KLT_VBR'ın kullanılması, benzer bit oranlardaki yeni ses kodlama standartlarına kıyasla kalite ve gecikme süresi anlamında makul bir performans sağlamıştır. Hesaplama gecikmesi, yüksek bit oranlarındaki kod kitabı boyutuyla ilişkili olarak artmaktadır. Yüksek bit oranlarındaki KLT temelli kodlamada, hesaplama gecikmesi sorununu gidermek için, nicemeleme uygulanmış ve hesaplama gecikmeleri azaltılmıştır.

Sentezlenen sinyaller için 12.2 kbps, 16 kbps, 16.4 kbps ve 19.85 kbps hızlarında yüksek kalite elde edilmiştir. Bununla birlikte, hesaplama gecikmelerinin birinci yaklaşımla bu bit oranlarında yüksek hesaplama gecikmesine sahip olduğu gözlemlenmiştir. Resim sıkıştırma için kullanılan KLT temelli yöntemler ile yapılan çalışmalarda, JPEG kodlayıcıdan biraz daha düşük kalitede resimler elde edilmiştir. Özellikle eğitim kümesindeki resimlerden elde edilen tek adetlik KLT özvektörü ile bulunan PSNR sonuçları, JPEG kodlama ile bulunan sonuçlara çok yakındır.

Deneysel çalışmalarla bulunan sonuçlar incelendiğinde, kodlayıcı tarafta ses sinyallerinin hesaplama gecikmesi, farklı arama yöntemleri geliştirerek daha da azaltılabilir. Ayrıca kod çözücü kısımda sentezlenen ses filtreden geçirilerek daha iyi bir ses kalitesi de elde edilebilir.

KAYNAKLAR DİZİNİ

- Ahmed, N., Natarajan, T., Rao, K. R., 1974. Discrete Cosine Transform. *IEEE transactions on Computers*. 100(1), 90-93.
- Atti, V., Sinder, D. J., Subasingha, S., Rajendran, V., Dewasurendra, D., Chebiyyam, V., Zhang, X., 2015. Improved error resilience for VOLTE and VOIP with 3GPP EVS channel aware coding. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE. pp. 5713-5717.
- Bruhn S., “Standardization of the new 3GPP EVS Codec”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2015, Brisbane, Australia, April 2015*.
- Cattermole, K. W. (1969). *Principles of pulse code modulation*.
- Chen, J.H., Cox, R.V., Lin, Y.C., Jayant, N., Melchner, M.J., 1992. A Low- Delay CELP Coder for the CCITT 16 kb/s Speech Coding Standard. *IEEE Journal on Selected Areas in Communications*. 10(5), 830–849. doi:10.1109/49.138988
- Comon P. 1994 Independent component analysis: a new concept? *Signal Process*.36, 287–314. (doi:10.1016/0165-1684(94)90029-9)
- Deng,L., O'Shaughnessy, D., 2003. *Speech processing, in: a dynamic and optimization-oriented approach*. CRC Press.,New York, pp. 41–50.
- Devalapalli, S. K., Rangarajan, R., Venkataramanan, R., 2003. Design of a CELP Speech Coder and Study of Complexity vs Quality Trade-offs for Different Codebooks. *EECS 651-Source Coding Theory*.
- Dunn, R. B., Quatieri, T. F., Reynolds, D. A., & Campbell, J. P. (2001, November). Speaker recognition from coded speech and the effects of score normalization. In *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on* (Vol. 2, pp. 1562-1567).
- Ferreira, A. J., Figueiredo, M. A.,2003. Class-adapted image compression using independent component analysis. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*. IEEE.1, pp. I-625.
- Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., & Pallett, D. S. (1993). *DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM*. NIST speech disc 1-1.1. NASA STI/Recon technical report n, 93.

KAYNAKLAR DİZİNİ (devam)

- Gerson, I.A., Jasiuk, M.A., 1990. Vector sum excited linear prediction (VSELP) speech coding at 8 kbps. *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on.* 1, 461–464. doi:10.1109/ICASSP.1990.115749
- Gibson, J.D., 2005. *Speech Coding Methods, Standards, and Applications.* IEEE Circuits and Systems Magazine. 5(4), 30-49.
- Goudarzi, M., Sun, L., 2009. Performance analysis and comparison of PESQ and 3SQM in live 3G mobile networks. *Advances in Communications, Computing, Networks and Security: Proceedings of the MSc/MResprogrammes from the School of Computing, Communications and Electronics, 2007-2008,* 6, 48.
- Goyal, V. K., 2001. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine.* 18(5), 9-21.
- Holma, H., Toskala, A., 2011. *LTE for UMTS: Evolution to LTE-advanced,* 2th ed. New York, John Wiley & Sons, pp.351-352.
- Hu, Y., Loizou, P. C., 2008. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on audio, speech, and language processing.* 16(1), 229-238.
- Hyvärinen, A., 2013. Independent component analysis of images. *Encyclopedia of Computational Neuroscience,* 1-5.
- Hyvärinen, A., Oja, E., 2000. Independent component analysis: algorithms and applications. *Neural networks.* 13(4), 411-430.
- Jain, A. K., 2010. Data clustering: 50 years beyond K-means. *Pattern recognition letters.* 31(8), 651-666.
- Jayant, Nuggehally S. "Digital coding of speech waveforms: PCM, DPCM, and DM quantizers." *Proceedings of the IEEE* 62.5 (1974): 611-632.
- Ju, H., Lee, S., Kim, M. Y., 2014. Complexity reduction in Karhunen-Loeve transform based speech coder for voice transmission. *IEEE Transactions on Consumer Electronics.* 60(1), 130-136.
- Jutten C, Héroult J. 1991 Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture. *Signal Process.* 24, 1–10. (doi:10.1016/0165-1684(91)90079-X)
- Kelly, John L., and Carol C. Lochbaum. "Speech synthesis." (1962): 1-4.

KAYNAKLAR DİZİNİ (devam)

- Kim, M. Y., Kleijn, W. B., 2004. KLT-based adaptive classified VQ of the speech signal. *IEEE transactions on speech and audio processing*. 12(3), 277-289.
- Kleijn, W. Bastiaan, and Kuldip K. Paliwal. *Speech coding and synthesis*. Elsevier Science Inc., 1995.
- Kondoz, A.M., 2007. *Digital speech, in: coding for low bit rate communication systems*. John Wiley & Sons.
- Krishnamoorthy, P., 2011. An overview of subjective and objective quality measures for noisy speech enhancement algorithms. *IETE technical review*. 28(4), 292-301.
- Kumar, P., Kumar, P., 2012. Performance evaluation of DFT-spread OFDM and DCT-spread OFDM for underwater acoustic communication. In *Vehicular Technology Conference (VTC Fall)*. IEEE. pp. 1-5.
- Kumar, S., Bhattacharya, S., Patel, P., 2014. A new pitch detection scheme based on ACF and AMDF. In *Advanced Communication Control and Computing Technologies (ICACCCT), 2014 International Conference on*, IEEE. 1235-1240. doi: 10.1109/ICACCCT.2014.7019296
- Lecomte, J., Vaillancourt, T., Bruhn, S., Sung, H., Peng, K., Kikuri, K., ... Faure, J., 2015. Packet-loss concealment technology advances in EVS. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE. pp. 5708-5712.
- Lee, Y., Kim, M. Y., 2010. KLT-based adaptive entropy-constrained quantization with universal arithmetic coding. *IEEE Transactions on Consumer Electronics*. 56(4), 2601-2605.
- Ney, H., & Ortmanns, S. (2000). Progress in dynamic programming search for LVCSR. *Proceedings of the IEEE*, 88(8), 1224-1240.
- Oger, M., Ragot, S., Antonini, M., 2006. Low-complexity wideband LSF quantization by predictive KLT coding and generalized Gaussian modeling. In *Signal Processing Conference, 2006 14th European*. IEEE. pp. 1-5.
- Ozerov, A., Kleijn, W. B., (2011). Asymptotically optimal model estimation for quantization. *IEEE Transactions on Communications*. 59(4), 1031-1042.
- Osahenvemwen, O. A., 2015. *Subjective Speech Evaluation on Mobile Communication Networks*.
- Proakis, John G. *Companders*. John Wiley & Sons, Inc., 2001.

KAYNAKLAR DİZİNİ (devam)

- Rabiner, L. R., & Schafer, R. W. (2007). Introduction to digital speech processing. Foundations and Trends® in Signal Processing, 1(1–2), 1-194.
- Rashed, M. A., El-Garf, T. A., Tarrad, I. F., Almotaafy, H. A., 2013. The effect of weight factor on the performance of G. 729A speech coder.
- Schroeder, M., and B. S. Atal. "Code-excited linear prediction (CELP): High-quality speech at very low bit rates." Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85.. Vol. 10. IEEE, 1985.
- Selen, N. 1979. Söyleyiş Sesbilimi Akustik Sesbilimi ve Türkiye Türkçesi. Ankara: Ankara Üniversitesi Basımevi. 1-8.
- Skodras, A., Christopoulos, C., Ebrahimi, T., 2001. The JPEG 2000 still image compression standard. IEEE Signal processing magazine. 18(5), 36-58.
- Supplee, L. M., Cohn, R. P., Collura, J. S., McCree, A. V., 1997. MELP: the new federal standard at 2400 bps. In Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on, IEEE. 2, 1591-1594.
- Switzer P., Green A., Min/max autocorrelation factors for multivariate spatial imaging. Technical report no 6, Department of Statistics, Stanford University, Stanford, California, 1984.
- Vasuki, A., Vanathi, P.T., 2006. A review of vector quantization techniques. IEEE Potentials. 25(4), 39–47.
- Wang, J., Chang, C. I., 2006. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. IEEE transactions on geoscience and remote sensing, 44(6), 1586-1600.
- Zue, V., Seneff, S., & Glass, J. (1990). Speech database development at MIT: TIMIT and beyond. Speech Communication, 9(4), 351-356.

ÖZGEÇMİŞ

21.02.1980 tarihinde Malatya’da doğan Serkan Keser, lise eğitimini 1996 yılında Mersin’deki Tevfik Sırrı Gür Lisesinde tamamlamıştır. 2004 yılında, Mustafa Kemal Üniversitesi, Elektrik-Elektronik Mühendisliği bölümünden mezun olmuştur. Kırşehir Ahi Evran Üniversitesi, Kaman Meslek Yüksek Okulunda, 2008 yılından beri öğretim görevlisi olarak çalışmaktadır. Doktora programına 2009 yılında Eskişehir Osmangazi Üniversitesi, Elektronik Mühendisliği bölümünde başlamıştır.

