

KONUŐMACIYA BAĐIMLI AYRIK SÖZCÜK TANIMA

Tayfun ÖZİŐBAKAN

**YÜKSEK LİSANS TEZİ
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĐİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**MART 2011
ANKARA**

Tayfun ÖZİŞBAKAN tarafından hazırlanan KONUSMACIYA BAĞIMLI AYRIK SÖZCÜK TANIMA adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Prof. Dr. Müzeyyen SARITAŞ

Tez Danışmanı, Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Elektrik ve Elektronik Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof.Dr. İrfan KARAGÖZ

Elektrik ve Elektronik Mühendisliği Anabilim Dalı, G.Ü.

Prof. Dr. Müzeyyen SARITAŞ

Elektrik ve Elektronik Mühendisliği Anabilim Dalı, G.Ü.

Prof. Dr. Nihal Fatma GÜLER

Elektronik ve Bilgisayar Eğitimi Anabilim Dalı, G.Ü.

Tarih 04/03/2011

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Bilal TOKLU

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Tayfun ÖZİŞBAKAN

KONUŞMACIYA BAĞIMLI AYRIK SÖZCÜK TANIMA

(Yüksek Lisans Tezi)

Tayfun ÖZİŞBAKAN

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Mart 2011

ÖZET

Sayısal işaret işleme ve bilgisayar teknolojisindeki gelişmeler, ses işaretlerinin depolanmasını, sıkıştırılmasını, sentezlenmesini, tanınmasını ve tekrar oluşturulmasını mümkün kılmaktadır. Konuşmacıya bağımlı ayrik sözcük tanıma çalışmasında, ses örneklerine ait temel ses parametrelerinin çıkartılması için, Doğrusal Öngörülü Kodlama, DÖK, (Linear Predictive Coding, LPC) ve Mel-Frekansı Kepstral Katsayılar, MFKK, (Mel-Frequency Cepstrum Coefficients, MFCC) yöntemlerinden faydaylanıldı ve bu iki yöntem ile elde edilen sonuçlar karşılaştırıldı. Kullanıcı, alıştırma evresinde, sözcüklerin sisteme depolanmasını sağlar. Analog ses işaretleri, sayısala dönüştürülüp uygun biçimde depolandıktan sonra, temel ses parametreleri çıkartılır. Sözcük tanıma evresinde, sistem, kullanıcı tarafından söylenen kelimeleri tahmin etmeye çalışır. Örnek sözcük şabonlarına ait ses parametre vektörleri ile tanınmaya çalışılan sözcük ses parametre vektörleri arasındaki en kısa mesafe, Dinamik Zaman Çakıştırma, DZÇ, (Dynamic Time Warping, DTW) yöntemiyle araştırılır. Bu çalışma, bilgisayar destekli bir sistem olarak uyarlandı ve Matlab teknik programlama dili kullanıldı. Sistem performansını ölçebilmek için, ayrik sözcükler olarak Türkçe ve İngilizce'deki 0'dan 9'a kadar olan sayılar seçildi. İngilizce sayıları tanımada başarı oranı % 85-95 arasında gerçekleşirken, Türkçe sayıları tanımada başarı oranı %75-90 arasında gerçekleşti. MFCC yönteminin, LPC yöntemine oranla daha başarılı sonuçlar verdiği gözlemlendi. Ayrıca, ortam gürültüsünün, sözcük tanıma olumsuz

etkisini azaltmak için uygun teknikler araştırıldı. Gürültü süzme amaçlı olarak, Ayırık Dalgacık Dönüşümü, ADD, kullanıldı. ADD kullanıldığında, Türkçe sözcük tanımadaki başarının İngilizce sözcük tanımaya yaklaştığı görüldü.

Bilim Kodu : 905.1.067
Anahtar Kelimeler : Sözcük tanıma, Doğrusal Öngörülü Kodlama, Mel-Frekans Kepstrum Katsayıları, Dinamik Zaman Çakıştırma
Sayfa Adedi : 76
Tez Yöneticisi : Prof. Dr. Müzeyyen SARITAŞ

SPEAKER DEPENDENT ISOLATED WORD RECOGNITION**(M.Sc. Thesis)****Tayfun OZISBAKAN****GAZİ UNIVERSITY****INSTITUTE OF SCIENCE AND TECHNOLOGY****March 2011****ABSTRACT**

Recent developments in digital signal processing and computer technologies have made speech signals to be stored, to be compressed, to be synthesized and to be recognized digitally possible. In speaker dependent isolated word recognition implementation, two methods have been utilized in feature extraction of speech signals: Linear Predictive Coding, LPC, and Mel-Frequency Cepstrum Coefficients, MFCC. The results by LPC and MFCC are compared. After the speech signals are converted from analog to digital and stored in appropriate format, the basic parameters of speech features are extracted. The minimum distance between feature parameter vectors of the template words and the feature parameter vectors of word to be recognized is calculated by Dynamic Time Warping, DTW, method. Speaker dependent isolated word recognition is implemented on computer based system. In the training phase, a user records the words in computer. In word recognition phase, system tries to recognize which word is uttered by the user. Matlab technical computation programming language is used for this study. In order to measure the performance of system, isolated words are chosen the numbers between 0 and 9 from Turkish and English language. English numbers are recognized by 85-95 % and Turkish numbers are recognized by 75-90 %. In general, MFCC method has much better performance than LPC method to recognize isolated words. An appropriate method has been investigated in order to eliminate the negative effect of background noise on the word

recognition. In denoising the speech signal, Discrete Wavelet Transformation, DWT, is utilized. It is observed that with the combined use of DWT in recording, the performance of Turkish isolated word recognition is very near to the performance of English isolated word recognition.

Science Code : 905.1.067

Key Words : Word recognition, Linear Predictive Coding, Mel-Frequency Cepstrum Coefficients

Page Number: 76

Adviser : Prof. Dr. Muzeyyen SARITAS

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren, sabrını hiçbir zaman esirgemiyen Hocam Prof. Dr. Muzeyyen SARITAŞ'a teşekkürü bir borç bilirim. Yüksek lisans çalışmamın sona ulaşmasında, yakın ilgi ve katkılarından dolayı Emine TÜTÜNCÜ'e de ayrıca teşekkür ederim.

İÇİNDEKİLER

	Sayfa
ÖZET.....	iv
ABSTRACT.....	vi
TEŞEKKÜR.....	viii
İÇİNDEKİLER	ix
ÇİZELGELERİN LİSTESİ.....	xi
ŞEKİLLERİN LİSTESİ	xii
RESİMLERİN LİSTESİ.....	xiv
KISALTMALAR	xv
1. GİRİŞ	1
2. SES SİNYALİ ve SAYISAL SES İŞLEME	3
2.1. Ses İşareti ve İnsan Ses Üretme Sistemi	3
2.2. Ses İşaretinin İfade Edilmesi.....	6
2.3. Sayısal Ses İşlemi Sistem Birimleri.....	8
3. KONUŞMA TANIMA İŞLEMİ	15
3.1. Konuşma Tanıma Sistemleri.....	16
3.2. Konuşma Tanıma İşleminde Temel Birimler.....	17
3.3. Mel-Frekans Kepstrum Katsayıları, MFKK	28
3.4. Doğrusal Öngörülü Kodlama, DÖK	33
3.5. Dinamik Zaman Çakıştırma, DZÇ	37
3.6. Ortam Gürültüsünün Azaltılması	42
4. KONUŞMACIYA BAĞIMLI AYRIK SÖZCÜK TANIMA SİSTEMİNİN TASARIMI ve GERÇEKLEŞTİRİLMESİ.....	45
5. SONUÇLAR ve İLERİYE YÖNELİK ÇALIŞMALAR.....	49

Sayfa

KAYNAKLAR	52
EKLER.....	54
EK-1 Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Örnekleri ve Sonuçları..	55
EK-2 Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Matlab program algoritması.. ..	57
EK-3 Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Matlab programı.....	58
ÖZGEÇMİŞ	76

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Bark ve Mel basamakları için, işitme spektrumu boyunca, merkez frekansına karşılık gelen kritik bant genişlikleri.....	31
Çizelge 3.2. Tipik gürültü kaynakları	43
Çizelge 5.1. Konuşmacıya bağımlı, ortamdan bağımsız, Türkçe ve İngilizce rakamlar ile yapılan sözcük tanıma test sonuçları.....	51

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. İnsan ses mekanizmasının şematik gösterimi.	5
Şekil 2.2. Konuşma ses işareti için sayısal bir model	6
Şekil 2.3. Ses üretme sisteminin ifade edilmesi.....	7
Şekil 2.4. Sesin dalga-biçimi ve parametrik ifadesi	8
Şekil 2.5. Örnekleme frekansı	10
Şekil 2.6. Doğrusal kuantizasyon.....	11
Şekil 2.7. Doğrusal olmayan kuantizasyon	13
Şekil 2.8. Doğrusal olmayan kuantizasyon ile örnekleme	14
Şekil 3.1. Klasik konuşma tanıma sistemi.....	21
Şekil 3.2. Analog ses işaretinin, sayısallaştırılması	21
Şekil 3.3. 2sn. süre uzunluğunda “bir.wav” ses kaydı, uç-noktaların tespiti	23
Şekil 3.4. DP’yi açıklayacak ızgara yapısı	25
Şekil 3.5. (i, j) ’e düğümüne, bütün geçişlerin ileriye doğru olduğu bir yol yapısında optimal yol $(i - 1, j)$ ’dir.....	26
Şekil 3.6. Hava koşulları için Saklı Markov modeli	28
Şekil 3.7. Mel-Frequency Cepstrum Coefficients (MFCC) hesaplanmasının grafiksel gösterimi.....	30
Şekil 3.8. Triangular Filtre Bankası	32
Şekil 3.9. Mel basamakları filtre bankı	32
Şekil 3.10. İki özellik kümesinin çakışan noktalarından oluşan yol	38
Şekil 3.11. Monotonluk kısıtlaması.....	40

Şekil	Sayfa
Şekil 3.12. Itakura'nın global yol arama kısıtlaması.....	41
Şekil 3.13. Lokal kısıtlamalar	42
Şekil 3.14. Dalgacık Dönüşümünü	43
Şekil 3.15. Gürültü süzme modeli.....	44
Şekil 4.1. LPC için modelleme	46
Şekil 4.2. Gürültü süzme modeli.....	47
Şekil 4.3. LPC ve MFCC ile Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Sistemi Blok Diagramı	48

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 2.1. İnsan ses üretme mekanizmasının kesiti	4

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
ADD	Ayrık Dalgacık Dönüşümü
A/S	Analog/Sayısal
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DP	Dinamik Programlama
DÖK	Doğrusal Öngörülü Kodlama
DTW	Dynamic Time Warping
DZÇ	Dinamik Zaman Çakıştırma
DWT	Discrete Wavelet Transform
LPC	Linear Predictive Coding
MFCC	Mel-Frequency Cepstrum Coefficients
MFKK	Mel-Frekans Kepstrum Katsayıları
ODF	Olasılık Dağılım Fonksiyonu (Probability Density Function)

1.GİRİŞ

Konuşma, insanlar için en uygun ve en etkili iletişim yoludur. Konuşma tanıma teknolojisi, konuşularak aktarılan bilgileri alabilen ve bu bilgiler üzerine hareket edebilen sistemleri geliştirmeyi ve üretmeyi amaçlar. Bir sonraki amaç, bilgi akışını sadece insandan makinalara değil, konuşma sentezi yoluyla makinalardan insana sağlamaktır. Konuşma tanıma çalışmaları, konuşulan mesajları alabilen, anlayabilen, metne dönüştüren, bu mesajlara göre harekete geçen ve konuşarak bilgi alışverişini tamamlayan yapay zekalı makinaların bir fonksiyonunun parçası olarak ele alınabilir.

Konuşma işaretinin, iletilmesi, sayısal olarak depolanması ve işlemi sırasında, içeriğinin kayıp olmaksızın korunması, ses işaretlerinin ifade edilmesini önemli kılmaktadır. Konuşma yoluyla aktarılan bilgi, özünde, ayrık yapıya sahiptir. Her dilde, bu en küçük yapı, fonem olarak adlandırılır. Konuşma tanıma sistemleri ile temel olarak ulaşılmak istenen amaç, fonem ses işaretlerinin, sayısal olarak dönüştürülmesi ve kodlanması ve tanınmasıdır. Her dilde yaklaşık, 30 ile 50 arasında fonem olduğu ve konuşma işaretinin ifade edilmesinde 500 ile 1 milyon bit/saniye aktarılması gerektiği düşünülecek olursa, konuşma tanımanın zor bir çalışma disiplini olduğu kolaylıkla anlaşılabılır.

Tüm zorluklara rağmen, konuşma tanıma alanında sağlanacak olan başarılar, insanlık açısından çok önemli olacaktır. Konuşma tanıma teknolojisindeki gelişmeler sayesinde ağır işitenler, daha iyi işitebilmekte, sağır olanlar konuşmalarını yazıya çevrilmesi ile konuşulanları anlayabilmektedir. Konuşma tanımayla ilgili bu tip örnekleri çoğaltmak mümkündür. İnsan sesinin, tıpkı parmak izi gibi ayırt edilebilir bir özelliğinin bulunması, kimlik belirleme ve güvenlik işlemlerinde alternatif bir başvuru noktası olmasını sağlamaktadır. Bu yöndeki çalışmalar ile, konuşma tanımanın yanısıra, konuşmacıyı tanıma ve belirleme disiplinleri önem kazanmaktadır.

Bu tez çalışmasında,

- Türkçe için konuşmacıya bağımlı, ayrık sözcük tanıma modeli

- Türkçe ve İngilizce ayırık sözcük tanıma performanslarının karşılaştırılması
- Ortam gürültüsünün azaltılması ile sözcük tanımanın iyileştirilmesi amaçlanmıştır.

2. bölümde, ses işareti ve sayısal ses işleme ele alınmıştır. Ses işaretinin, kısa zaman birimleri dahilinde durağan işaretlerden oluşan yapısı üzerinde durularak, insan ses üretme sistemi için, sesin sayısal ifade edilmesi üzerinde çalışılmıştır. Örnekleme, kuantazasyon, gibi sayısal ses işleme sisteminin temel birimleri üzerinde durulmuştur.

3. bölümde, konuşma tanıma sistemleri araştırılmıştır. Ayırık sözcük tanımadan daha karmaşık birleşik sözcük tanıma ve sürekli konuşma tanıma sistemleri üzerinde durulmuştur. Örnek sözcük şablonları ile tanınması hedeflenen sözcükler arasında, karşılaştırmayı gerçekleştirecek, dinamik programlama ve stokastik yaklaşımlar üzerinde durulmuştur. Ses örneklerinin temel paramterelerinin çıkartılmasında, Doğrusal Öngörülü Kodlama ve Mel-Frekans Kepstrum Katsayılarının elde edilmesi yöntemleri kullanılmış ve elde edilen sonuçları değerlendirebilmek için Dinamik Zaman Çakıştırma yöntemi ile kullanılmıştır.

4. bölümde, konuşmacıya bağımlı ayırık sözcük tanımanın, Matlab teknik programlama dili kullanılarak bilgisayar ortamında nasıl gerçekleştirildiği ele alınmıştır. Çalışmanın blok diagramı ve kullanıcının seçebileceği sistem parametreleri açıklanmıştır. Bu parametrelerin, sistem performansını nasıl etkilediği son bölümde ele alınmıştır.

Son bölümde, Türkçe ve İngilizce rakamlar ile gerçekleştirilen ayırık sözcük tanıma ve ekler bölümünde sunulan diğer Türkçe sözcük tanıma çalışmalarının sonuçları irdelenmiştir ve ileriye yönelik çalışmalar üzerinde durulmuştur.

2. SES SİNYALİ ve SAYISAL SES İŞLEME

2.1. Ses İşareti ve İnsan Ses Üretme Sistemi:

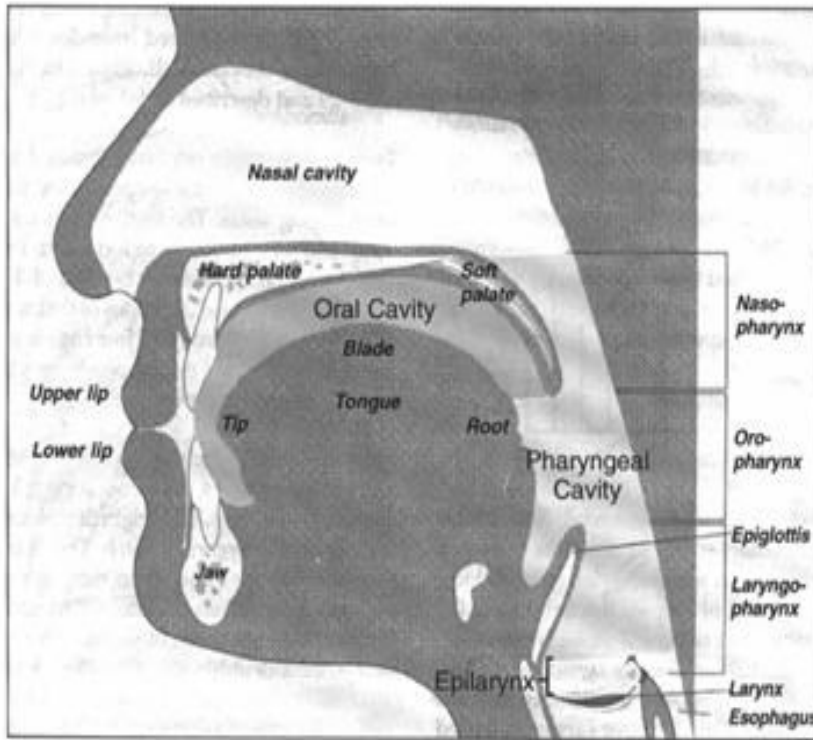
Ses İşareti:

Ses işareti şekilleri üzerine yapılan çalışmalar şunu göstermiştir ki konuşulan dil, insan kulağının ayırt edebileceği basit ses komponentlerine bölünerek analiz edilebilir. Bu basit ses komponentlerine ‘fonem’ (phonem) adı verilir. Kelimeler, cümleler, ünlemler fonemlerden oluşur. Amerikan İngilizcesi 40 fonemden oluşur. Normal konuşmada, her saniyede 10 ya da 15 fonem kullanılmaktadır. 40 fonem, 6-bit kodlama sistemi ile kodlanabileceğinden, konuşma işareti, saniyede minimum 60 ile 90 bit ile ses üretilebilir. Ses işaretinin içindeki mesaj bilgisi, insan veya otomatik makinalar tarafından kolaylıkla algılanabilmesi için, bu işaretin ifade ediliş tarzı önem taşımaktadır. Bu sebeble, ses işaretinin ifade edilmesinde 60 bits/s oranından 1 milyon bits/s oranına kadar hasasiyetler gerekmektedir. Tasarım ve uygulama safhalarında, işaret işleme yönteminin çok büyük rolü vardır.

İnsan ses üretme sistemi :

İnsanın ses üretme sistemini kavramak, elektronik ses sentezi üzerine yapılacak tartışmalar için yararlı olacaktır. İnsanın ses üretme sistemi, vokal yol (vocal tract) ve burun yolundan (nasal tract) oluşur. Vokal yol, ses tellerinden başlayıp dudaklarda son bulur. Değişebilen kesit alanlarına sahip vokal yol, akustik bir tübü andırır. Burun yolu ise, vokal yola velumun flape hareketi ile bağlanan bir başka boşluktur. İnsanlar, ses tellerinin, akciğerlerin, vokal ve nasal boşlukların birbirleriyle etkileşimlerinden dolayı farklı konuşma yetisine sahiptirler. Akciğerler, ses üretiminin güç kaynağıdır. Akciğerler, havanın ses telleri üzerinden vokal yol boyunca geçmesini sağlarlar. Vokal boşluk, dudaklar, dil, dişler ve çene kısımlarından oluşur. Resim 2.1 boğazdaki resonant odalarının, ağız ve burun boşluklarının, dilin pozisyonunun ve boğaz orifis büyüklüğünün kontrolü ile konuşmacı bir fonem üretir. Akciğerler ve ses telleri, impulsar (voiced sounds) ve

hissler (unvoiced sounds) üretme kabiliyetine sahiptir. Vokal yol, esnek bir organ-pipe olup ses tellerinden dudaklara kadar kesiti, isteğe bağlı olarak değişebilmektedir. Toplam kesit alanı, dudakların, dilin çenenin ve velumun pozisyonuna bağlı olarak sıfırdan 20 cm² kadar değişmektedir. Akustik bir dalga olan konuşma sesi, akciğerlerden salınan havanın, vokal yol tarafından sınırlandırılması ve şekillendirilmesi sonucu yayılır. İnsanın ses yolu iki farklı konuşma sesi oluşturur: periyodik sesli konuşmalar (periodic voiced sounds) ve aperiodyk sessiz konuşmalar (aperiodic unvoiced sounds). Sesli konuşmalar, sesli harflerden oluşur. a, e, ı, o ve u gibi sesli konuşmalar, havanın akciğerler tarafından vokal yol boyunca salınması, ses tellerinin kasılarak hava akışını module etmeleri sonucu yarı periyodik pulslar tarafından oluşturulur [3].

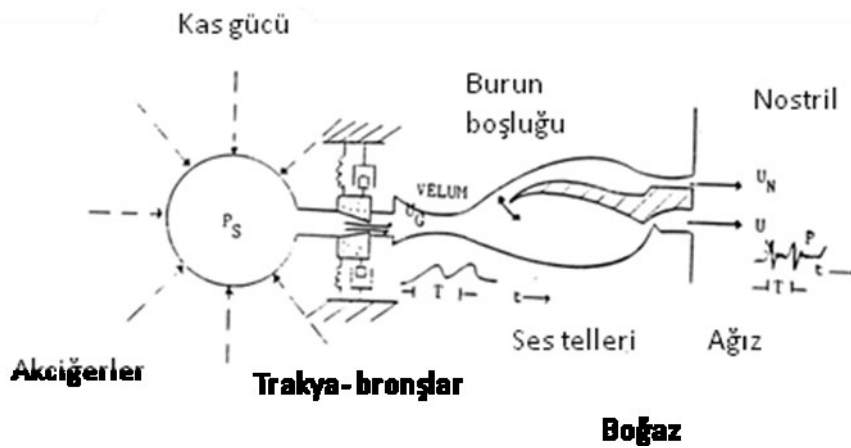


Resim 2.1. İnsan ses mekanizmasının kesiti

Sessiz konuşmalar ise iki gruba ayrılır: ‘frikatif’ (fricative) ve ‘plosif’ (plosive) sesler. S, z, f, h gibi frikatif sesler, vokal yolun uzunluğunun belirli bir noktasında

sınırlandırılması ve bu sınırlı bölgede türbulans oluşması sonucu yaratılır. Plosif sesler, p, t, k gibi, vokal yolun boğazın sonunda dil veya dudaklar sınırlandırılıp, sınırın gerisindeki basınçlı havanın bir anda serbest bırakılmasıyla oluşur. Genellikle plosif sesleri frikatif sesler takip eder.

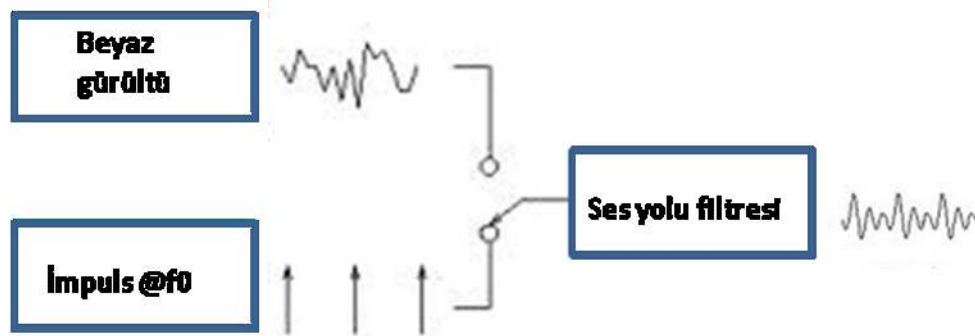
Vokal sistem, zamanla değişen bir filtre olarak algılanabilir. Ses frekansının spektrumu, 100 Hz. dolaylarından 3000 Hz. den daha yukarı frekanslara kadar uzanır. Vokal ve nasal yollar, kesiti düzgün olmayan akustik tüpler olarak algılanabilir (Şekil 2.1). Sesler, bu tüplerden yapılan salınımlarla üretilirler. Seslerin frekans spektrumu, tüplerin frekans seçiciliğine bağlıdır. Bu spektrum içinde bir takım frekans değerleri, vokal yolun akustik rezonans bölgelerine rastlar. Bu rezonans frekanslarına formant frekansları veya kısaca formant adı verilir. Sesin dalga boyu, vokal yolun uzunluğuyla oranlanacak bir parametre haline gelir. Vokal yol, bir ucu ağız tarafında açık diğer ucu ses telleri tarafında kapalı olan bir kornaya benzetilebilir. Formant frekansları, vokal yolun şekil ve uzunluğuna bağlıdır. Vokal yolun her şekli, yeni bir formant frekans setine karşılık gelir. Zamanla vokal yolun şekli değiştikçe konuşmanın da spektral özellikleri değişecektir.



Şekil 2.1. İnsan ses mekanizmasının şematik gösterimi

Örneğin, sesli konuşmalar için rezonant frekansları 500, 1500, 2500 Hz dir. Nasal yol bu durumda kapalıdır. Eğer nasal yol da açık ise 1000 Hz.lik rezonans frekansı daha

oluşur. Şekil 2.2.' de konuşma ses işareti için sayısal bir model görülmektedir. Giriş işaretinin uyartımı açısından vokal yol, bir modulator filtre olarak çalışır.



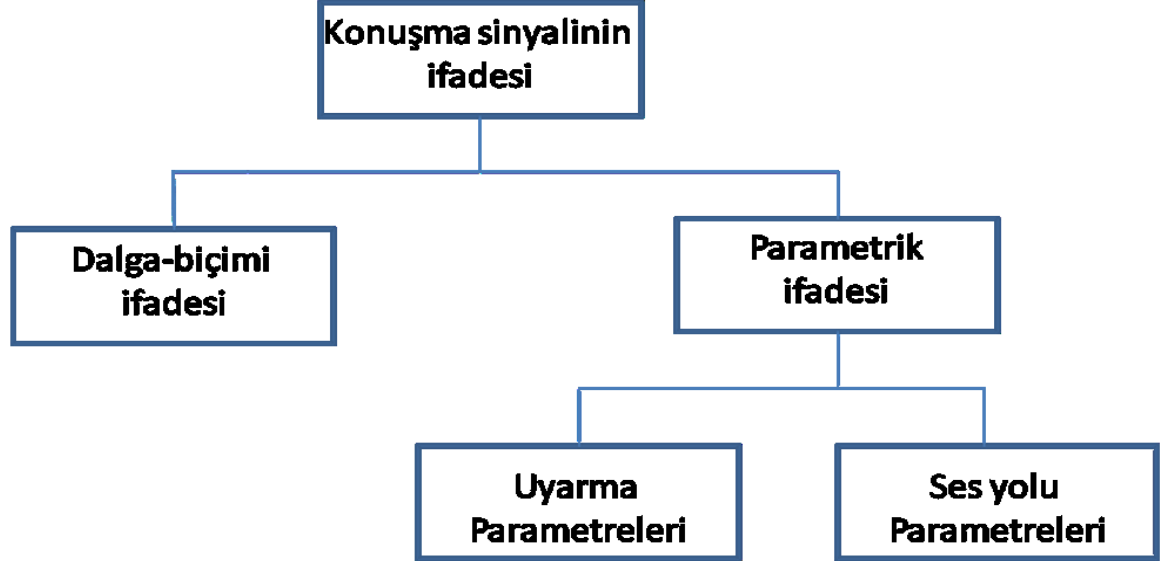
Şekil 2.2. Konuşma ses işareti için sayısal bir model

İşlem, gerçek konuşma sesinin oluşmasından farklı değildir. Akciğerlerin ses tellerini geçirerek gönderdiği hava kaynak olup, sistemin geri kalanında bu kaynak şekillenmektedir. Sayısal modelde, filtreye giriş olarak periyodik ya da gelişigüzel puls serileri uygulanmaktadır. Resonans frekansları ise fitrenin zamana göre karakteristiğinin değiştirilmesiyle olacaktır. Sayısal modeldeki filtrenin örnekleme çıktıları, vokal yolun iletim karakteristiğine yakınsamaktadır. Ses işaretinin parametrelerinin birkaç 10 mili saniye boyunca değişmeyeceğini kabul ederek, bu süre boyunca filtrenin karakteristiği sabit kalır. Bu aralıklar için filtrenin giriş-çıkış ilişkisi belirli katsayılarla ifade edilebilir. Bu katsayılara DÖK (LPC) katsayıları adı verilir [2]. Sesli konuşmalar için filtre periyodik impuls serisiyle sürülmekte, sessiz konuşmalar için ise gelişigüzel gürültü kaynağı aktif olmaktadır. Bu modelleme, ses işaretinin ifade edilmesinde ve sentez işlemlerinde geniş ölçüde kullanılmaktadır.

2.2. Ses İşaretinin İfade Edilmesi :

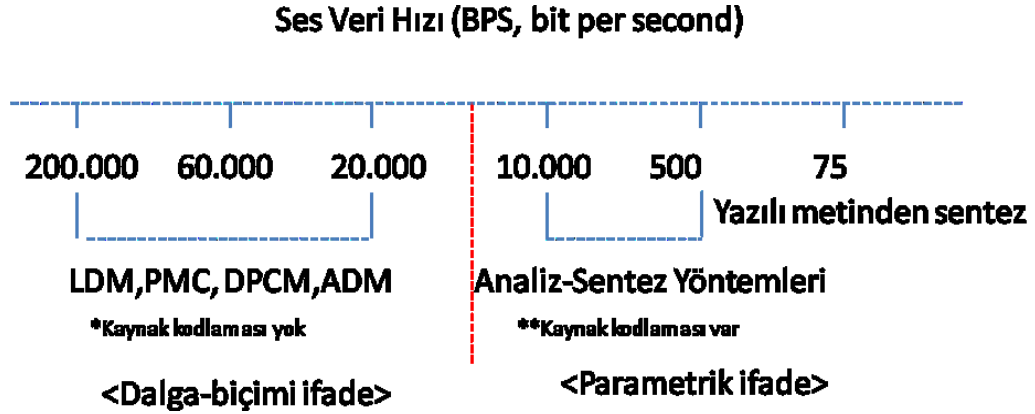
Ses işaretinin ifade edilmesi iki ana grupta incelenir: Dalga-biçimi ifadesi ve parametrik ifade. Şekil 2.3'de, dalga-biçimi ifadesi, isminden de anlaşılacağı gibi analog işaretin dalga şeklinin örnekleme ve kuantizasyon işlemleri boyunca

korunması ile ilgilenir. Parametrik ifade ise ses işaretini, bir ses üretme sisteminin çıktısı olarak kabul eder ve sistemin parametreleriyle ilgilenir.



Şekil 2.3. Ses üretme sisteminin ifade edilmesi

Ses işareti örneklenme ve kuantizasyon işlemlerinden sonra, ses üretme sisteminin parametrelerini çıkartmak için tekrar işlem yapılır. Modelin parametreleri uyartım parametreleri ve vokal yol parametreleri olmak üzere iki grupta ele alınır. Uyartım parametreleri ses işaretinin kaynağı ile ilgilenir. Vokal yol parametreleri ise kişisel konuşma sesleriyle ilgilenir. Şekil 2.4' de ses işaretinin çeşitli ifade biçimlerine karşılık, bu ifade biçimi için gerekli bir saniyedeki data oranları verilmiştir. 75 bits/s data oranıyla yazılı dokümandan sentez yapılabilirken, 200.00 bits/s data oranıyla dalga şekli tam olarak ifade edilebilmektedir. Tabii ki bir saniyedeki data oranları ses işaretinin ifade biçimini seçmek için tek kıstas değildir. Maliyetler, esneklik ve sesin kalitesi diğer göz önünde bulundurulacak faktörlerdir [11].



Şekil 2.4. Sesin dalga-biçimi ve parametrik ifadesi [11]

LDM, Doğrusal Delta Mod Modülasyonu (Linear Delta Modulation)

PCM, Darbe Kod Modülasyonu (Pulse Code Modulation)

DPCM, Ayrımsal Darbe Kod Modülasyonu (Differential Pulse Code Modulation)

ADM, Uyarlanır Delta Modülasyonu (Adaptive Delta Modulation)

2.3. Sayısal Ses İşlemi Sistem Birimleri :

Bant-geçirgen filtre:

Analog işaret öncelikle filtre edilip, ses dışındaki frekans komponentlerinden arındırılır. Sağlıklı örnekleme yapabilmek için, analog işaretin bant-sınırlı olması gerekmektedir. Bunları sağlamak için, yüksek kazançlı yüksek-geçirgen filtre ve yüksek kazançlı alçak-geçirgen filtreden oluşan bir bant-geçirgen filtre kullanılabilir. Bu filtre 300 Hz. ile 3400 Hz. 'ye sınırlandırılmış olmalıdır.

Örnekleme:

Ses işaretlerini sayısal olarak işleme tabii tutabilmek için analog formdaki işaret sayısalı dönüştürülmelidir. Sayısal dönüşüm, örnekleme, sürekli zamanın eşit aralıklara bölünerek noktasal ifade edilmesi ve kuantizasyon, örneklenen genliklerin

belirli seviyelere yerleştirilmesinden oluşmaktadır. Örneklenen dalga şekli şöyle ifade edilir.

$$s(n) = s_a(nT) \quad -\infty \leq n \leq \infty \quad (2.1)$$

s_a , analog dalga şekli ve T de iki ardışık örnekleme arasındaki zaman veya örnekleme periyodudur. Örnekleme teoremine göre, $s_a(t)$ işaretinin bant-sınırlı Fourier dönüşümü, aşağıdaki gibi hesaplanır [4].

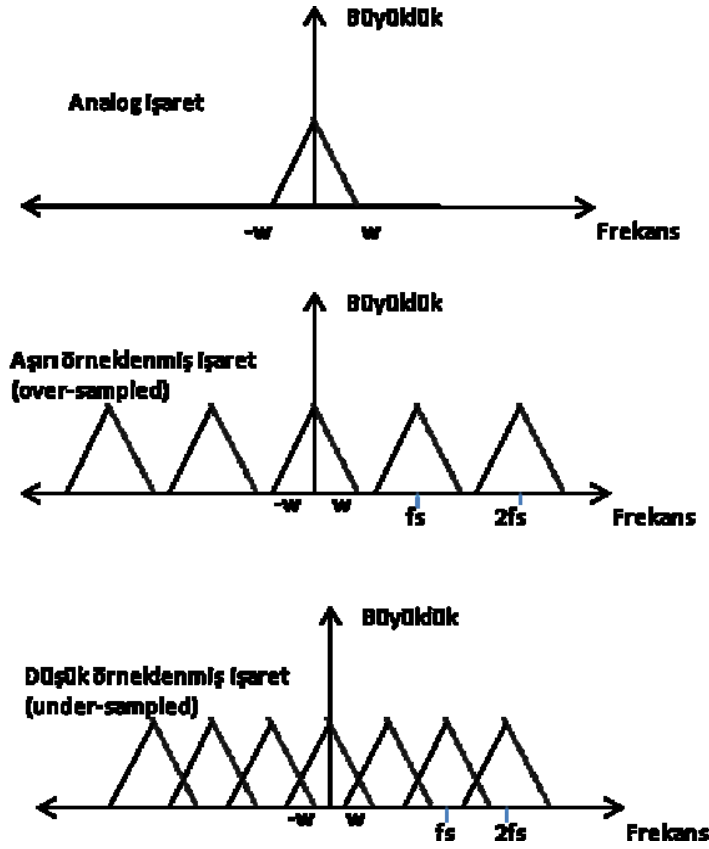
$$S_a(j\omega) = \int_{-\infty}^{\infty} s_a(t) e^{-j\omega t} dt \quad (2.2)$$

$$s_a(j\omega) = 0 \text{ için } \omega \geq 2\pi T \text{ ve } T \leq 1/2\omega$$

değerlerinde, analog işaret, örneklenen şeklin tekrarından oluşturulur. Bu denklemde, “ ω ”, Nyquist frekansı olarak bilinir. Analog işaretin bant-sınırlı Fourier dönüşümü, örnekleme frekansının tam katlarında aynı şekilde tekrarlanır. Çünkü, örneklenen işaretin Fourier dönüşümü, örnekleme frekansının tam katlarında hesaplanır. Bu aşağıdaki ilişkiyi önermektedir.

$$S(e^{j\omega t}) = \frac{1}{T} \sum_{n=-\infty}^{\infty} S_a(j\omega + j2\pi n / T) \quad (2.3)$$

Eğer ki örnekleme frekansı, Nyquist frekansının iki katından az ise ardışık örnekleme frekans spektrumları çakışacaktır. Yüksek frekans komponentlerinin düşük frekans komponentlerini bozması örtüşme (aliasing) olarak bilinir. Örtüşmeyi önlemek için giriş işareti örnekleme frekansının yarı değerine bant-sınırlı olmalı ya da örnekleme frekansı, analog işaretin en yüksek frekans komponentinin iki katı değerine eşit olmalıdır.



Şekil 2.5. Örnekleme frekansı

$1/T > 2W$ koşulu için örnekleme serinin Fourier dönüşümü, temel banttaki analog işaretin Fourier dönüşümü ile doğru orantılıdır.

$$S(e^{j\omega T}) = \frac{1}{T} S_a(j\omega) \quad |\omega| < \frac{\pi}{T} \quad (2.4)$$

Yukarıdaki denklemi kullanarak, orjinal analog işaret, örnekleme serisinden interpolasyon kullanılarak elde edilir [4].

$$s_a(t) = \sum_{n=-\infty}^{\infty} s_a(nT) \frac{\sin[\pi(t - nT) / T]}{\pi(t - nT) / T} \quad (2.5)$$

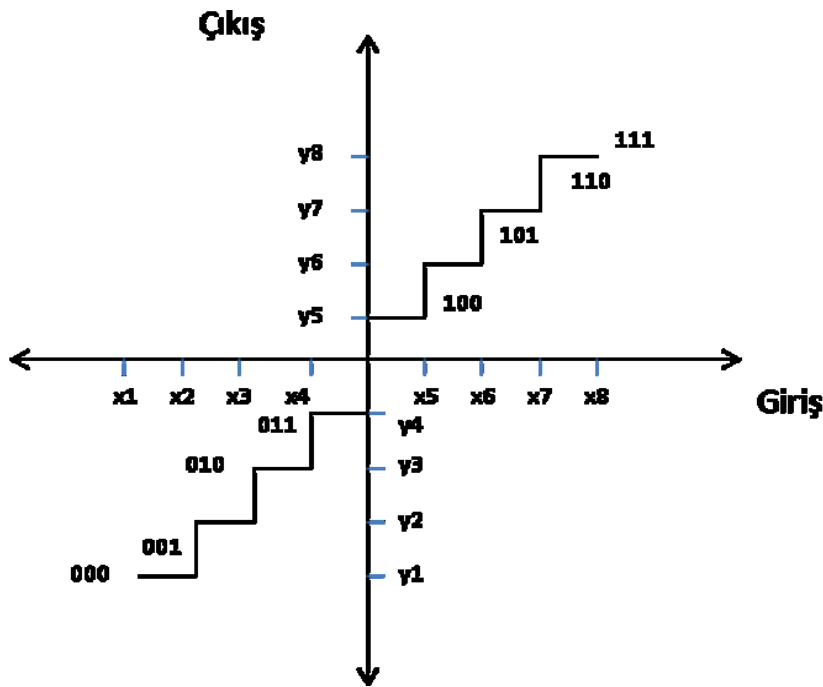
$$s_a(t) = \sum_{n=-\infty}^{\infty} s_a(nT) \text{sinc}(\phi) \quad \phi = \pi(t - nT) / T \quad (2.6)$$

Sonuç olarak, eğer örnekleme frekansı, Nyquist frekasının en az iki katı ise, analog işaret örnekleme serisinden, örnekleme noktalarına merkezli $\text{sinc}(\phi)$ fonksiyonlarının birbirine eklenip genliğinin örnekleme genliğiyle çarpılması sonucu

elde edilir. Sayısal modelde, $\text{sinc}(\phi)$ fonksiyonları ideal düşük-geçerirgen filtre olarak ifade edilirler. Pratikte, ön bant sınırlaması genellikle ideal olmayan düşük geçirgen filtrelerle yapılmaktadır ve bu da örtüşmeye sebep olabilmektedir. Örtüşme bozulmasını önlemek için, örnekleme frekansı Nyquist frekansının iki katından daha fazla seçilmektedir. Telekomünikasyon sistemlerinde, analog ses işareti 8000 Hz.'de örneklenmekte ve 300 Hz. ile 3400 Hz.'e bant-sınırlı durumundadır.

Kuantizasyon:

Kuantizasyon, sürekli genlikteki işareti, örneklenmiş genlikteki işarete çevirir. Örneklenmiş durumdaki işaret, sürekli durumdaki işareten kuantizasyon gürültüsü veya hatası kadar farklıdır. Her bir örneklenmiş işaret ayrı ayrı kuantizasyon işlemine giriyor ise buna skaler veya doğrusal kuantizasyon denilir (Şekil 2.6).



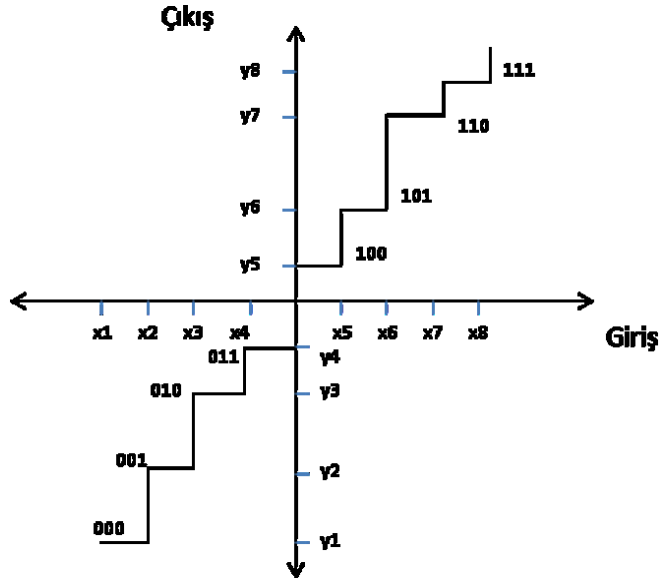
Şekil 2.6. Doğrusal kuantizasyon

Genlik seviyeleri arasındaki mesafeye, kuantizör adım büyüklüğü Δ denir. Her bir örnekleme genlik seviyesi Δ_n , bir kod-kelimesi (code-word) $C(n)$ ile ifade edilir. De-kuantizasyon kısmında kod-kelime $C(n)$, örneklenecek genlik seviyesini ifade

eder. Örneklenmiş genlik seviyelerinin hepsinin B kadar bit ile kodlandığını düşünecek olursak ve f_s , örnekleme frekansı için kanal bit iletim oranı aşağıda verilmiştir.

$$T_c = Bf_s \text{ bits/s} \quad (2.1)$$

Sabit örnekleme frekansı için, kanal bit iletim oranı T_c 'yi azaltmanın tek yolu kod-kelimesini kısaltmaktır. Ancak, kod-kelimesini azaltınca daha az sayıda genlik seviyeleri oluşacak ve bunların araları daha geniş Δ ile ayrılacaktır. Bu da örneklerinden oluşturulacak işaret için kalite kaybı anlamına gelir. Bu problemi çözmek için çeşitli kuantizör yapıları tasarlanmış ve pratikte kullanılmaktadır. Bu özel kuantizörün amacı giriş işaret karakteristiğinin kuantizöre, dinamik sınırlar ve ODF (olasılık dağılım fonksyonu) açısından en iyi uymasıdır. Bu tip bir kuantizörün genlik seviyelerinin seçilmesi önem taşımaktadır. Sabit kod-kelime uzunluğunda, işaret seviyesinin kuantizasyon gürültü seviyesine oranını maksimum yapmak için kuantizörün seviyelerinin analog işaretin ODF'e göre seçilmelidir. Ses tipindeki işaretler düzgün ODF'e sahip değildir. Düşük genlikli işaretlerin olasılığı, yüksek genlikli işaretlerin olasılığından daha fazladır. Sonuç olarak işaretin dinamik sınırlarını en iyi kapsayacak optimum kuantizör, doğrusal seviyeli olmayanlardır. Şekil 2.7.'de doğrusal olmayan optimum kuantizörün adım büyüklüğü, yüksek giriş seviyeleri için gittikçe artmaktadır. Kuantizasyon seviyelerinin doğrusal olmayan dağılımı, doğrusal olmayan bir sıkıştırıcının (compressor) ardına skaler bir kuantizör ile sağlanmaktadır. Doğrusal olmayan sıkıştırıcı, giriş işaretini istatistiksel özelliklere göre sıkıştırmaktadır. Birbirine benzer işaretler, birine daha az benzeyen işaretlerden daha fazla sıkıştırılır. Sıkıştırmanın karşı tarafında genişletici $C^{-1}(x)$ yer almaktadır. Sıkıştırma ve genişletme işlemleri işaret bozulmasına yol açmamaktadır.



Şekil 2.7. Doğrusal olmayan kuantizasyon

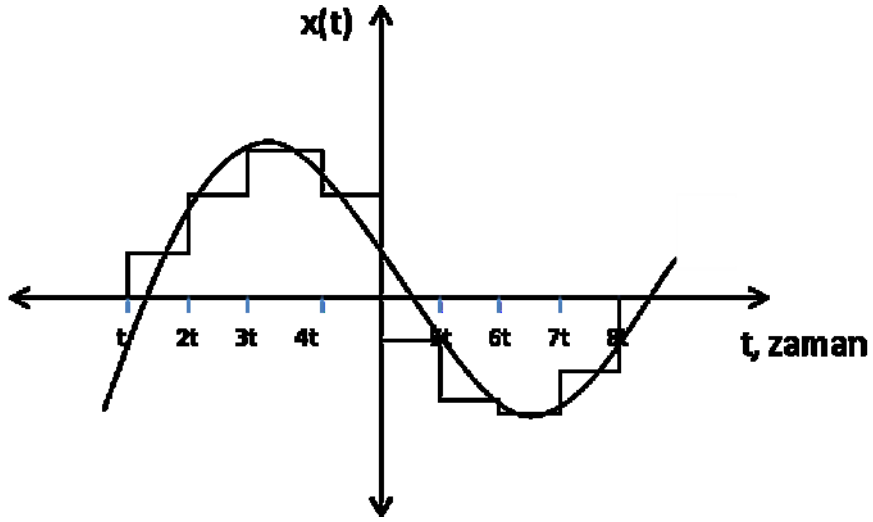
Örtüşme-önler (Anti-aliasing) filtre:

Çoğunlukla, sayısal modellerde analog işaret işleminde örnekleme zamanı minimize edilmek istenir. Bunun nedeni, matematik işlemlerin yoğunluğunun, işlem yapılan örneklerin sayısı ile doğru orantılı olmasıdır. Eğer analog işaret bant-sınırlı değil veya Nyquist frekansı çok yüksek ise ön-filtreleme gerekli olacaktır. Bu, analog işaretin A/S (analog-sayısal) dönüştürücüye girmeden önce bir düşük-geçirgen filtreden geçmesiyle sağlanabilir. A/S dönüştürücüden önce gelen bu düşük-geçirgen filtreye örtüşme-önler filtre adı verilir. Ses sentez sistemlerinde, yüksek kazançlı 3200 Hz. kaskat düşük-geçirgen filtre A/S dönüştürücüden önce kullanılır.

Analog Sayısal Dönüştürücü:

A/S dönüştürücüler, girişteki analog voltaj veya akım seviyesini, en yakın kuantizasyon seviyesinin ikilik koduna çeviren fiziksel elektronik ekipmanlardır. Dışarıdan saat kontrollü olup her saat periyodunda A/S dönüşüm başlayıp biter. Ancak, dönüşüm anlık olmadığı için yüksek performanslı A/S dönüştürücülerde

örnekleme ve tutma (holding) bir arada yapılmaktadır. Çok çeşitli çözünürlükteki A/S dönüştürücüler çıkış ikilik serisini, seri haberleşme imkanıyla dış birimlere aktarırlar. Şekil 2.8.'de analog bir işaretin, doğrusal olmayan kuantizasyonu verilmektedir.



Şekil 2.8. Doğrusal olmayan kuantizasyon ile örnekleme

3. KONUŞMA TANIMA İŞLEMİ

Konuşma, insanlar için en uygun ve en etkili iletişim yoludur. Konuşma tanıma teknolojisi, konuşularak aktarılan bilgileri alabilen ve bu bilgiler üzerinde değerlendirme yapabilen sistemleri geliştirmeyi ve üretmeyi amaçlar. Bir sonraki amaç, bilgi akışını sadece insandan makinalara değil konuşma sentezi yoluyla makinalardan insana sağlamaktır. Konuşma tanıma çalışmaları, konuşulan mesajları alabilen, anlayabilen, bu mesajlara göre harekete geçen ve konuşarak bilgi alışverişini tamamlayan yapay zekalı makinaların bir fonksiyonunun parçası olarak ele alınabilir.

Konuşma tanıma algoritmaları, konuşmacıya bağımlı veya konuşmacıdan bağımsız olarak iki farklı mod kullanırlar. Bu modların belirlenmesi öğrenme moduna bağlıdır. Konuşmacıya bağımlı tanımada, tek bir konuşmacının konuşma çıktıları, konuşma işlem modelini karakterize eden parametrelerin belirlenmesinde kullanılır. Böylece, konuşmacıya bağımlı tanıma sistemi, tanıma sonuçları açısından konuşmacıdan bağımsız tanıma sisteminden daha başarılıdır. Konuşmacıdan bağımsız tanıma sisteminde ise test modunda, öğrenme modundakilerden farklı konuşmacılarla tanıma gerçekleştirilmektedir. Konuşmacıya bağımlı tanımanın temel dezavantajı her bir yeni konuşmacı için öğrenme modunu tekrarlanmasıdır. Tanımda kesinlik ve tanıma sisteminin uygun seçimi gereğine göre yapılmalıdır. Örneğin, bir telefon sistemindeki sorulara cevap veren sistem konuşmacıdan bağımsız tanımayı kullanmalıdır. Halbuki, bir güvenlik sistemindeki konuşmayı tanımda eğitim modu söz konusu olan kişi tarafından gerçekleştirilmelidir.

Konuşma tanıma sistemleri veya algoritmaları, sözcük hazneleri açısından, küçük, orta ve geniş ölçekli olmak üzere incelenirler. Çeşitli literatürlerde nicelik olarak değişiklikler olmakla birlikte sözcük haznesi, 1-99 sözcük olanlar küçük, 100-999 olanlar orta, 1000 sözcük ve daha yukarısı geniş sözcük hazneli konuşma sistemi olarak sınıflandırılırlar. Genel olarak konuşma tanıma sistemleri üç ana grupta incelenebilir.

3.1. Konuşma Tanıma Sistemleri:

Ayrık Sözcük Tanıma:

Tanıma kullanılacak kısım, kelimenin tamamıdır ve diğer kelimelerden ayrılmıştır. Ard arda gelen kelimeler birbirlerinden yeteri kadar uzun duraklamalarla ayrılmış gibi görünseler de (minimum 200 ms.) zayıf frikatifler ve plosiflerdeki boşluklar zemin gürültüsünden kolayca ayırt edilemezler. Ayrık kelimeleri, tek kelimedenden oluşan sonsuz uzunlukta duraklamaları olan cümleler olarak kabul edebiliriz. Dolayısıyla, konuşma tanımanın temel taşlarından son-nokta kontrolü problemi çözümlmelidir. Son-nokta kontrol algoritması, basitçe, bir kelimenin başlangıç ve bitiş sınırlarını veya muhtemel adaylarını belirlemek için kullanılmaktadır. Bu basit tanıma stratejisi, kendisi ile işbirliği yapan konuşmacıları gerektirmektedir. Örneğin, kelimeyi duraklamalı olarak söylemek. Eğer kelime dağarcığı geniş ise, ayrık kelime tanıma sistemi ikincil kelime modelleri için özel olarak tasarlanmalıdır. Eğer ayrık kelimelerden oluşan cümlelerin tanınması söz konusu ise tanıma performansı, cümle içindeki kelimeler arasındaki olasılık hesaplarına dayanan ilişkilerin incelenmesini gerektirir.

Sürekli Konuşma Tanıma:

En karmaşık tanıma sistemi olan sürekli konuşma, konuşmacı mesajlarının bir bölümü veya tamamı üzerinde kısıtlayıcı kurallar koymaz. Böyle bir sistem, birincisi, akustik işaretin tahmin edilemeyen sınırlarını ele alabilecek kapasitede olmalıdır. İkincisi sistem, çeşitli ses teşkili problemlerinin olduğu ortamlarda iyi sonuç vermelidir. Sürekli konuşma tanıma sistemi, konuşmacı açısından en doğal tanıma sistemidir. Küçük sözcük dağarcıkları ve birtakım kısıtlamaların bulunduğu durumlarda (nümerik dizgilerin tanınması gibi) ayrık sözcük tanıma veya birleşik sözcük tanıma stratejileri uygulanabilir. Bu gibi durumlarda tanınacak her sözcük için sistemin hafızasındaki tüm adresler karşılaştırılır. Eğer sözcük haznesi daha genişlerse ve tanımlama yapısı daha karmaşıklarsa ki bu durum sürekli konuşma tanıma sistemidir, her sözcük için öğrenme-hafızaya kaydetme işlemi ve ikincil

sözcüklerin (heceler ve fonemler) dikkate alınması genellikle mümkün olmayabilir. Daha karmaşık arama algoritmalarının tasarlanması kaçınılmaz olacaktır. Bu tip algoritmaların en belirgin olanı, anlamsız veya gramer olarak yanlış yapıları elimine edecek olan dile (dilbilimine) ait olan kuralları içeren algoritmalarlardır. Dile ait olan kuralları içeren algoritmaların uygulanması (leksikal, sentetik ve semantik kurallar) bir dil modelinin ortaya konulması anlamına gelebilir.

Birleşik Sözcük Tanıma:

Küçük sözcük hazneli sürekli konuşma tanıma uygulamalarında, birleşik sözcük tanıma algoritmaları kullanılmaktadır. Burada birleşik sözcükten kastedilen, sözcüklerin birleşik olması değil bir konuşma tanıma stratejisi olmasıdır. Sözcükler arasındaki olasılığa dayanan ilişkiler biliniyorsa, bunlar birleşik sözcük tanıma yaklaşımında performansı iyileştirmek için kullanılabilir. Örneğin, yedi rakamlı telefon numaralarını söylerken ikinci kullandığımız sözcük yüz'dür (100). Eğer bu tip kurallara sadık kalınırsa tanıma başarısı artacaktır.

Bütünlük sağlanması açısından, konuşmacı tanıma ve doğrulanması problemleri konuşma tanıma problemleri ile yakın ilişkilidir. Konuşmacı tanıma amaç, test grubu içerisinde kim konuşmaktadır sorusunu cevaplamaktır. Kullanılan teknikler her iki problem için özde aynıdır.

Konuşma tanıma probleminin önemli zorluklarından birisi de zemin gürültüsüne karşı güçlü sistemler gerektirmesidir. Zemin gürültüsü; başka bir konuşmacının konuşması, cihazların çıkarttığı sesler (fluoresent lambanın gürültüsü, klimanın gürültüsü gibi), konuşmanın sisteme aktarıldığı kanalın gürültüsü, mikrofon değişiklikleri ve hatta konuşmacının dudak hareketleri, nefes alması, öksürük ve burundan gelen sesler olarak toplanabilir.

3.2. Sayısal Konuşma İşleminde Temel Birimler

Konuşma İşaretinin Çerçevesi :

Analog işaretlerin birçoğu zaman içerisinde durgun kalmazlar. Bu daha çok konuşma işareti için geçerlidir. Konuşma işareti birkaç on mili-saniyede bir değişmektedir. Bu sebeble konuşma işareti kısa zaman aralıklarında analiz edilmelidir. Bu zaman aralıklarına *çerçeve* adı verilir. Çerçeveler, 15 ile 30 mili-saniye uzunlukta olabilir. Bu çerçevelerde konuşmanın kısa zaman özelliklerinin çıkarılması amaçlanır. Çerçeve, konuşma işaret dizisinin kaydırılmış *pencere* ile çarpılması olarak ifade edilebilir.

$$f_s(n, m) = s(n) \cdot w(m - n) \quad (3.1)$$

$s(n)$, konuşma işaret örneklerini, $w(n)$ ise pencere katsayılarını ifade etmektedir. Pencerenin seçimi de önem taşımaktadır. Bu seçimi etkileyebilecek iki faktör vardır. Birincisi, seçilen pencere, konuşma işaretini etkilememeli ve ikincisi, pencerenin sınırlarında yumuşak geçişler yapmasıdır. Doğrusal öngörülü kodlama analizinde, öngörü hatası çerçevenin başlangıcında ve bitiminde çok büyük olmaktadır. Çünkü, çerçevenin başlangıcında, konuşma işareti sıfır olan işareten tahmin edilmekte aynı biçimde çerçevenin bitiminde sıfır olan işaret, önceki konuşma örneklerinden tahmin edilmektedir. Konuşma tanıma sistemlerinde genellikle Hamming penceresi tercih edilmektedir. Hanning ve Blackman pencereleri de sayısal ses analiz işlemlerinde kullanılmaktadır [4].

$$w(n)_{Hamming} = \begin{cases} 0.54 - 0.46 \cos(2\pi \frac{n}{N-1}), & 0 \leq n \leq N-1 \\ 0, & \text{diğer} \end{cases} \quad (3.2)$$

$$w(n)_{Hanning} = \begin{cases} 0.5 - 0.5 \cos(2\pi \frac{n}{N-1}), & 0 \leq n \leq N-1 \\ 0, & \text{diğer} \end{cases}$$

$$w(n)_{Blackman} = \begin{cases} 0.42 - 0.5 \cos(2\pi \frac{n}{N-1}) + 0.08 \cos(4\pi \frac{n}{N-1}), & 0 \leq n \leq N-1 \\ 0, & \text{diğer} \end{cases}$$

Ön Vurgu Filtresi :

Ön vurgu (preemphasis) filtresi aşağıdaki gibi bir transfer fonksiyonuna sahiptir.

$$P(z) = 1 - \mu z^{-1} \quad (3.3)$$

Doğrusal öngörülü kodlama önce kullanılır. μ 'nün değeri 0,9 ile 1 arasında değişir. Konuşma işaretine bu filtrenin uygulanması yüksek frekans spektrumunun bağlı enerjisinin artmasını sağlar. Doğrusal öngörü parametreleri hesaplanmadan önce bu filtrenin uygulanmasının iki temel nedeni vardır. Birincisi, sistem fonksiyonuna $z=1$ civarında bir sıfır ekelenmesi, gırtlak ve dudaklardan kaynaklanan spektral etkileri elimine eder. Böylelikle doğrusal öngörülü analiz, sadece vokal yol parametrelerinin elde edilmesini sağlar. Konuşma üretme sistemi modeli, karmaşık fiziksel bir sistemin basit analitik modelidir. Bu modelin bir parçası olarak ön vurgu filtresi, doğrusal öngörülü spektrumu boğaz veya dudakların yayılımlarından bağımsız kılar. İkinci sebep ise nümerik karasızlığı önlemektir. Spektrumun dinamik alanı arttıkça doğrusal öngörülü model, kötü durumda otokorelasyon matrisleri ile sonuçlanır. Ön vurgu filtresi, konuşma işaretinin dinamik alanını azaltır.

Son-nokta Kontrolü :

Sayısal konuşma dizini içersinde son-noktaların yerlerinin belirlenmesi önemli bir problemdir. Dinamik zaman çakıştırma uygulamalarında, iki konuşma paterni arasındaki toplam mesafe minimize edilmeye çalışılırken, son-nokta kontrolü sistem performansı açısından önem taşımaktadır. Son-nokta kontrolünde kullanılan algoritmalar, enerji ve sıfır geçiş kavramlarını kullanırlar. Son-nokta kontrolü, gürültülü ortamlarda ve kanal karakteristiği zaman zaman değişen sistemlerde oldukça güç bir problem haline gelir. Bu sistemlerde, enerji ve sıfır geçiş ölçümleri iyi sonuç vermez. Bu sebeble laboratuvar koşulları tercih edilir.

Zayıf frikatiflerle başlayan (h,t,f) veya sesli frikatiflerle başlayıp sessiz frikatiflerle biten (örneğin, has) sözcük paternlerinde son-nokta kontrolü zordur. Bu sesleri zemin gürültüsünden ayırt edebilmek için sıfır geçişleri kullanılmaktadır. Konuşma içindeki frikatifler zemin gürültüsünden daha çok sıfır geçişine sahiptir.

Konuşma işaretinin enerji sıralarının ve sıfır geçişlerinin doğru olarak belirlenmesi, gürültünün az olduğu ortamlarda, son-nokta kontrolü sonuçlarının başarılı olmasını sağlar. Önce enerji seviyelerine göre son-noktalar bulunur ardından sıfır geçiş sayılarına göre son-noktalar düzeltilir. Enerji seviyelerinin hesaplanmasında, çerçeve uzunluğu N için kısa zaman enerji ölçümü m anında biten çerçeve için aşağıdaki gibi tanımlanır.

$$E(m) = \sum_{n=m-N+1}^m |s(n)| \quad (3.4)$$

Enerji ölçümünde, konuşma işaretinin mutlak değeri kullanılmaktadır. Bunun sebebi, enerji seviyesi düşük kısımlar ile yüksek kısımlar arasındaki farkı artırmaktır. Bazen konuşma işaretinin karesi de hesaplamalarda kullanılır.

$$E(m) = \sum_{n=m-N+1}^m s(n)^2 \quad (3.5)$$

Son-noktalara karar vermek için önce her bir çerçeve için enerji seviyeleri yukarıdaki formüller ile hesaplanır. İlk birkaç çerçeve için konuşma kaydı yapılmadığını kabul edersek, bu çerçeveler için ortalama, minimum ve maksimum enerji seviyeleri hesaplanır. Daha sonra da enerji için aşağı ve yukarı eşik değerleri hesaplanır.

ITL, aşağı eşik seviyesi ve ITU ise yukarı eşik seviyesini göstermektedir. Buradaki a ve b değerleri uygulamaya ve kanal özelliklerine göre değişir. Genel olarak a yaklaşık 0.03 ve b yaklaşık 3 olarak seçilebilir.

$$I1 = a * (\max_enerji + \min_enerji) + \min_enerji \quad (3.6)$$

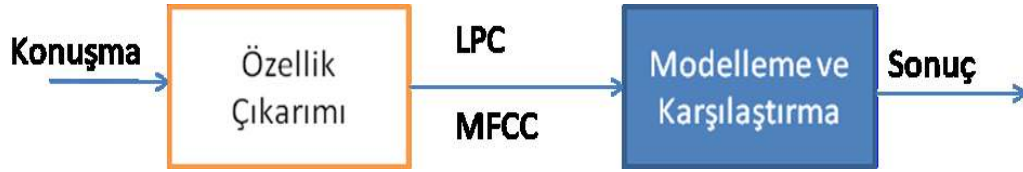
$$I2 = \text{ortalama enerji}$$

$$ITL = \min(I1, I2)$$

$$ITU = b * ITL$$

Parametrik Ses Özelliklerinin Çıkartılması:

En genel haliyle, bir konuşma tanıma sistemi (Şekil 3.1.), iki temel blok yapıdan oluşur: ses örneklerinin çıkartılması ve modelleme-karşılaştırma yapılması. Buradaki hedef, ses tanımayı kolaylaştıracak temel parametreleri, mümkün olan en kapsamlı bir şekilde orjinal konuşma örneğinden elde ederken, bunu mümkün olan en az veri kaybı ile gerçekleştirmektir.



Şekil 3.1. Klasik konuşma tanıma sistemi

Ses örneklerinin özelliklerinin çıkartılması, geri dönüşü olmayan, kayıplı, bir işlemdir. Hem referans konuşma şablonlarının oluşturulmasında hem de test ses örneklerinin karşılaştırılması aşamalarında kullanılır. Bu işlem sonucunda elde edilen parametreler, konuşma örneğinin tam olarak yeniden oluşturulmasında etkin değildir, ancak konuşma örneğini tanımak veya konuşmacıyı belirleyebilmek için çok iyi araçlardır. Matematiksel hesaplamalardaki karmaşıklık ve sistemin kararlılığı, ses bilgisinin kaybını iki temel sebebidir. Parametrik ifadelerin hasasiyetinin artması ve sonuç olarak daha fazla matematiksel işlem, sistem stabilitesini arttırmamaktadır.

Konuşma tanıma süreci, analog ses işaretlerinin kaydedilmesi ile başlar. Bu aşamada *sesin sayısallaştırılması* işlemi gerçekleştirilir (Şekil 3.2.). Araştırmalar göstermiştir ki, ses işareti kısa zaman aralıkları dahilinde sabit karakteristik özellikler taşımaktadır. Bu sebeple, konuşma işareti, *pencere* adını verebileceğimiz, 20-30ms'lik bölümlere ayrılırlar. Herbir pencerenin komşu olan iki pencere ile daha önceden belirlenmiş uzunlukta bir bölümünü içermesi suretiyle, bu pencereler arasındaki geçişler düzeltilmiş olmaktadır.



Şekil 3.2. Analog ses işaretinin, sayısallaştırılması

Sayısallaştırılmış ses işaretlerinin özellikleri çıkartılırken, ses işaretlerinin düzeltilmesi amacıyla filtreler kullanılır. Genellikle bu amaçla, *Hamming penceresi* kullanılmaktadır. Bu pencereleme fonksiyonu sayesinde, pencereler arasındaki

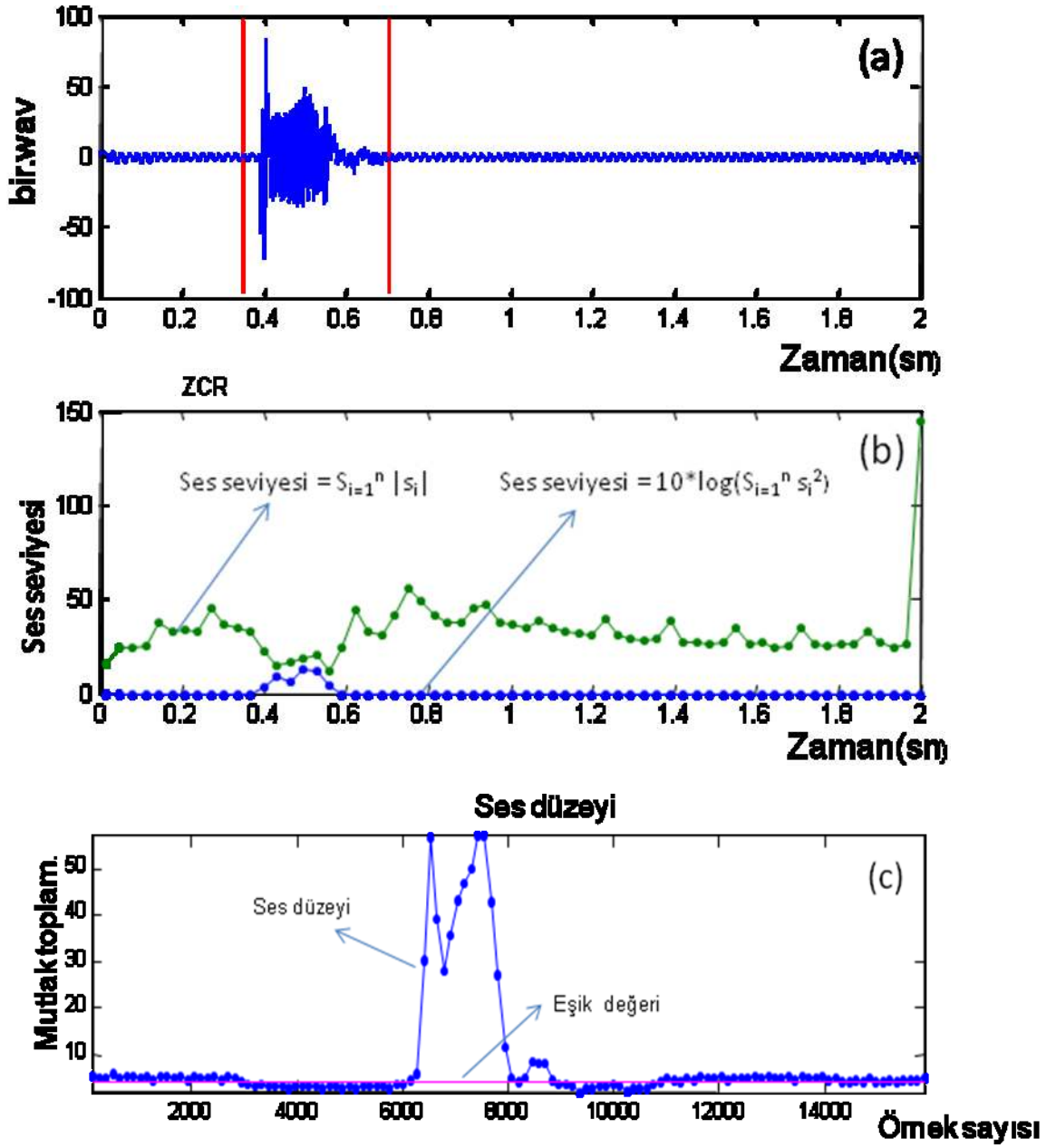
süreksizlikler giderilmekte ve ses işaretinin merkezi belirginleştirilmektedir. Bu çalışmada, ayrıca Hanning ve Blackman pencerelerinden de faydalanılmıştır.

Ses özelliklerini içeren parametreler her bir pencere için ayrı ayrı hesaplanır. Bu noktada herbir pencerenin, ses işareti içerip içermediği kontrol edilir. Bu amaçla herbir pencerenin konuşma örneğinin başlangıç ya da bitiş *uç-noktası* olup olmadığını tespit etmek için;

- *logaritmik enerji seviyesine*, ses şiddetine, ortam gürültüsünden ayırt edilecek seviyede olup olmadığı kontrolü

- *sıfır-geçiş skor* kontrolü yapılır [1].

Aşağıda, Türkçe “bir” kelimesi için kaydedilmiş 2sn sürelik konuşma örneği için yapılan, enerji seviyesi ve sıfır geçiş skor kontrolüne ait analiz verilmektedir (Şekil 3.3.).



Şekil 3.3. 2sn. süre uzunluğunda “bir.wav” ses kaydı ve uç-noktaların tespiti (a), sıfır geçişleri (b), enerji değeri (c).

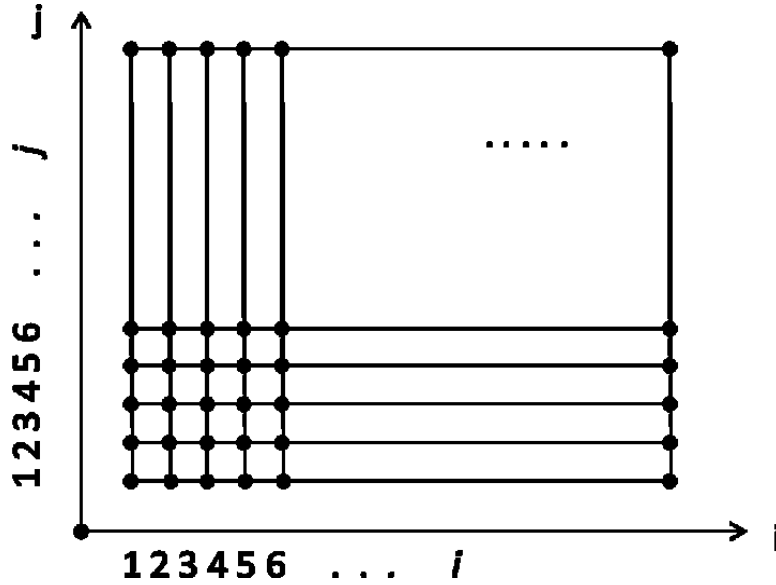
Son olarak, yukarıda belirtilen üst üste binen pencerelere ayrılmış, Hamming pencere filteresi uygulanmış ve uç-noktaları belirlenmiş konuşma örneği, popüler ses analiz tekniklerinden LPC veya MFCC uygulanarak, ses örneğinin özellikleri çıkartılır [9].

Şablon Karşılaştırma:

Dijital bilgisayarlardan yararlanan modern konuşma tanıma algoritmalarının tümü, iki ana teknik yöntem kullanırlar. Bunlardan birincisi ve bu çalışmada kullanılacak olan temeli patern karşılaştırmasına dayalı yöntemidir. Bu sınıftaki yöntemler, özelliklere dayalı istatistiki patern tanıma problemlerinden çıkarılmıştır. Fakat konuşma tanıma problemi diğer patern karşılaştırma problemlerinden önemli farklar içermektedir. Bu farklar, bir karşılaştırma skoru hesaplamadan önce test konuşmasının ve referans konuşmasının özelliklerinin karşılaştırılmasını, kaydırılmasını, sıkıştırılmasını gerektirir. Bu problemin çözümü için dinamik programlama kullanılır. Dinamik programlamanın çeşitli biçimlerinden birisi de konuşma tanımanın kalbini oluşturmaktadır. Dinamik programlama, matematiksel bir kavram olarak işlemlerde optimal karar vermeyi işler. Çakıştırma terimi, zaman ekseninde kaydırma, sıkıştırma yapılarak test ve referans konuşma özelliklerini dinamik programlama tekniği ile karşılaştırmayı içermektedir. Bu sebeple özellik karşılaştırma yöntemi, konuşma tanıma yaklaşımında *dinamik zaman çakıştırması* olarak adlandırılır.

Dinamik zaman çakıştırma, asgari bir donanım ve yalın bir algoritma gerektirmesine karşı basit uygulamalar için oldukça başarılıdır. Bu yöntem , ayırık sözcük tanıma ve hatta birleşik sözcük tanıma stratejilerini kullanan sürekli konuşma tanıma için de başarılı olarak kullanılır. Yöntem, herbir tanınacak konuşma için bir referans paterni veya birbirine bağlı paternler kümesi gerektirdiği için konuşma kaynaklarındaki değişikliklere göre genelleme yapmak mümkün değildir. Ayrıca, geniş sözcük hazneleri içeren karmaşık tanıma problemlerinde tercih edilmez. Sürekli konuşma tanıma işleminde, birleşik sözcük tanıma durumu dışında kullanılmaz.

Dinamik programlamayı, Şekil 3.4 ‘deki ızgara yapısı üzerinde düşünelim. Izgara yapısı, kartezyen düzleminin birinci çeyrek dairesi ve sıra ile eşleşmiş düğüm olarak adlandırabileceğimiz nokta çiftlerini içermektedir.



Şekil 3.4. Dinamik Programlama'yı açıklayacak ızgara yapısı

Temel problem, ızgara üzerinde orjinal düğüm $(0,0)$ 'dan başlayan ve çıkış düğümü (I,J) 'ye olan “en kısa mesafenin” veya “en az maliyetli yolun” bulunmasıdır. Bir yol için mesafe veya maliyet üç biçimde oluşmaktadır: Geçiş, düğüm ve her ikisi (geçiş ve düğüm) .

$d_T[(i_k, j_k)|(i_{k-1}, j_{k-1})] \equiv (i_{k-1}, j_{k-1})$ düğümünden (i_k, j_k) düğümüne geçiş mesafesi,

$d_N(i, j) \equiv (i, j)$ düğümünün maliyeti,

$d_B[(i_k, j_k)|(i_{k-1}, j_{k-1})] \equiv d_T[(i_k, j_k)|(i_{k-1}, j_{k-1})] + d_N(i_k, j_k)$ ise her ikisinin maliyeti olarak tanımlanabilir. $(0,0)$ 'dan (i_k, j_k) 'a olan toplam yol, D ise

$D = d[(i_k, j_k)|(i_{k-1}, j_{k-1})]$ geçişlerin ve düğümlerin toplamıdır.

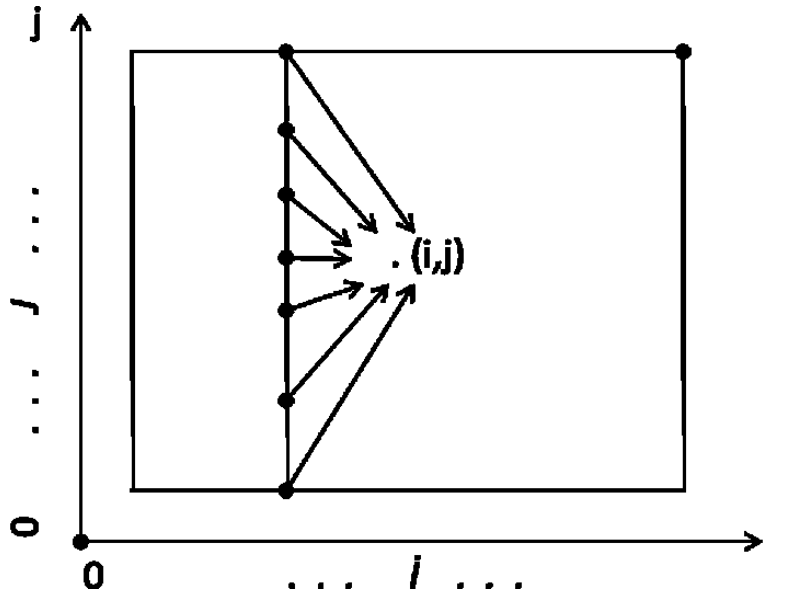
En kısa mesafe veya en az maliyetli yol problemi için çözüm Dinamik Programlama algoritmalarında *Bellman optimallik prensibi* ile sağlanır. Gelişi güzel seçilmiş

(s, t) 'den başlayan ve (u, v) 'e biten yolu $(s, t) \rightarrow_* (u, v)$ olarak gösterilirse en uygun

yolu da $(s, t) \xrightarrow{*(w,x)} (u, v)$ olarak gösterilirse Bellman optimallik prensibine göre

$$(s, t) \xrightarrow{*(w,x)} (u, v) \equiv (s, t) \rightarrow_* (w, x) \oplus (w, x) \rightarrow_* (u, v) \quad , \quad 0 \leq s, w, u \leq I \quad \text{ve} \quad 0 \leq t, x, v \leq J$$

olarak tanımlanır. Bellman prensibi, bir yolun minimum olabilmesini kendisi oluşturan ara yol parçacıklarının da minimum olmasına bağlamaktadır. Örneğin, Şekil 3.5. ileriye doğru ilerleme kısıtı olan bir yol için (i, j) düğümüne olan olası mesafeler gösterilmektedir.



Şekil 3.5. (i, j) 'e düğümüne, bütün geçişlerin ileriye doğru olduğu bir yol yapısında optimal yol $(i-1, j)$ 'dir.

Dinamik zaman çıkıştırma yaklaşımı, ayrık sözcük ve birleşik sözcük tanıma algoritmaları için oldukça yalın ve olgun bir teknoloji zemini oluşturmaktadır. Birçok pratik konuşma tanıma sisteminde kullanılmıştır. Hatta, sadece bu tekniği uygulayan VLSI devre entegreleri tasarlanmıştır. Fakat, dinamik zaman çıkıştırma yaklaşımının birkaç sınırlaması vardır. Konuşmacıdan bağımsız sistemler için yaklaşık 100 sözcük kapasitesi ve konuşmacıya bağımlı sistemler için yaklaşık 500 sözcük kapasitesi vardır. Daha geniş sözcük hazneli uygulamalar için hafızaya kaydedilen patern

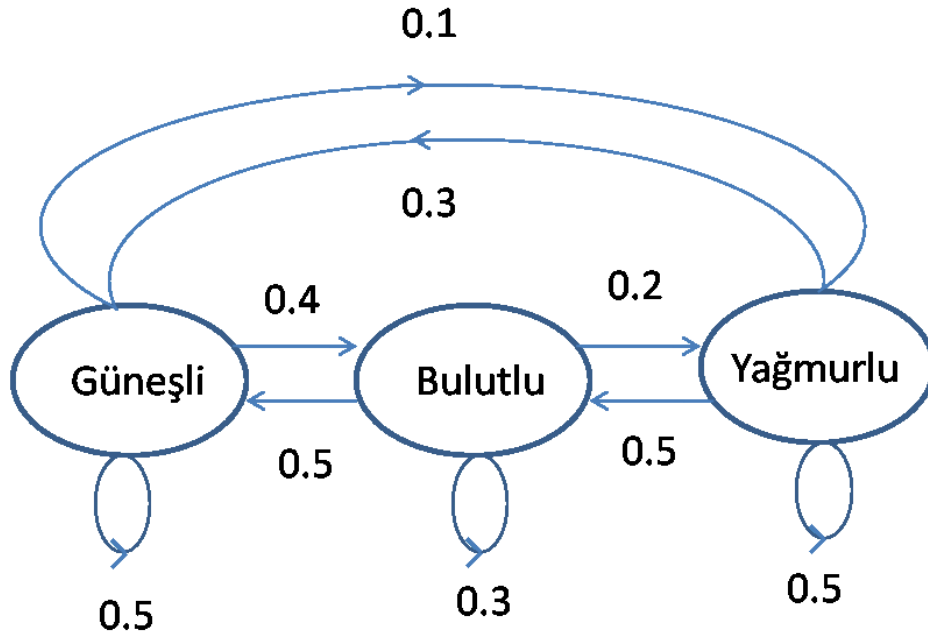
sayısının ve patern karşılaştırmaların doğrusal olarak artması uygulamayı değersiz kılmaktadır. 1970 ve 1980’lerde konuşma araştırmacıları, daha büyük ölçekli sistemler için değişkenlik problemini çözebilecek stokastik yaklaşımlar uygulamaya başladılar. Stokastik yaklaşım terimi, konuşmada değişkenliği karakterize eden modelleri işaret etmektedir. Bu durum, dinamik zaman çıkıştırma yaklaşımında olduğu gibi patern karşılaştırmasına dayalı modellere ters düşmektedir. Stokastik yaklaşım yerine yapısal yaklaşım da kullanılabilir, çünkü, bu modeller önemli matematiksel yapılara dayanmaktadır.

İki farklı stokastik yöntem üzerinde araştırmalar yapılmaktadır. Birincisi, alışılmış sayısal bilgisayar hesaplamalarını kullanan *saklı Markov yöntemi*dir. Konuşma araştırmaları içinde, son otuz yılda mühendisliğin ilgi alanı saklı Markov yöntemi üzerine olmuştur. Bu yöntem, laboratuvar ve ticari alanlarda birçok geniş ölçekli başarılı konuşma tanıma sisteminin temeli olmuştur. Buna karşı, ikinci stokastik yöntem olan *yapay sinir ağları*, daha çok biyolojik sinir ağları sistemine paralel çalışabilecek alternatif hesaplama mimarilerinin geliştirilmesi üzerine yoğunlaştığı için bu alanda konuşma tanıma üzerine yapılan çalışmalar saklı Markov yöntemi yaklaşımının gerisinde kalmıştır.

Saklı Markov yönteminin geçmişi, neredeyse konuşma alanında kullanılmasıyla başlamış ve bundan sonra daha yaygın kullanılmıştır. Saklı Markov yönteminin, konuşma tanıma çalışmalarına girişi, 1975 yılında Canegie-Mellon Üniversitesi’nden Baker ve IBM’den Jelinek ve arkadaşlarının çalışmalarıyla olmuştur. 1980’lerde ise yazılı patern tanıma teorileriyle saklı Markov modeli arasında yakın ilişkiler ortaya konulmaya başlamıştır.

Markov zincirleri teorisinin temeli bu yüzyılın başlarına kadar geçmişe uzansa da saklı Markov modelinin konuşma alanında kullanılmasından önce Levinson ve Poritz ‘in bu konudaki çalışmaları bilinmektedir. 1950’lerde istatistikçiler, rastgele işlemlerde tamamlanmamış gözlemler problemi üzerinde çalışırken bu modelin temellerini ele almışlardır. Onların probleme yaklaşımları kararsız stokastik işlemler şeklinde olmuştur. Yani, bir sensör vasıtasıyla gözlenen data, geçen doğru (saklı)

işlemin sonucu olarak düşünülmekte bu sensör ise gözlenen ikinci işlemi üretmektedir. Şekil3.6.'da hava koşulları için basit bir Markov yöntemi gösterilmektedir. Herbir durum diğer bir duruma dallanabilmekte ancak her dallanmanın bir olasılığı mevcuttur. Dolayısıyla bir durumun oluşma olasılığı, kolayca bir önceki durumdan belirlenebilmektedir. Konuşma tanıma sistemleri Markov yöntemini, test konuşmasının olasılığının modellenmesinde kullanırlar. Fakat, konuşma problemi daha karmaşıktır. Çünkü, konuşma tanıma sistemi her hangi bir zamanda hangi durumda olduğunu bilemiyebilir. Bu Markov modelinin saklı kısmıdır. Hava koşulları örneğine dönecek olursak sıcaklık ve nem oranı bilgileriyle, geçmiş günlerin hava koşullarını belirleyebiliriz.



Şekil 3.6. Hava koşulları için Saklı Markov modeli

3.3 MFKK (Mel-Frekans Kepstral Katsayıları)

Sayısal ses işleminde, mel-frekans kepstrum (MFC), bir ses örneğinin, kısa süreli güç spektrumu ifadesidir. Bu ifade, doğrusal olmayan “mel-skala” frekansları üzerinde, logaritmik güç spektrumunun doğrusal kosine transformu ile belirtilir. Kepstral (MFCC) katsayılar ise, o ses örneği için mel-frekans kepstrumu oluştururlar. Mel-skala, sesin temel perdelerinin (pitch), birbirlerinden eşit uzaklıklarda konumlanmış

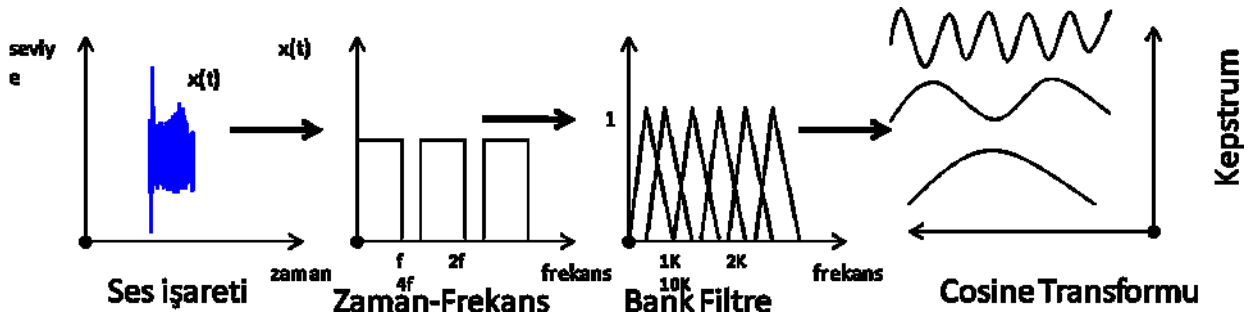
dinleyiciler tarafından algılandığı basamakları ifade eder. Bu skala ile gerçek ölçülen frekans arasındaki referans noktası, işitme eşiğinin 40 dB üzerindeki 1000 Hertz frekansında bir ses işaretinin, 1000 mel basmağında bir ses perdesi olarak tanımlanmasıdır.

$$c(n) = \frac{1}{N_s} \sum_{k=0}^{N_s-1} \log_{10} |S(k)| e^{(2\pi / N_s k_n)}, \quad 0 \leq n \leq N_s - 1 \quad (3.7)$$

Yukarıdaki eşitlikte, $c(n)$, kepstum olarak adlandırılır. $S(k)$ değeri ise, kepstum değerinin ait olduğu frekans aralığı için alınan Fourier dönüşümünü belirtir. N_s , o andaki çerçevenin boyutunu gösterir. Örneğin, $c(0)$ değeri, spesifik olarak, o andaki “Discrete Fourier Transform” (DFT) spektrum değerini göstermektedir. Sesli ifadenin, gürültüden ayırt edilmesi için önce spektrumun logaritması alınır ve sonra ters Fourier dönüşümü yapılır. Bu şekilde belirlenen kepstum değerleri Fourier dönüşümünden türetilmiş kepstal katsayılar olarak adlandırılır. Burada hesaplanan kepstum değeri, sesli ifade tanımada kullanılan önemli bilgileri elde etmede etkilidir. Fourier dönüşümünde kullanılan frekans, mel skalasında örneklenirse elde edilen kepstum değerleri, MFC değerleri olarak adlandırılır .

MFC katsayıları genellikle, aşağıdaki adımlar izlenerek hesaplanır:

1. Herbir ses örneğinin, ait olduğu pencere için Discrete Fourier Transform’u (DFT) hesaplanır.
2. DFT katsayılarının Mel filtre bankasının, genlik frekans cevabına göre ağırlıkları hesaplanır.
3. Herbir mel frekans basmağı için logaritmik enerjiler hesaplanır.
4. Enerji değerlerine göre, sesli örnekler için, mel logaritmik enerjilerinin, Discrete Cosine Transformu (DCT) hesaplanır.
5. MFC katsayıları, elde edilen spektrumun genlik değerleridir.



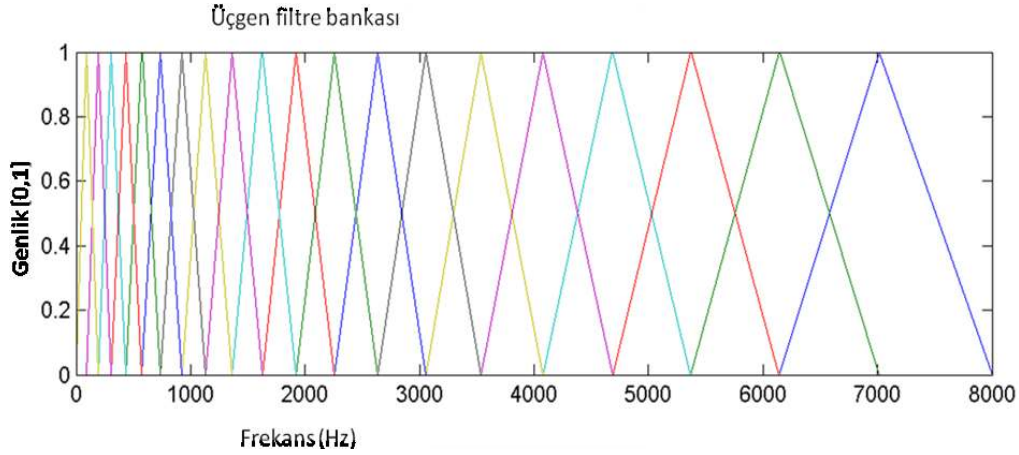
Şekil 3.7. Mel-Frequency Cepstrum Coefficients (MFCC) hesaplanmasının grafiksel gösterimi .

İnsan işitme sisteminin karakteristik özelliklerine baktığımızda, kulağın, işitme frekanslarını, işitme spektrumu boyunca, doğrusal olmayan bir biçimde çözümlendiğini ve deneysel bir kanıt olarak, benzer şekilde, doğrusal olmayan bir ön-sonu (front-end) filtre tasarımıyla, konuşma tanıma performansını geliştirdiği gözlenmiştir. İnsan kulağının frekans yanıtını karakterize eden ölçüye Mel-Ölçüsü veya Melodi Ölçüsü de denir. Bark ve Mel işitme sistemi tepki modelleri ile yapılan deneylerde aşağıdaki frekans cevapları elde edilmiştir [1].

Çizelge 3.1. Bark ve Mel basamakları için, işitme spektrumu boyunca, merkez frekansa karşılık gelen kritik bant genişlikleri

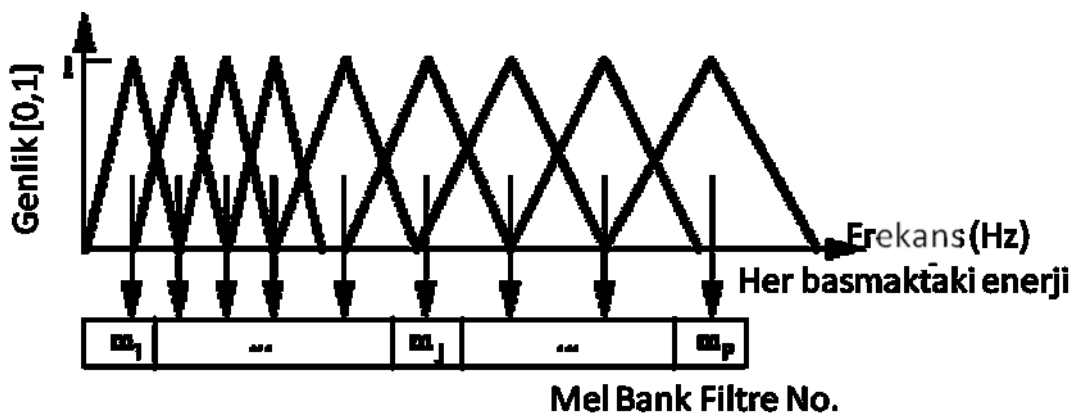
Filtre No	Bark		Mel	
	Frekans (Hz)	Bant genişliği (Hz)	Frekans (Hz)	Bant genişliği (Hz)
1	50	100	100	100
2	150	100	200	100
3	250	100	300	100
4	350	100	400	100
5	450	110	500	100
6	570	120	600	100
7	700	140	700	100
8	840	150	800	100
9	1000	160	900	100
10	1170	190	1000	124
11	1370	210	1149	160
12	1600	240	1320	184
13	1850	280	1516	211
14	2150	320	1741	242
15	2500	380	2000	278
16	2900	450	2297	320
17	3400	550	2639	367
18	4000	700	3031	422
19	4800	900	3482	484
20	5800	1100	4000	556
21	7000	1300	4595	639
22	8500	1800	5278	734
23	10500	2500	6063	843

Mel filtrebandı, bir dizi üçgen bant-geçirgen filtreden oluşan, insan işitme sisteminin ön bant geçirgen filtre yapısını taklit ederek tasarlanan bir süzücüdür. Bu yapı, sabit bant-genişliğinde, mel merkez frekansında konumlanmış bant-geçirgen filtreler karşılık gelmektedir. Bu filtrelerin frekans karakteristiği, 1KHz'e kadar doğrusal ve 1KHz üstü logaritmiktir.



Şekil 3.8. Üçgen Filtre Bankası

Üçgen band-geçirgen filtre seçimi, genlik spektrumunun düzleştirilmesi ve matematiksel olarak daha küçük boyutlu ses özelliklerine indirgenmesi açısından avantajlar sunar. Mel filtreleri, frekans aralığı, 0 Hz frekansından, örnekleme yapılacak frekansın iki katına eşit olan Nyquist frekansına kadar dağılmışlardır. Ancak, filtre bankları (Şekil 3.9.), özellikle, ses işaretinin kayda değer enerjisinin olmadığı durumlarda ve örneğin, ortam gürültüsü gibi istenmeyen ses bileşenlerinin varlığında, alt ve üst limit frekans değerleri ile bu bölümlerin eliminasyonunda yardımcı olmaktadır.



Şekil 3.9. Mel basamakları filtre bankı

3.4. Doğrusal Öngörülü Kodlama, DÖK :

Ses analiz teknikleri arasında, en etkin kodlama ve sıkıştırmaya imkan verenlerden birisi, doğrusal öngörülü kodlama analizdir. Bu yöntem, perde, formant, spectra, vokal yol fonksiyonları gibi temel ses parametrelerinin tahmin edilmesi ve düşük bit oranlarında ses ile haberleşme ve sesin depolanması için sistemin ifade edilmesinde vazgeçilmez bir tekniktir. Bu yöntemin önemi, ses parametrelerinin büyük oranda doğrulukla tahmin edilmesi ve matematiksel işlemlerdeki hız üstünlüğüdür. Temel olarak DÖK analizi, ses örneğinin, geçmiş ses örneklerinin doğrusal kombinasyonu olarak ifade edilmesi prensibine dayanmaktadır. Sonlu bir zaman aralığı için gerçek ses örnekleri ile doğrusal olarak tahmin edilen örneklerinin kareleri farkının toplamının minimize sonucu öngörü katsayıları elde edilir. Bu öngörü katsayıları, doğrusal kombinasyonların ağırlık katsayıları olacaktır. Ses üretme sistemini daha önce doğrusal, zamanla değişen , yarı-periyodik pulslar (sesli konuşma) veya gelişigüzel gürültüler (sessiz konuşma) ile sürülen bir sistemin çıktısı olarak modellemiştik. DÖK böyle bir sistemi kabul eder ve sistemi karakterize eden parametrelerin tahmin edilmesinde güvenilir ve hızlı bir yol izler. DÖK yönteminde, doğrusal terimi, ses dalga şeklinin modellenmesi probleminde birbirine eş, çok çeşitli formülasyon tiplerinin kullanılıyor olmasıdır. DÖK tekniğinde kullanılan belli başlı formülasyonlar şunlardır [1]:

1. Kovarians yöntemi
2. Otokorelasyon formülasyonu
3. Lattice yöntemi
4. Ters filtre formülasyonu
5. Spectral tahmin formülasyonu

Ses üretme sistemini modellediğimiz doğrusal, zamanla değişen sayısal filtrenin durgun hal sistem fonksiyonu şöyledir:

$$H(z) = \frac{S(z)}{X(z)} = \frac{G \left[1 - \sum_{j=1}^M b_j z^{-j} \right]}{1 - \sum_{i=1}^N a_i z^{-i}} \quad (3.8)$$

Yukarıdaki sistem fonksiyonunda hem kutup hem de sıfır mevcuttur. Paydanın yüksek kuvvetleri için $H(z)$, sıfırlardan bağımsız ve sadece kutuplarla ifade edilirse

$$H(z) = \frac{G}{1 - \sum_{j=1}^p a_j z^{-j}} = \frac{G}{A(z)} \quad (3.9)$$

Yukarıdaki denklemin örneklenen zaman bölgesine dönüşümü sonucu

$$s(n) = Gx(n) + \sum_{j=1}^p \alpha_j s(n-j) \quad (3.10)$$

elde edilir. Bu denklem DÖK fark denklemi olarak bilinir. Denklem, $s(n)$ çıktısını, şimdiki giriş, $Gx(n)$, ve geçmiş çıktıların ağırlıklı toplamıyla ifade edildiğini göstermektedir. Öyleyse, DÖK tekniği, $s(n)$ gibi verilen bir işaret için a_j parametrelerinin belirlenmesiyle sistemi tanımlayan $H(z)$ fonksiyonunun elde edilmesidir. $s(n)$ örneklerinin, geçmiş örneklerden elde edilmesi işlemindeki hata ya da kalan ise $e(n) = s(n) - \sum_{j=1}^p b_j s(n-j)$ dir. b_j , DÖK tahmin katsayılarını ifade etmektedir. Hata örnekleri incelendiğinde, hatanın sistem fonksiyonu aşağıdaki gibi olan bir sistemin çıktısı olduğu anlaşılır.

$$A(z) = 1 - \sum_{j=1}^p b_j z^{-j} \quad (3.11)$$

Eğer ses işareti, DÖK fark denklemini sağlıyor ise ve ideal olarak $a_j = b_j$ ise $e(n) = Gx(n)$ dir. Tahmin hata filtresi $A(z)$, sistemin ters filtresidir. Hatanın karesini minimize eden parametre seti

$$E = E\{e^2(n)\} = E\left\{\left[s(n) - \sum_{j=1}^p b_j s(n-j)\right]^2\right\} \quad (3.12)$$

ifadesinden elde edilir. b_j değerleri, $\partial E_n / \partial b_j = 0$ ifadesini $j=1,2,3,\dots,p$ değerleri için çözümüdür. Sonuç olarak aşağıdaki denklem elde edilir:

$$\sum_m s_n(m-i) s_n(m-j) = \sum_{j=1}^p b_j \sum_m s_n(m-i) s_n(m-j) \quad (3.13)$$

i , 1 ile p arasında bir değerdir. m ise örnekleme zamanı içinde sonlu bir zamandır.

$\phi_n(i, j) = \sum_m s_n(m-i)s_n(m-j)$ olarak $\phi(i, j)$ fonksiyonu tanımlarsak

$$\sum_{j=1}^p b_j \phi_n(i, j) = \phi_n(i, 0) \quad i=1,2,3..p \quad (3.14)$$

olur. p kadar denklem, p kadar bilinmeyen DÖK katsayıları için bulunur. Hata ifadesi ise

$$E_n = \phi_n(0,0) - \sum_{j=1}^p b_j \phi(0, j) \quad (3.15)$$

olur. Belirli zaman aralığı için, toplam minimum hata DÖK katsayılarına bağlıdır. Ses üretme modelleri için sıkça kullanılan vokal yol, birbirine bağlı kayıplı akustik tüpler olarak düşünülebilir. Herhangi bir andaki tüplerin toplam kesitini $A(x)$ alan fonksiyonu ile gösterirsek p kadar kesit için tüp modeli, DÖK analiz ile elde edilen sistem fonksiyonuyla aynı özelliktedir. DÖK katsayılarını bulmak için yukarıdaki p kadar bilinmeyenli, p kadar denklemden oluşan sistemi çözmek üzere çeşitli formülasyonlar kullanılabileceğinden bahsetmiştik. Şimdi, otokorelasyon yöntemi kullanarak DÖK katsayılarını bulalım. Giriş işareti, $s_n(m)$, $0 \leq m \leq N-1$ zaman aralığı dışında sıfır genliğe sahip olduğunu düşünelim. N , ise örneklenen dizinin uzunluğudur [1].

$$\phi_n(i, j) = \sum_{m=0}^{N+p-1} s_n(m-i)s_n(m-j) \quad 1 \leq i \leq p, 0 \leq j \leq p \quad (3.16)$$

$$\phi_n(i, j) = \sum_{m=0}^{N-1-(i-j)} s_n(m)s_n(m+i-j) \quad 1 \leq i \leq p, 0 \leq j \leq p \quad (3.17)$$

Yukarıdaki denklem, kısa zaman otokorelasyon fonksiyonuna dönüştürülürse,

$$\phi_n(i, j) = R_n(|i-j|) \quad i=1,2,3..p \quad j=1,2,3..p \quad (3.18)$$

$$R_n(j) = \sum_{m=1}^p s_n(m)s_n(m+j)$$

$$\sum_{j=1}^p b_j R_n(|i-j|) = R_n(i) \quad 1 \leq i \leq p \quad (3.19)$$

Matriks formunda ise

$$\begin{bmatrix} R_n(0) & R_n(1) & \dots & R_n(p-1) \\ R_n(1) & \dots & \dots & R_n(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ R_n(p-1) & \dots & \dots & R_n(0) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ \vdots \\ R_{n(p)} \end{bmatrix} \quad \text{dir.} \quad (3.20)$$

NxN otokorelasyon matrisi, Toeplitz matristir. Yani, simetrik ve diagonal üzerindeki elemanları eşittir. NxN otokorelasyon matrisinin tersini alarak yapılacak hesaplamalar oldukça karmaşık yapılardır. Otokorelasyon matrisinin çözümü için verimli tekrar yerine koyma yöntemleri geliştirilmiştir [1]. Levinson yöntemi, bu tip denklemleri çözmek için iyi bir yöntemdir. Durbin'in yöntemi ise Levinson'un yönteminden iki kat daha hızlıdır :

$$\begin{aligned} E^{(0)} &= R(0) \\ k_i &= \frac{R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} \cdot R(i-j)}{E^{(i-1)}} \quad , \quad 1 \leq i \leq N \\ \alpha_i^{(i)} &= k_i \\ \alpha_j^{(i)} &= \alpha_j^{(i-1)} - k_i \cdot \alpha_{i-j}^{(i-1)} \quad , \quad 1 \leq j \leq i-1 \\ E^{(i)} &= (1 - k_i^2) \cdot E^{(i-1)} \end{aligned} \quad (3.21)$$

Son iterasyonda çözüm :

$$\alpha_j = \alpha_j^{(N)} \quad , \quad i \leq j \leq N \text{ 'dir.}$$

Levinson-Durbin algoritması olarak bilinen tekrar yerine koyma yöntemi aşağıdaki gibidir [2]:

Verilen $R(j)$, $j = 0, 1, \dots, p$ için

Adım1: $E^{(0)} = R(0)$

Adım 2: $i=1$ (3.22)

$$\begin{aligned} \text{Adım 3: } k_i &= \frac{R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j)}{E^{(i-1)}} \\ \alpha_j^{(i)} &= k_i \end{aligned}$$

Adım 4: $j=1$

Adım 5 : $\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$

Adım 6 : $j=j+1$

Adım 7 : Eğer ki $j \leq i$ ise Adım 5 'e git !

Adım 8 : $E^{(i)} = (1 - k_i^2) E^{(i-1)}$

Adım 9 : $i=i+1$

Adım 10 : Eğer ki $i \leq p$ ise Adım 3'e git !

Adım 11: $\alpha_i = \alpha_i^{(p)}$, $i = 1, \dots, p$

3.5. Dinamik Zaman Çakıştırma, DZÇ

Dinamik zaman çakıştırma yöntemi bu çalışma boyunca ayrıık sözcük tanıma algoritmalarına uygulanabililiği açısından ele alınacaktır. Dinamik zaman çakıştırma, referans ve test patern özelliklerinden oluşan kümelerin karşılaştırılması algoritmasını içeren tekniktir. Temel düşünce, bu iki kümenin zaman açısından birbirine uyacak şekilde sıralanmasını sağlamaktır [1]. Dinamik zaman çakıştırma sürecinde , test patern özellikler kümesi, doğrusal olmayan bir şekilde sıkıştırılarak veya genişletilerek referans patern özellikler kümesine uydurulmaya çalışılır. Bu, patern karşılaştırılması yaklaşımında kritik bir yapıdır; çünkü, bir kelimenin veya kelimeler grubunun söylenmesi konuşmacının o andaki koşulları ile ilişkilidir. Test ve referans kümeleri arasındaki mesafe, karşılaştırmanın ne kadar iyi olduğunun ölçütüdür. Yani daha az maliyet, daha iyi bir karşılaştırmadır.

Dinamik zaman çakıştırmasında, herbir tanınacak sözcük veya sözcük grupları için bir referans patern özellikler kümesi oluşturmak gerekir. Dolayısıyla geniş sözcük haznesi gerektiren karmaşık sistemler için tercih edilmez. Bu tekniğin başarısı, son-noktaların doğru olarak belirlenmesine bağlıdır.

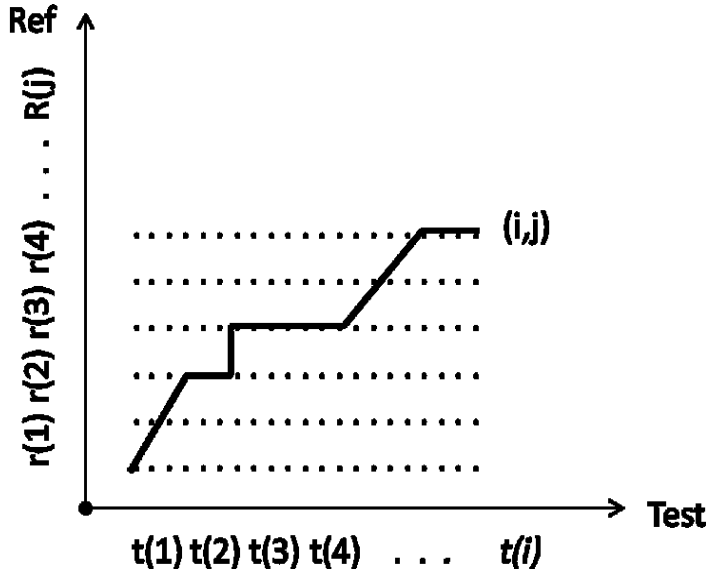
Konuşma tanıma sistemi, daha önceden hafızaya kaydedilmiş özellikler kümesinin, verilen özellikler kümesi ile karşılaştırılması uygulamasıdır. İki küme karşılaştırılırken, birbirlerine en iyi uyacak şekilde yani maliyet veya karşılaştırma hatası minimum olması düşünülür. $T(i)$, $i=1,2,\dots,M$ kümesi test konuşmasının

özellikler kümesi, $R(j)$, $j=1,2,\dots,L$ kümesi ise daha önceden hafızya kaydedilen ilgili referans özellikler kümesi olsun. Bu özellik vektörleri, konuşma örneğinin çerçeveler içindeki doğrusal öngörü katsayıları veya kepstral katsayılar gibi her tür spektral bilgi olabilir. Burada, $i=1,2,\dots,M$ ve $j=1,2,\dots,L$ çerçeve indisleridir.

Test : $T(1), T(2), T(3), T(4), T(5), T(6), \dots M$ (test kümesi uzunluğu)

Ref : $R(1), R(2), R(3), R(4), R(5), R(6), \dots L$ (referans kümesi uzunluğu)

Şekil 3.10.'da (i,j) düğümünün $c(i,j)$ olan maliyeti, $(1,1)$ düğümünden (i,j) düğümüne kadar mümkün olan bütün düğüm kombinasyonlarının minimum toplam mesafesi olarak tanımlanır [1].



Şekil 3.10. İki özellik kümesinin çakışan noktalarından oluşan yol

$T(i)$ ve $R(j)$ çerçeveleri arasındaki mesafe,

$$d[T(i), R(j)] = [T(i) - R(j)]^T \cdot [T(i) - R(j)] \quad (3.23)$$

olarak ifade edilir. Maliyet $c(i,j)$ ise

$$c(i, j) = \min \sum_{k=1}^i \sum_{m=1}^j d[T(k), R(m)] \quad (3.24)$$

olası bütün yollar için tanımlanır. Bütün düğüm noktaları için mesafeler hesaplandıktan sonra bir yol bulunur. Yol, $(1,1)$ düğümünden başlayıp (M,L) düğümünde bitecektir. Yol, gerçekte test ve referans kümelerinin çakışması sonucu

ortaya çıkmaktadır. Yolun toplam maliyeti D , (M,L) düğümünün maliyetidir. Sonuçta karar verilen yol, minimum toplam maliyetli yol olacaktır. Düğümlerin maliyetleri hesaplanırken bazı kısıtlamalar uygulanır. Bu lokal ve global yol kısıtlamaları, dinamik zaman çakıştırma yönteminin konuşma tanıma performansını etkiler. Dinamik zaman çakıştırmasına ait algoritma aşağıda görülmektedir.

$$\begin{aligned} D_{\min}(1,j) &= d_N(1,j), \quad j = 1, \dots, \varepsilon \\ \text{Başlangıç : } D_{\min}(i,1) &= d_N(i,1), \quad i = 1, \dots, \varepsilon \\ \delta_1(j) &= D_{\min}(1,j), \quad j = 1, \dots, J \\ \delta_2(j) &= 0, \quad j = 1, \dots, J \end{aligned}$$

$$\begin{aligned} \text{Tekrar : } & \text{For } i=2, \dots, I \\ & \text{For } j=J, \dots, 2 \end{aligned}$$

$$D_{\min}(i,k) = \min_{(i_{k-p}, j_{k-p})} \left\{ D_{\min}[i_{k-p}, j_{k-p}] + \sum_{m=0}^{p-1} d[i_{k-m}, j_{k-m} | i_{k-m-1}, j_{k-m-1}] \right\}$$

$$\delta_2(j) = \delta_1(j)$$

$$\delta_1(j) = D_{\min}(i,j)$$

$$(D_{\min}[i-2, j]) \text{ değeri } \delta_2(j) \text{ 'de ve } D_{\min}[i-1, j] \text{ değeri } \delta_1(j) \text{ tutulur.})$$

Next j

Next i

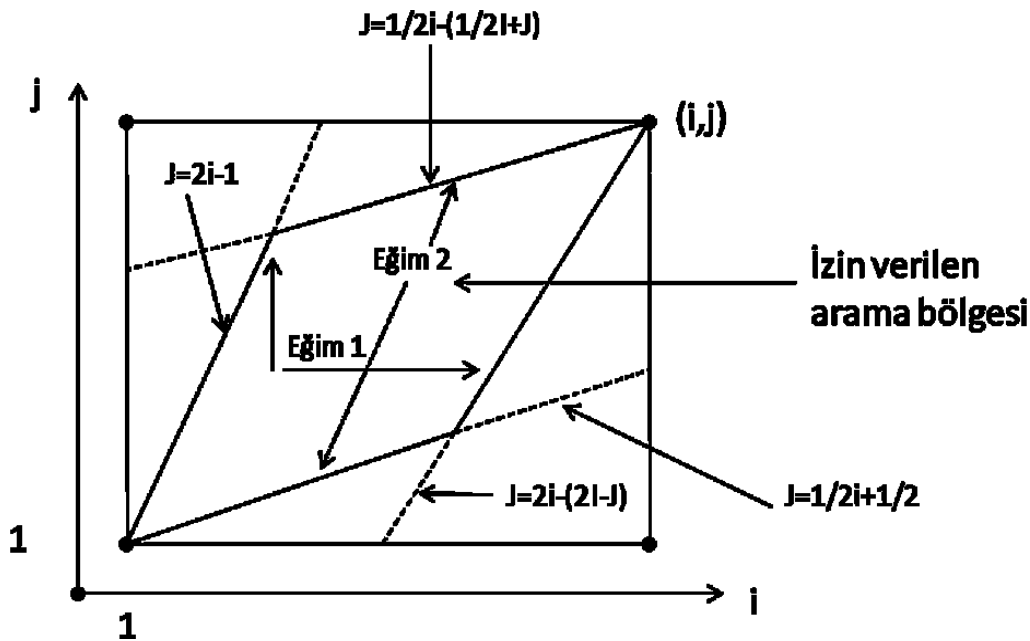
$$\text{Sonuç : } \text{En iyi yol}$$

$$D_{\min} = \begin{cases} D_{\min}(I,j)/I & , \quad j = J - \varepsilon, \dots, J \\ D_{\min}(i,J)/I & , \quad i = I - \varepsilon, \dots, I \end{cases}$$

Dinamik zaman çakıştırma yönteminde, ızgara üzerinde yol arayışları oldukça yoğun işlemler gerektirmektedir. Fakat performansı artırmak için gerek işlem miktarındaki sınırlamalar gerekse test ve referans dizilerinin karşılaştırmasının mümkün olmayacağı bölgelerde dahi yapılması, bir takım kısıtlamaların kullanılmasını gerektirmiştir. Dört tip kısıtlama kullanılmaktadır.

Son-nokta kısıtlamaları, en genel şekli ile test ve referans dizilerinin son-noktalarının örtüşmesini gerektirmektedir. Bazı durumlarda son-noktalar bilinmemekte ve dinamik zaman çakıştırması arayışları sırasında bulunmaktadır. Son-nokta kısıtlamaları $T(1)$ düğümünü $R(1)$ düğümüne mümkün olan yollardan biri ile eşlerken, $T(I)$ düğümünü de $R(J)$ düğümüne mümkün olan yollardan biri ile eşler. Dolayısıyla minimum yol için araştırılacak bütün yollar $(1,1)$ düğümünden başlayıp (I,J) düğümünde son bulur.

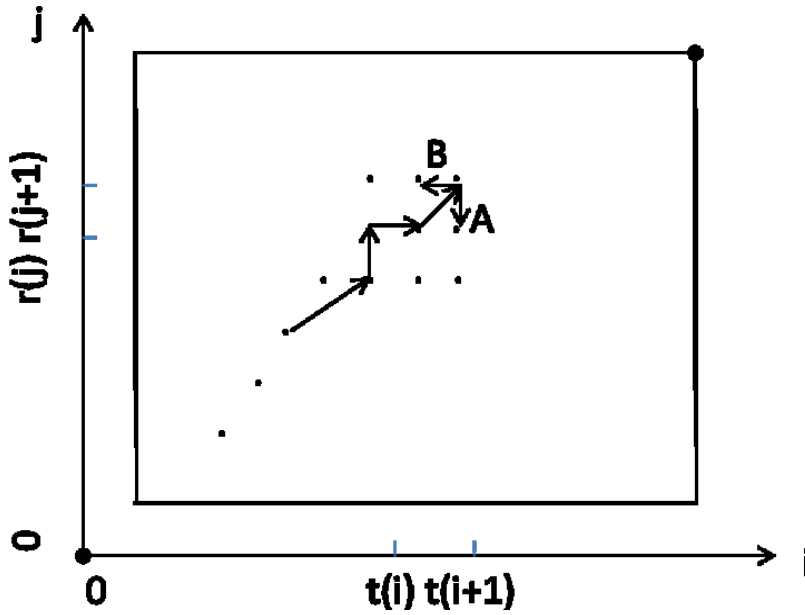
Yol için bir başka kısıtlama, *monoton* olmalıdır. Yani $i_{k-1} \leq i_k$ ve $j_{k-1} \leq j_k$ olmalıdır. Bu ise söz konusu olan yolun hiçbir zaman güneye veya batıya gitmemesi demektir. Pratik anlamda, test patern özellikleri hiçbir zaman daha önceden çakıştırılmış referans patern özellikleri ile tekrar çakıştırılmazlar. Yolun kendi üzerinde tekrar geri dönüşünü önlenmesi, birçok yanlış çakıştırmalar sonucu başarılı karşılaştırmaların önlenmesi demektir (Şekil 3.11.).



Şekil 3.11. Monotonluk kısıtlaması, yol üzerinde hiçbir zaman negatif i veya negatif j yönünde geçiş yapmasına izin vermez.

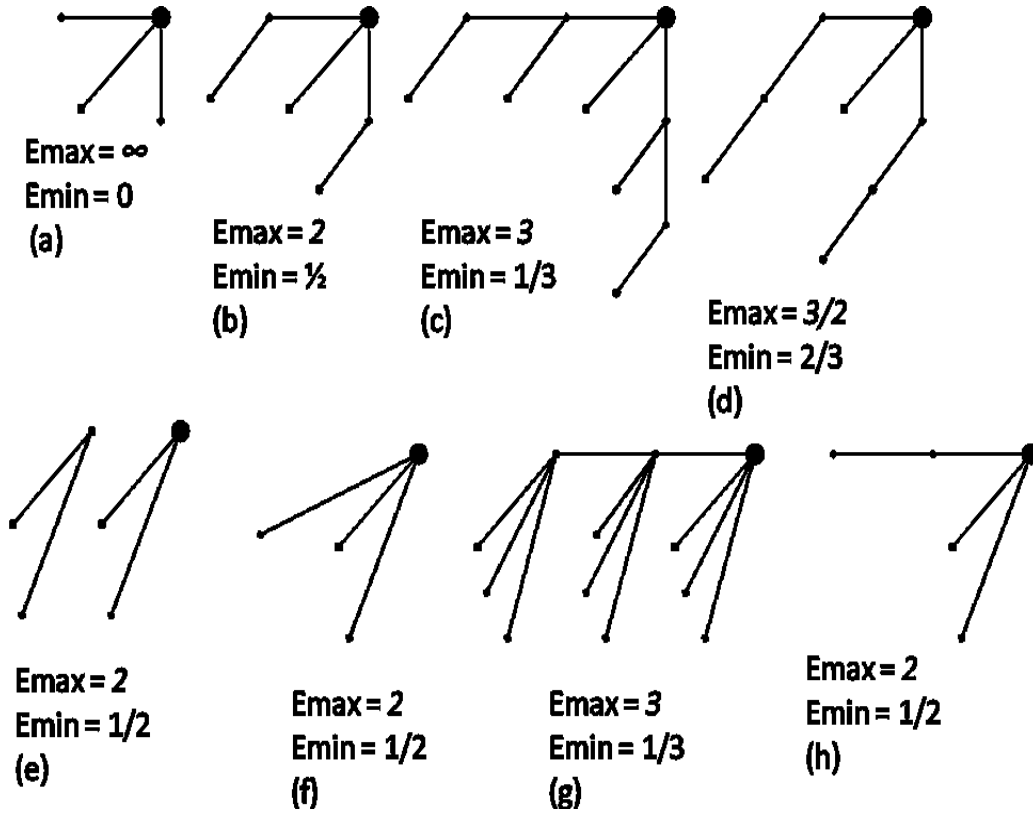
Global yol kısıtlamasında, test dizisi referans dizisine uydurulurken; bu kısıtlama test dizisindeki uzun sıkıştırma veya genişletmelerin yayılması durumunda etkin

olmaktadır. Yani kısaca, muhtemel yolların geçmemesi gereken bölgeleri belirlenerek ızgara üzerindeki düğümlerin sayısını azaltmak suretiyle yol arayışları kolaylaştırılır. En çok kullanılan global kısıtlama Itakura tarafından önerilmiştir [1]. Şekil 3.11.'de sıkıştırma ve genişletme katsayısı 2 olan arama alanı gözükmemektedir. En kötü durum yollarını $(1,1)$ ve (I,J) için çizerek yani sıkıştırma katsayısı 2, eğim 0.5 değerinde ve genişletme katsayısı 2, eğim 2 değerinde aşağıdaki gibi bir paralelogram elde edilir. Bu paralelogramın iç bölgesi ve sınırları arama için uygun düğüm noktalarını kapsar. Sıkıştırma veya genişletme katsayısı 2 ve $I \approx J$ durumu için ızgarada I^2 yerine $I^2/3$ kadar düğüm noktası kullanılır. İkinci global kısıtlama olarak, hiçbir düğüm pencere genişliğinden, W , daha uzak bir yol üzerinde olmaz. Yani (i_k, j_k) düğümünün üzerinde bulunduğu yol için $|i_k - j_k| \leq W$ sınırlaması konulabilir.



Şekil 3.12. Itakura'nın global yol arama kısıtlaması, sıkıştırma ve genişletme katsayısı 2

Diğer bir kısıtlama tipi ise lokal olanlardır. Lokal yol kısıtlamalarında amaç test paterninin sıkıştırma ve genişletme oranını bir sonraki komşu düğüme kadar olan kısımda gerçekleştirmektir. Birçok araştırmacı tarafından çeşitli lokal yol arama kısıtlamaları tanımlanmıştır (Şekil 3.13.).



Şekil 3.13. Lokal kısıtlamalar : (a)’dan (d)’e kadar Sakoe ve Chiba, (h) Itakura ve (b), (e)’den (h)’e kadar Myers et al tarafından önerilmiştir [1].

3.6. Ortam Gürültüsünün Azaltılması

Gürültü, işaret ve dataların kalitesini düşüren, istenmeyen elektrik veya elektromanyetik enerjidir. Gürültü çeşitlerini, frekans spektrumuna ve zamana bağlı olarak iki ana grupta toplayabiliriz. Frekans spektrumuna bağlı olarak gürültü, "beyaz gürültü", her frekans merkezinde belirli bir bant aralığında, düz güç spektrumuna sahip gelişigüzel bir işarettir. Bir başka deyişle, beyaz gürültüyü meydana getiren sesin frekansı, tüm frekans boyunca yayılmıştır. Tabiatta mevcut bulunan bütün renklerin karışımı nasıl beyaz ışığı meydana getirirse, bütün frekans aralıklarına sahip sürekli spektrumlu sesler de beyaz gürültüyü meydana getirir. Zamana bağlı olan gürültüler ise gözlem süresi boyunca gürültü seviyesinde önemli değişiklikler yaratan gürültülerdir. Tipik gürültü kaynakları Çizelge 3.2.’de verilmiştir.

Çizelge 3.2. Tipik gürültü kaynakları

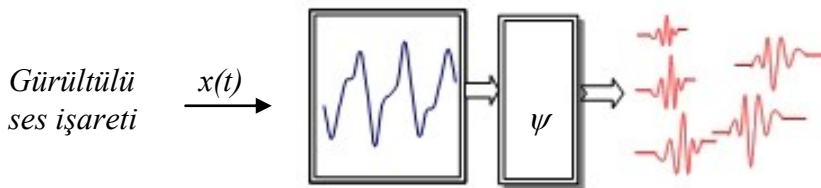
Yer	Gürültü kaynağı
Havaalanı	Uçak motoru, anons, müzik
Araba	Araba motoru, korna
Toplantı	Fısıltı, müzik, deklanşör
Restorant	Fısıltı, arka plan konuşmaları
İstasyon	Anons, ayak sesleri
Cadde	Motor sesleri, korna, alarm, siren
Tren	Ray, vagon sesleri

Konuşma tanıma sisteminde, haberleşme, biyomedikal, matematik gibi birçok işaret işleme uygulamasında başarıyla ve kolaylıkla kullanılmakta olan, “*ayrık dalgacık dönüşümü*” (denoising with wavelet transform) ile gürültü süzme ele alınacaktır.

Dalgacık dönüşümü matematiksel olarak şöyle ifade edilir:

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \cdot \overline{\psi\left(\frac{t-b}{a}\right)} dt \quad (3.25)$$

Bu denklemde, $a > 0$, $b \in \Re$ olmak üzere a , ölçekleme parametresini; b dönüşüm parametresini, $x(t)$, ses işaretini, ψ , dalgacık fonksiyonunu, $W(a,b)$ da işaretin sürekli dalgacık dönüşümünü Şekil 3.14. belirtir [10].



Şekil 3.14. Dalgacık Dönüşümünü

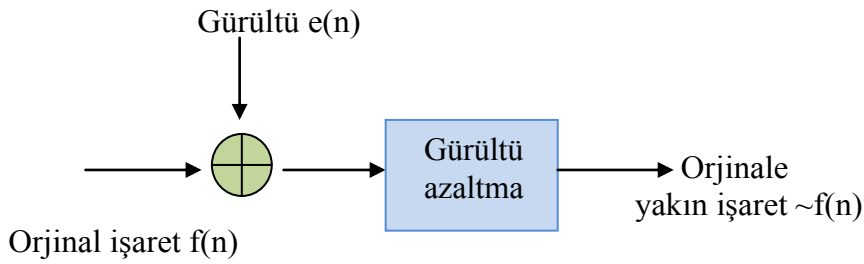
Ters dalgacık dönüşümü ise aşağıdaki formülü ile verilir.

$$x(t) = \frac{1}{C} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{a^2} W(a,b) \psi_{a,b}(t) d_a d_b \quad (3.26)$$

Gürültülü işaret için model aşağıdaki formdadır. Burada n zamanı, eşit aralıklara ayrılmıştır.

$$s(n) = f(n) + \sigma \cdot e(n) \quad (3.27)$$

En basit haliyle bu modelde, $e(n)$ 'nin, $N(0,1)$ 'de Gaussian beyaz bir gürültü olduğu, gürültü seviyesin de $\sigma = 1$ olduğu varsayılacaktır. Gürültü süzmenin (Şekil 3.15.) amacı, s işaretinin gürültülü kısmını bastırmak ve orjinal işaret $f(n)$ 'i elde etmektir [10].



Şekil 3.15. Gürültü süzme modeli [8]

Gürültü azaltmada dalgacık dönüşüm hesaplamaları için sayısal formata dönüştürülen konuşma işaretleri, Matlab programı standard wavelet fonksiyonları kullanılarak, ortam gürültülerinden elimine edilmektedir.

4. KONUŞMACIYA BAĞIMLI AYRIK SÖZCÜK TANIMA SİSTEMİNİN TASARIMI VE GERÇEKLEŞTİRİLMESİ

Konuşmacıya bağımlı, ayrik sözcük sistemi, iki ana yapıdan oluşmaktadır. Birinci yapı, konuşmacının, tanınması için test edileceği sözcüklerin, özelliklerinin çıkarılması ve parametrik olarak ifade edilmesi için şablonların oluşturulmasıdır. İkinci yapı ise, konuşma tanıma için test edilecek sözcüklerin sisteme kaydedilmesi ve sonucun bildirilmesidir.

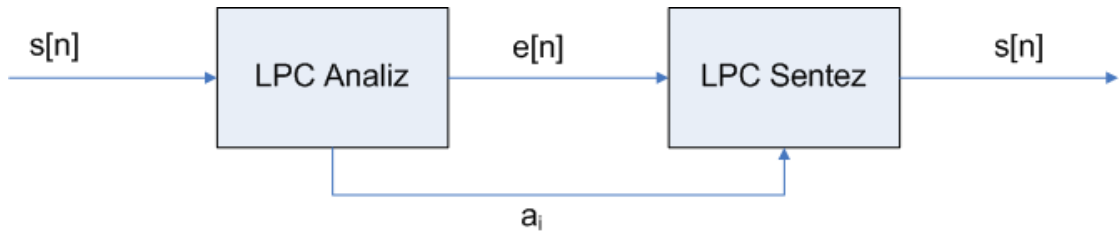
Sistemin gerçekleştirilmesi için bilgisayar ortamı tercih edilmiştir. Matlab, teknik hesaplama programı ve sayısal ses işlemi için birçok denenmiş güvenilir kütüphaneler sunmaktadır. Tüm uygulama programları, Matlab versiyon 9 üzerinden gerçekleştirilmiştir. Geliştirilen program, x486 tabanlı bilgisayarlarda, Microsoft Windows ortamında çalıştırılabilmektedir. Analog ses örneği, bilgisayarın ses kartına bağlanan pasif bir mikrofon aracılığıyla sayısal olarak kaydedilmektedir. Analog sayısal dönüşüm için 8Kbit/s hızında 8-bit çözünürlük tercih edilmiştir. Bu data hızı ve çözünürlük, sistem performansının artırılması için minimumda tutulmuştur. Sayısal ses örnekleri, tipik olarak 20ms'lik pencerelere bölünmüş, pencereler arasında 10ms'lik örtüşmeye imkan verilmiştir. Konuşmacıya bağımlı ayrik sözcük tanıma programının, konfigürasyon seçeneğinden;

- Örnekleme frekansı, Hz
- Örnekleme çözünürlüğü, bit
- Örnekleme süresi, sn
- Pencere tipi(Hamming, Hanning, Blackman)
- Pencere süresi, ms
- Pencere örtüşme süresi, ms değiştirilebilmektedir.

Kaydedilmiş ses örneklerinin ses parametrelerini analiz edebilmek ve performansları karşılaştırmak için, iki farklı teknik ses analiz yöntemi, LPC ve MFCC kullanılmıştır. Konuşmacıya bağımlı ayrik sözcük tanıma programının, ana menü seçeneğinden çalışma yöntemi;

- 39 parametre MFCC
- 11.derece LPC olarak seçilebilmektedir.

LPC analiz, temel olarak, ses örneğinin, geçmiş ses örneklerinin lineer kombinasyonu olarak ifade edilmesidir. Konuşmacıya bağımlı ayrık sözcük tanıma sisteminde 11. derece otokorelasyon formülasyonu kullanılmıştır. Zaman bölgesi ses işaretleri ile karşılaştırıldığında, veri miktarı azaltılmıştır. Bu sebeble, LPC aynı zamanda iyi bir sıkıştırma işlemidir (Şekil 4.1).



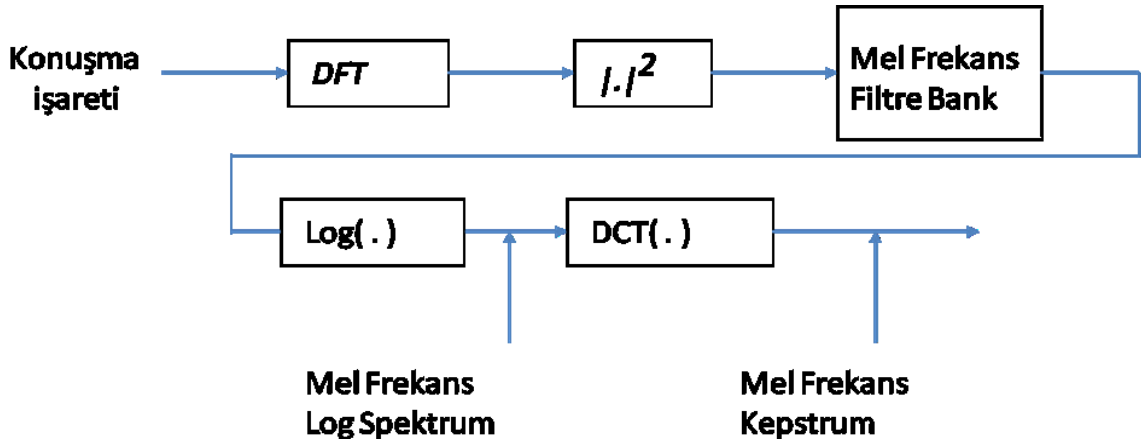
Şekil 4.1 LPC için modelleme

$s[n]$, sayısal konuşma işareti

$e[n]$, hata işareti

a_i , LPC katsayıları

Sayısal ses işleminde, MFC, bir ses örneğinin, kısa süreli güç spektrumu ifadesidir. Mel Kepstrum katsayısını elde etmek için, ilk olarak her bir ses örneğinin, ait olduğu pencere için Discrete Fourier Transform'u (DFT) hesaplanır. DFT katsayılarının Mel filtre bankasının, genlik frekans cevabına göre ağırlıkları hesaplanır. Her bir mel frekans basmağı için logaritmik enerjiler hesaplanır. Enerji değerlerine göre, sesli örnekler için, mel logaritmik enerjilerinin, Discrete Cosine Transform (DCT) hesaplanır. MFC katsayıları, elde edilen spektrumun genlik değerleridir (Şekil 4.2).



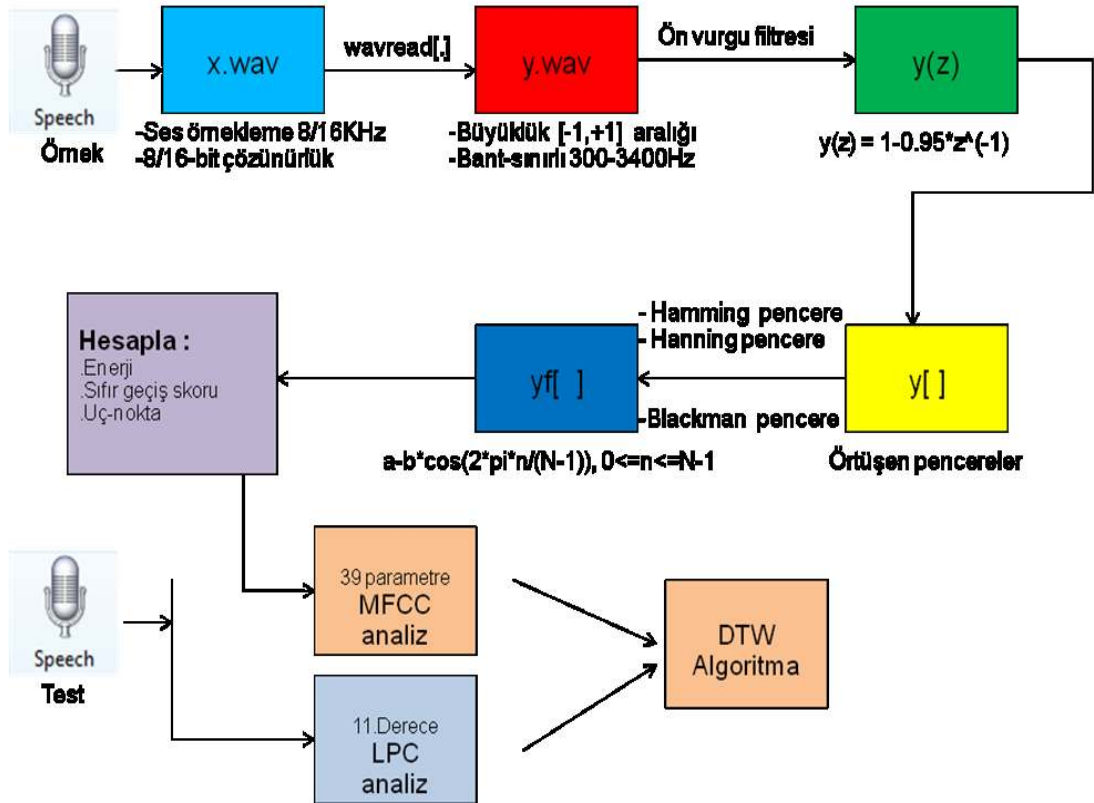
Şekil 4.2. Mel Kepstrum katsayılarının hesaplanması

Yukarıdaki teknikler kullanılarak elde edilen, konuşma örneklerine ait ses parameterinin oluşturduğu, çok vektörlü değerler arasındaki mesafe veya fonetik olarak birbirlerine ne kadar benzedikleri, dinamik zaman çakıştırma (DTW) tekniği ile ölçülmekte ve test için en yakın sonuç, LPC ve MFCC için ayrı ayrı değerlendirebilmektedir.

Bu uygulamada, 0'dan 9'a kadar Türkçe ve İngilizce rakamlar hedef alınarak, konuşmacıya bağımlı olarak, sonuçlar değerlendirilmiştir. Böylece, bu çalışma, Türkçe ve İngilizce'nin, ayrık sözcük tanıma açısından bir karşılaştırılmasını yapma imkanı da vermektedir. Ayrıca, ekler bölümünde, fonetik özellikleri birbirine çok yakın olan (Ali, veli, deli, kedi gibi) Türkçe kelimeler ile, fonetik özellikleri birbirinden çok farklı olan (elma, portakal, muz, mandalina gibi) Türkçe kelimeler ile ve Türkçe birleşik kelimeler (karadudu, karabiber, karafatma, karavana gibi) ile yapılan çalışmaların sonuçlarına da yer verilmiştir.

Konuşmacıya bağımlı ayrık sözcük tanıma uygulamasında, ortamın gürültüsünü test koşullarından elimine etmek için, popüler gürültü azalma algoritmalarından, Ayrık Dalgacık Dönüşümü tekniği kullanılmıştır. Ayrık Dalgacık Dönüşümünde, ayrık ses işaretlerine, ardışıl olarak alçak ve yüksek geçirgen filtreler uygulanmakta ve ortaya çıkan işaretler veri azaltmaya tabii tutulmaktadır. Test sonuçları, gürültü azalma özelliği kullanılarak da karşılaştırılmıştır.

LPC ve MFCC ile KONUŞMACIYA BAĞIMLI AYRIK SÖZCÜK TANIMA SİSTEMİ



Şekil 4.3. LPC ve MFCC ile Konuşmacıya Bağimli Ayrık Sözcük Tanima Sistemi Blok Diagramı

5. SONUÇLAR VE İLERİYE YÖNELİK ÇALIŞMALAR

Bu çalışmada, konuşmacıya bağımlı, ortamdan bağımsız, Türkçe ve İngilizce için ayırık sözcük tanıma sistemi geliştirilmiştir. LPC ve MFCC ses özelliklerinin çıkarılma teknikleri ile oldukça yüksek tanıma oranlarına ulaşılmıştır. Test sözcükler için hazırlanan şablon sözcük haznesi, bu uygulama için, 2 saniyelik 10 sözcük kaydı ile sınırlandırılarak, dinamik zaman çakıştırma tekniğinin, en yüksek performansta çalıştırılması sağlanmıştır. Dinamik zaman çakıştırma tekniği ile bu şablon sözcük limiti, sistem performansı düşürülerek birkaç yüz adete çıkarılabilir. İlerideki çalışmalarda, test ve şablon ses örneklerinin daha geniş ölçekte karşılaştırılması hedeflenirse, stokastik yaklaşımlardan Hidden Markov Yöntemi veya yapay sinir ağları teknikleri kullanılabilir.

Ortam gürültüsünün etkisini azaltmak amacıyla kullanılan ayırık dalgacık transformasyonu, analog ses işareti üzerine binen ve eşik değeri sabit olarak daha önceden kararlaştırılan, gürültü eliminasyonunu yapmaktadır. Ayırık dalgacık transformasyonu, oldukça basit ve tipik gürültü kaynakları için etkili bir teknik olmakla beraber, her ortam gürültüsünü ayrı modelleyen, adaptif gürültü eliminasyon teknikleri, daha ileri seviyeli gürültü eliminasyonuna imkan verecektir.

Uygulamayı ayırık sözcük tanımadan, sürekli konuşma tanımaya doğru bir gelişim içinde düşünülecek olursa, Türkçe dilinin, bitişik sözcükler için oldukça karmaşık modellemeler gerektirebileceği, kayıt süresinin uzamasıyla Türkçe sözcüklerin tanınma oranında düşüşler olabileceği düşünülmektedir.

Türkçe, sözcük sonundan ses eklemeli morfolojik bir yapıya sahiptir. Kök kelimeye, sondan ekler yapılarak, birçok yeni sözcük türetmek mümkündür. 10 milyon ayrı kelimelik dünya dil hazinesi ile 400 bin kelimelik Türkçe dili karşılaştırıldığında, böyle seyreklik, konuşma tanıma sistemi için oluşturulacak, çıkarılması gereken parametre setlerinin sayısını artıracaktır. Birleşmeye yönelik (aglutinatif) dil ailelerinde, kök kelime, birden fazla ek alarak, çok sayıda yeni sözcük türetebilir. Herbir ek, ayrı bir zaman, ayrı bir durum veya ayrı bir yargı gibi morfolojik bilgiler

içerebilir. Bu özellik, birleşmeye yönelik dil ailelerinde, kökleri aynı, sonları farklı biten birçok sözcük anlamına gelmektedir.

Türkçe’de kullandığımız birçok sözcük, dilbilgisi sözlüğünde yer almasına karşı, yeni türetilcek birçok sözcük için, konuşma tanıma sisteminde model oluşturmak oldukça zor olmaktadır. Örneğin, “Osman” sözcük kökünden oluşturulan ve ayrık bir sözcük olarak ele alabilceğimiz, “Osmanlılaştıramayabileceklerimizdenmişcesine” sondan ek alarak türetilmiş kelimesini inceleyecek olursak, kelime kökü 12 ek almıştır [Tunali, V, 2005]. Türkçe gibi birleşmeye yönelik (aglutinatif) dil ailelerinde, konuşma tanıma yönelik geliştirilecek dil modelleri, tahmin edilecek sözcük sayısı arttıkça daha karmaşık bir yapıya dönüşmektedir. İngilizce dili için bu durum, Türkçe’ye oranla daha ele alınabilir. Bir dil ailesi için, sözcüklerin en küçük birimleri olan heceler, daha az sayıda ortak seslere indirgenebildiğinde, o dilin evrenselliğinden bahsedilebilmesi mümkün olmaktadır. Tabii ki, burada o dili kullananların dünya üzerindeki sosyo-politik varlığı da göz önünde tutulmalıdır. Bugün, İngilizce, dünyada konuşma tanıma sistemleri içinde üzerinde en çok çalışma yapılan bir dil olmuştur.

İngilizce, ayrıca, morfolojik ve fonetik özellikler açısından Türkçe gibi birleşmeye yönelik dil gruplarından, konuşma tanıma sistemleri için daha elverişli bir konumdadır. İngilizce’de de, sözcükler önden veya sondan bir takım ekler alarak yeni sözcükler oluşturulmaktadır, ancak bu birleşmeye yönelik dil gruplarındaki kadar yoğun değildir. İngilizce dilinde, sözcük üzerindeki eklerin konumu veya sözcüklerin cümle içindeki konumları, onları dilbilgisi olarak tasnif etmeye olanak vermektedir.

Çizelge 5.1.’de görüldüğü gibi 0-9 arasındaki sayıların tanınmasında LPC ve MFCC ses özelliklerinin çıkarılması tekniğinden yararlanılmıştır. Türkçe ayrık sözcük tanımadaki, LPC’de en iyi sonuç % 75-85 arasında değişirken, İngilizce ayrık sözcük tanımadaki başarı % 85-90 arasında olmuştur. MFCC kullanarak ayrık sözcük tanımadaki, Türkçe sözcükler için başarı % 85-90 arasında iken, İngilizce sözcükleri tanımadaki başarı % 90-95 arasındadır. Görüldüğü gibi, İngilizce sözcükleri

tanımadaki başarı yüzde olarak daha fazladır. Ortam gürültüsü azaltıldığında ise Türkçe ve İngilizce sözcük tanımadaki başarı eşitlenerek, % 90-95 arasında olduğu gözlenmiştir.

Sonuç olarak, bu tez çalışmasında, konuşmacıya bağımlı ve ayırık kelime tanıma üzerinde çalışılmış, sistem performansı bu tercihler göz önünde bulundurularak optimize edilmiştir. Türkçe, sondan eklemeli kelime türeten morfolojik dil yapısına sahip olduğu için birleşik kelime ve sürekli konuşma tanıma açısından, örneğin İngilizce'ye oranla daha karmaşık tekniklerin kullanılmasını gerektirecektir. Konuşma tanıma performansını, yapılan ses kaydı, ortam gürültüsü, konuşmacıların seslerini aynı biçimde kullanmaları direk olarak etkilediği de göz önünde bulundurulmalıdır.

Çizelge 5.1.Konuşmacıya bağımlı, ortamdaki bağımsız, Türkçe ve İngilizce rakamlar ile yapılan sözcük tanıma test sonuçları

Sözcük	LPC 1.en iyi sonuç	LPC 2.en iyi sonuç	MFCC 1.en iyi sonuç	MFCC 2.en iyi sonuç	Gürültü azaltma (LPC/MFCC) 1.en iyi sonuç	Gürültü azaltma (LPC/MFCC) 2.en iyi sonuç
Bir / One	Bir,%80 One, %85	Beş,%10 Nine, %10	Bir,%90 One, %95	Beş,%5 Nine, %5	Bir,%95 One, %95	Beş,%5 -
İk i / Two	İki,%80 Two, %85	Yedi,%15 Four, %5	İki,%85 Two, %90	Yedi,%10 Zero, %5	İki,%90 Two, %95	Yedi,%5 -
Üç / Three	Üç,%80 Three, %90	Bir,%10 Two, %5	Üç,%85 Three, %95	Beş,%10 -	Üç,%90 Three, %95	Beş,%5 -
Dört / Four	Dört,%80 Four, %85	Dokuz,%10 Five, %10	Dört,%90 Four, %90	Bir,%5 Five, %5	Dört,%95 Four, %95	Bir,%5 Five, %5
Beş / Five	Beş,%80 Five, %85	Bir,%15 Four, %10	Beş,%90 Five, %90	Bir,%90 Four, %5	Beş,%90 Five, %90	Bir,%90 Four, %5
Altı / Six	Altı,%75 Six, %85	Yedi,%15 Seven,%5	Altı,%85 Six, %95	Yedi,%10 -	Altı,%90 Six, %95	Yedi,%10 -
Yedi /Seven	Yedi,%80 Seven, %80	İki,%10 One, %10	Yedi,%90 Seven, %90	İki,%5 One, %5	Yedi,%90 Seven, %95	İki,%5 One, %5
Sekiz / Eight	Sekiz,%85 Eight, %90	Dokuz,%10 Five, %5	Sekiz,%85 Eight, %90	Beş,%10 Five, %5	Sekiz,%90 Eight, %95	Beş,%5 -
Dokuz / Nine	Dokuz,%85 Nine, %85	Sekiz,%10 One, %5	Dokuz,%90 Nine, %95	Dokuz,%5 -	Dokuz,%90 Nine, %95	Dokuz,%5 -
Sıfır / Zero	Sıfır,%75 Zero, %85	Sekiz,%10 Four, %5	Sıfır,%90 Zero, %95	Bir,%5 -	Sıfır,%95 Zero, %95	Bir,%5 -

KAYNAKLAR

1. Deller, J. R., Proakis, J.G., Hansen, John H.L., “Discrete-Time Processing of Speech Signals”, **Prentice Hall**, New Jersey, 151-211, 266-352, 601-677(1993).
2. Frank, F., William, W., “Computer Speech Processing”, **Prentice Hall**, London 81-124 (1985).
3. Kondo, A.M., “Digital Speech”, **John Wiley&Sons**, 2nd Edition, England, 6-9, 23-36, 219-229 (1995).
4. Oppenheim, A.V, Schafer, R.W., “Discrete-time Signal Processing”, **Prentice Hall**, New Jersey, 1989, 403-430, 581-610, 768-775 (1989).
5. Proakis, J.G., Ingle, V. K., “Digital Signal Processing using Matlab V4”, **PWS Publishing**, Boston, 116-170, 182-208, 401-405 (1997).
6. Kinnunen, T., Hautamäki, V., Fränti, P., "Fusion of Spectral Feature Sets for Accurate Speaker Identification", **9th International Conference Speech and Computer (SPECOM'2004)**, 361-365 (2004).
7. İkizler, N. , Çavdar, I.H., “Çapraz ilişki yöntemi kullanılarak konuşmacı bağımlı Türkçe ayırık hece tanıma sisteminin gerçekleştirilmesi”, **9. İşaret İşleme ve Uygulamaları Kurultayı, SIU2001**, 1-6 (2001).
8. Vatansever, F., Uysal, F., Uzun, A., “Ayırık Dalgacık Dönüşümü ile gürültü süzme”, **Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu, ELECO'2002**, 1-5 (2002)
9. Aida, K.R., Ardil, C., Rusmatov, S.S, “Investigation of combines use of MFCC and LPC in speech recognition systems”, **World Academy of Science, Engineering and Technology-2006**, 1-7 (2006).
10. Mihov, S. G., Ivanov, R. M., Popov, A.P., “Denoising Speech signal by Wavelet Transform”, **Annual Journal of Electronics, 2009, ISSN 1313-1842**, 1-5 (2009)

11. Ozden, N., “Thesis Speaker-dependent, isolated word recognition system”, Yüksek Lisans, *Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 10-25 (1985).
12. Tunali, V., “Thesis A speaker dependent, larger vocabulary, isolated word speech recognition system for Turkish”, Yüksek Lisans, *Marmara Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, 3-23 (2005).
13. Saxena, A., Singh, A., “A microprocessor based speech recognizer for isolated Hindi digits”, *IEEE India Student Paper Contest*, 1-5 (2002).
14. İnternet : Matlab Information Technologies Services, “Matlab Tutorial”
http://www.mathworks.com/academia/student_center/tutorials/launchpad.html
 (2009).
15. İnternet : Sharif University, ”Speech recognition using DTW”
<http://ce.sharif.edu/~veisi/index.htm> (2005).

EKLER

EK-1 Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Örnekleri ve Sonuçları

Çizelge 1.1. [Ali, Veli, Deli, Kedi] sözcükleri için tanınma DZÇ skorları

Kelime	Tipik LPC-DZÇ sonuçları	Tipik MFCC-DZÇ sonuçları
ali	ali , 3.143080 veli , 6.063239 deli , 4.796390 kedi , 4.644647	ali , 3.179638 veli , 5.181395 deli , 4.562105 kedi , 4.869956
veli	ali , 4.949646 veli , 4.540064 deli , 5.278934 kedi , 7.086650	ali , 3.540782 veli , 3.238296 deli , 4.972241 kedi , 3.456724
deli	ali , 5.128488 veli , 8.343279 deli , 4.391852 kedi , 4.841881	ali , 5.735601 veli , 7.328974 deli , 4.165311 kedi , 5.473632
kedi	ali , 4.846353 veli , 6.003684 deli , 4.705487 kedi , 4.048604	ali , 7.050003 veli , 6.029225 deli , 6.854321 kedi , 4.733900

EK-1 (Devam) Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Örnekleri ve Sonuçları

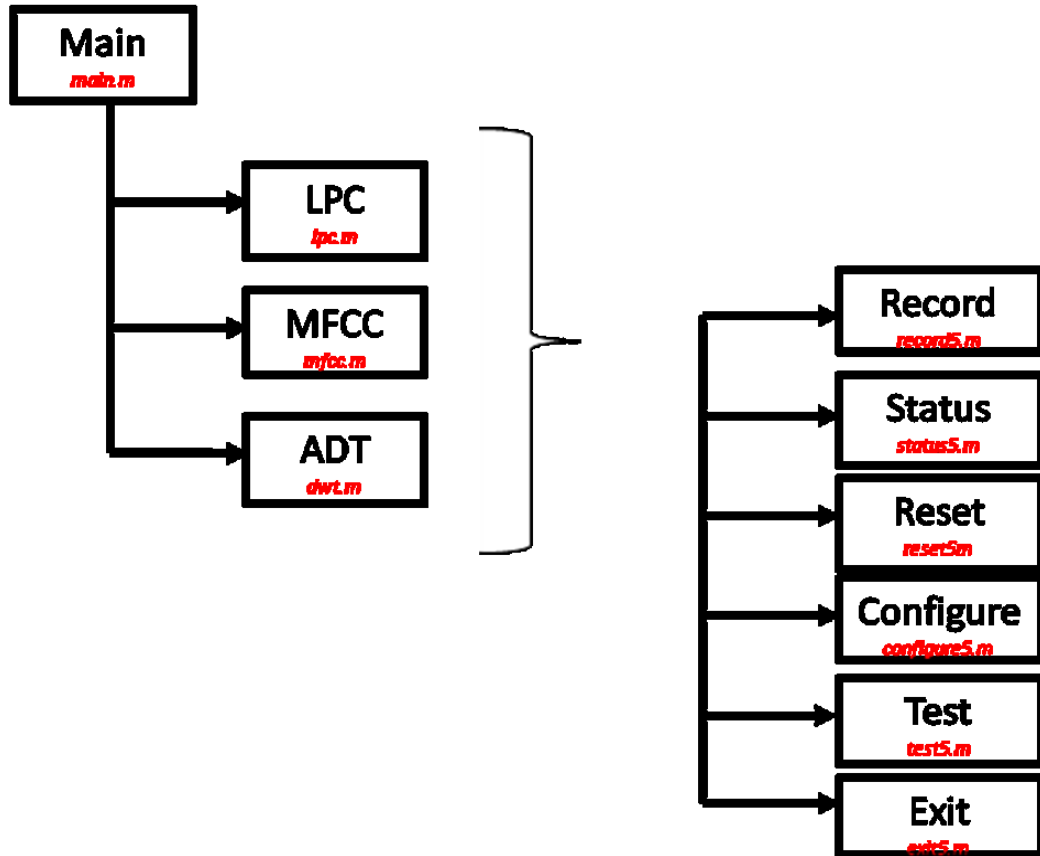
Çizelge 1.2. [Elma, Portakal, Muz, Mandalin] sözcükleri için tanınma DZÇ skorları

Kelime	1.tanınan	2.tanınan
elma	Elma (~%100)	
portakal	portakal(~%100)	
muz	muz (~%100)	
mandalin	mandalin(~%100)	

Çizelge 1.3. [Karafatma, Karabiber, Karadudu, Karavan] sözcükleri için tanınma DZÇ skorları

Kelime	Tipik LPC-DZÇ sonuçları	Tipik MFCC-DZÇ sonuçları
Karafatma (1)	(1) , 4.946689 (2) , 5.700903 (3) , 6.569671 (4) , 5.863463	(1) , 4.402874 (2) , 5.234914 (3) , 7.358733 (4) , 4.601419
Karabiber (2)	(1) , 7.352304 (2) , 6.571031 (3) , 7.314692 (4) , 6.351712	(1) , 7.521708 (2) , 6.489230 (3) , 7.211908 (4) , 6.671261
Karadudu (3)	(1) , 8.966780 (2) , 9.386133 (3) , 8.387099 (4) , 8.388749	(1) , 8.741727 (2) , 9.581930 (3) , 8.182619 (4) , 8.176216
Karavan (4)	(1) , 6.480397 (2) , 5.636893 (3) , 6.764954 (4) , 6.077249	(1) , 7.352304 (2) , 6.571031 (3) , 7.314692 (4) , 6.351712

EK-2 Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Program Algoritması

Program algoritması

EK-3 Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```

<Start.m>
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009
clc

type startmenu % Calls user friendly main menu
i_option = input('Make your choice : ');
if (i_option <= 0) | (i_option > 3)
    break
end
if i_option==1
mode='MFCC';
end
if i_option==2
mode='LPC';
end
if i_option==3
break
end

disp(' ');
m_denoise = input('with Discrete Wavelet Transformation for
denoising, Yes/No : ','s');

if ((m_denoise=='Y')||(m_denoise=='y'))
mode_denoise=1;
end

if ((m_denoise=='N')||(m_denoise=='n'))
mode_denoise=0;
end

config_done=0;

main
<Start_Menu.m>
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009

*****

% Speaker Dependent Isolated-Word Speech Recognition
% using 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

- 1) 39-parameters MFCC
- 2) 11th order LPC
- 3) Exit

```
<Main.m>
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009
clc

while 1
    menu = ['record5      '           % Calls functional files
            'status5     '
            'reset5      '
            'configure5'
            'test5       '
            'exit5       '];

    if strcmp(mode,'LPC')
        if mode_denoise==1
            type mainmenu_LPC_denoise
        else
            type mainmenu_LPC           % Calls user friendly
        end
    end
    if strcmp(mode,'MFCC')
        if mode_denoise==1
            type mainmenu_MFCC_denoise
        else
            type mainmenu_MFCC           % Calls user friendly
        end
    end
    i_option = input('Make your choice : ');
    if ((i_option <= 0) | (i_option > 5))
        break
    end
    menu = menu(i_option,:);
    eval(menu);
end
```

```
<MainMenu_LPC.m>
*****
```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

Speaker Dependent Isolated-Word Speech Recognition

Using DTW for 11th order LPC

Tayfun Ozisbakan May 2009

- 1) Record a new template in database
- 2) Status of database
- 3) Reset record database & parameters
- 4) Configure recording parameters & database
- 5) Test your word!
- 6) Exit

<MainMenu_MFCC.m>

Speaker Dependent Isolated-Word Speech Recognition

Using DTW for 39 MFCC features

Tayfun Ozisbakan May 2009

- 1) Record a new template in database
- 2) Status of database
- 3) Reset record database & parameters
- 4) Configure recording parameters & database
- 5) Test your word!
- 6) Exit

<Record5.m>

```
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009
```

```
if config_done==0
```

```
Fs=8000; %Sampling rate
```

```
Duration=2; %Recording duration in seconds
```

```
no_max_templates=10; %Default
```

```
N_bit=16;
```

```
end
```

```
fid = fopen('template_database.txt', 'rt');
```

```
no_max_templates=str2num(fgetl(fid));no_blank_templates=str2num(fgetl(fid));
```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```

fclose(fid);

if no_blank_templates==0
    fprintf('End of template recording...');
    pause(15)
    main
end
%TestWave=Record(Fs,Duration);

%fprintf('2 seconds recording (16-bit, mono, Fs=16KHz)\n');
fprintf('%d seconds recording (%d-bit, mono, Fs=%d
Hz)\n',Duration,N_bit,Fs);

FileRoot=input('Please enter a file name for the wave record:
\n','s');
FileName=strcat(FileRoot, '.wav');

fprintf('Press any key to start...\n');
pause;
fprintf('    >> Start recording ...\n');

speechIn = wavrecord(Duration*Fs, Fs); %Duration*Fs is the total
number of sample points

if mode_denoise==1                %Discrete Wavelet denoising
[speechIn,cfs] = cmdddenoise(speechIn,'sym4',5,'s');
end

wavwrite(speechIn, Fs, N_bit, FileName);
fprintf('    >> Finished recording.\n');
sound(speechIn,Fs);
%wavplay(speechIn, Fs, 'sync');', FileName);

%MFCC_initial=Feature_Extruction(speechIn,Fs);
%MFCC_final=CMS_Normalization(MFCC_initial);
%FeatureName=(strcat(FileRoot, '_MFCC'));
%v=genvarname(FeatureName);
%v=genvarname(strcat(FileRoot, '_MFCC'));
%eval([v ' = MFCC_final'])
%eval([genvarname(strcat(FileRoot, '_MFCC')) ' = MFCC_final'])
%eval([genvarname(strcat(FileRoot, '_MFCC')) ' =
CMS_Normalization(MFCC_initial)'])

if strcmp(mode, 'LPC')
eval([genvarname(strcat(FileRoot, '_LPC')) ' =
CMS_Normalization(Feature_Extruction_LPC(speechIn,Fs));'])
end

if strcmp(mode, 'MFCC')
eval([genvarname(strcat(FileRoot, '_MFCC')) ' =
CMS_Normalization(Feature_Extruction_MFCC(speechIn,Fs));'])
end

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Matlab Programı

```

fid = fopen('template_database.txt', 'rt');
no_max_templates=str2num(fgetl(fid));
no_blank_templates=str2num(fgetl(fid));

for index=1:(no_max_templates-no_blank_templates)
    template_name(1,index)={fgetl(fid)};
end
fclose(fid);

fid = fopen('template_database.txt', 'wt');
fprintf(fid, '%d \n', no_max_templates);
fprintf(fid, '%d \n', (no_blank_templates-1));

for index=1:(no_max_templates-no_blank_templates)
    fprintf(fid, '%s \n', template_name{1,index});
end
fprintf(fid, '%s \n', FileRoot);
fclose(fid);

fid = fopen('template_database.txt', 'rt');
no_max_templates=str2num(fgetl(fid));
no_blank_templates=str2num(fgetl(fid));

for index=1:(no_max_templates-no_blank_templates+1)
    template_name(1,index)={fgetl(fid)};
end
fclose(fid);

%main

<Reset5.m>
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009

config_done=0;
fid = fopen('template_database.txt', 'w');
no_max_templates=10;
no_blank_templates=10;

for index = 1:no_max_templates
    fprintf(fid, ' \n');
    template_name(1,index)={' '};
end
fclose(fid);

fid = fopen('template_database.txt', 'w');
fprintf(fid, '%d \n', no_max_templates);
fprintf(fid, '%d \n', no_blank_templates);
fclose(fid);

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```

delete('*.wav')

%disp(' >> Record database is reset. ');
cprintf('err', ' Record database is reset!\n ');
cprintf('err', ' 2 seconds recording (16-bit, mono, Fs=8000 Hz) ');

<Status5.m>
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009

disp( '-----' );
fid = fopen('template_database.txt', 'rt');

no_max_templates=str2num(fgetl(fid));
fprintf('Total number of record templates = %d\n',no_max_templates);

no_blank_templates=str2num(fgetl(fid));
fprintf('Number of blank templates = %d \n',no_blank_templates);

for index=1:(no_max_templates-no_blank_templates)
    template_name(1,index)={fgetl(fid)};
end

for index=1:(no_max_templates-no_blank_templates)
    fprintf('Record no.%d ---> %s\n',
index,template_name{1,index});
end
disp( '-----' );

fclose(fid);

%main

<Test5.m>
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009

disp(' ');
disp(' Speaker Dependent Isolated Word Speech Recognition ');
disp(' Using DTW for 39 CMS-MFCC features ');
disp(' Tayfun Ozisbakan May2009 ');
disp( '-----' );

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Matlab Programı

recognition

<Recognition.m>

```
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009
```

```
% clear all;
% close all;
% clf;
% clc;
```

record_test_template

```
disp('=====');
disp(' Start recognition by DTW using 39 CMS-MFCC features... ');
```

```
if strcmp(mode,'LPC')
Test_LPC_Features=
CMS_Normalization(Feature_Extruction_LPC(speechIn,Fs));
    if (isempty(Test_LPC_Features))
        %disp(' No audio signal has been recorded! ');
        fprintf('err', ' No audio signal has been recorded! ');
        break
    end
end
```

```
if strcmp(mode,'MFCC')
Test_MFCC_Features=
CMS_Normalization(Feature_Extruction_MFCC(speechIn,Fs));
    if (isempty(Test_MFCC_Features))
        %disp(' No audio signal has been recorded! ');
        fprintf('err', ' No audio signal has been recorded! ');
        break
    end
end
```

```
fid = fopen('template_database.txt', 'rt');
no_max_templates=str2num(fgetl(fid));
no_blank_templates=str2num(fgetl(fid));

for index=1:(no_max_templates-no_blank_templates)
    template_name(1,index)={fgetl(fid)};
end
fclose(fid);
```

```
%for i=1:No_Templates
for i=1:(no_max_templates-no_blank_templates)
```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```

    % [Template_MFCC_Features, Template_Name] = SelectNextTemplate(i);
    % Template_Name{1,i} = template_name{1,i};
    Template_Name = template_name{1,i};
    FileRoot = Template_Name;
    % FileRoot = Template_Name{1,i};

    header = strcat(FileRoot, '.wav');
    y = wavread(header);

    if strcmp(mode, 'LPC')
        eval([genvarname(strcat(FileRoot, '_LPC')) ' =
        CMS_Normalization(Feature_Extruction_LPC(y, Fs));'])
        header = strcat(FileRoot, '_LPC');
        testheader = genvarname(header);
        Template_LPC_Features = eval(testheader);
    end

    if strcmp(mode, 'MFCC')
        % eval([genvarname(strcat(FileRoot, '_MFCC')) ' =
        CMS_Normalization(Feature_Extruction_MFCC(y, Fs));'])
        % header = strcat(FileRoot, '_MFCC');
        % testheader = genvarname(header);
        Template_MFCC_Features = CMS_Normalization(Feature_Extruction_MFCC(y, F
        s));
    end

    % Construct the 'local match' scores matrix as the cosine distance
    between the featurers
        if strcmp(mode, 'LPC')
            Local_Distance =
            LocalDistance(abs(Template_LPC_Features), abs(Test_LPC_Features));
        end

        if strcmp(mode, 'MFCC')
            Local_Distance =
            LocalDistance(abs(Template_MFCC_Features), abs(Test_MFCC_Features));
        end

    % Find the lowest-cost path across Local_Distance matrix
    [Path_y, Path_x, Distance] = DTW(Local_Distance);

    % Least cost (final cost) is value in top right corner of
    Distance matrix
    Distance_from_Template(i) = Distance(1, size(Distance, 2));
    if i > 1
        if Distance_from_Template(i) < Answer_DistanceFrom
            Answer_Name = Template_Name;
            Answer_Distance = Distance;
            Answer_Path_x = Path_x;
        end
    end

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```

        Answer_Path_y=Path_y;
        Answer_DistanceFrom=Distance_from_Template(i);
    end
else
    Answer_Name=Template_Name;
    Answer_Distance=Distance;
    Answer_Path_x=Path_x;
    Answer_Path_y=Path_y;
    Answer_DistanceFrom=Distance_from_Template(i);
end

    % Plot the min cost path though Distance matrix for all
    Templates
    % colormap(1-gray);
    % subplot(2,5,i);
    % imagesc(Distance)
    % hold on; plot(Path_x,Path_y,'r'); hold off
    % str=['DTW for ',Template_Name,' ',Distance=
    ',num2str(Distance_from_Template(i))];
    %title(Template_Name);

end

    % Plot the min cost path though Distance matrix for Answer
    (Template with MIN final cost)
    % figure(2);
    % colormap('jet');
    %imagesc(Answer_Distance)
    % hold on; plot(Answer_Path_x,Answer_Path_y,'r'); hold off
    %str=['Answer is: ',Answer_Name,' ',Distance=
    ',num2str(Answer_DistanceFrom)];
    %title(str);
    %bar(Distance_from_Template)
    %title('DTW Distances');
    %xlabel('Template #');
    disp([' It's seem that answer is: <<',Answer_Name,'>>']);
    fprintf('\n');

    disp('DTW "distance-cost" results for each recorded template:');
    for index=1:(no_max_templates-no_blank_templates)
        fprintf('Record no.%d ---> %s, %f \n',
index,template_name{1,index},Distance_from_Template(index));
    end
end

```

<DTW.m>

```

% Speaker Dependent Isolated Word Speech Recognition
% Using DTW for 11th order LPC with autocorrelation method
% Tayfun Ozisbakan Oct2010

function [Path_y,Path_x,Distance] = DTW(LocalDistance)
    % [Path_y,Path_x] = DTW(LocalDistance)
    % Use dynamic programming to find a min-cost path through
    matrix LocalDistance.

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```
% Return state sequence in Path_y, Path_x
% Modified version of dp !
%

[Row, Col] = size(LocalDistance);

% costs
Distance = zeros(Row+1, Col+1);
Distance(Row+1, :) = NaN;
Distance(:, 1) = NaN;
Distance(Row+1, 1) = 0;
Distance(1:(Row), 2:(Col+1)) = LocalDistance;

AllPath = zeros(Row, Col);

for i = Row+1:-1:2;
    for j = 1:Col;
        [SelPath, tb] = min([Distance(i, j), Distance(i, j+1),
Distance(i-1, j)]);
        Distance(i-1, j+1) = Distance(i-1, j+1) + SelPath;
        AllPath(i-1, j) = tb;
    end
end

% Traceback from top left for finding Path
i = 1;
j = Col;
Path_y = i;
Path_x = j;
while i < Row & j > 1
    tb = AllPath(i, j);
    if (tb == 1)
        i = i+1;
        j = j-1;
    elseif (tb == 2)
        i = i+1;
    elseif (tb == 3)
        j = j-1;
    else
        error;
    end
    Path_y = [i, Path_y];
    Path_x = [j, Path_x];
end

Distance = Distance(1:(Row), 2:(Col+1));
```

<Feature_Extraction_LPC.m>

```
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009
```

EK-3 (Devam) Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Matlab Programı

```

% *****
% ***** Feature extraction for 11 order LPC *****
% *****
function Featurs= Feature_Extruction_LPC(InputWave,Fs)

% Return 12 LPC parameters + log energy vector of
InputWave

if nargin<1
    disp('Error: in Feature_Extruction, no wave file. ');
end
if isstr(InputWave),
    [InputWave,Fs,NBits] = wavread(InputWave);
elseif nargin==1
    Fs=8000;
end

% ===== Set Parameters.
% Frame size: N(ms),Overlapping region is M(ms)
% Generally , M = (1/2)*N , which N = 24.
FrameSize_ms = 24; % Ex. N=32 = (256/8000)*1000 , each
frame has 256 points.
Overlap_ms = (1/2)*FrameSize_ms;
FrameSize = round(FrameSize_ms*Fs/1000); % 256
Overlap = round(Overlap_ms*Fs/1000); % 86
%No_of_Frames= floor((size(InputWave,1)/(FrameSize-Overlap)) -
1)+2;

Threshold = 0.0001; % for energy test ==> remove fromes with
energy bellow this amount.

% ===== Step 1: Pre-emphasis.
InputWave = filter([1, -0.95], 1, InputWave);

% ===== Step 2: Windowng & overlapping.
Frame = buffer(InputWave, FrameSize, Overlap);

normalize_coff = 10;
energy = sum(Frame.^2)/FrameSize;
index = find(energy < Threshold);
energy(index) = [];
logEnergy = 10*log10(energy)/normalize_coff;
Frame(:, index) = []; % Remove empty frames
Featurs = [];
for i = 1:size(Frame, 2); % size(Frame, 2)=No_of_Frames

% ===== Step 3: Hamming window.
WindowedFrame = hamming(FrameSize).*Frame(:,i);
%WindowedFrame = hanning(FrameSize).*Frame(:,i);
%WindowedFrame = blackman(FrameSize).*Frame(:,i);

% ===== Step 4: 11th order autocorrelation LPC
abc = lpc(WindowedFrame.*Frame(:,i),11);
fprintf('FrameSize is %g .\n', abc);

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```

end;
Featur = [Featur; logEnergy];
<Feature_Extraction_MFCC.m>
% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009

%
*****
%           *****      Feature extraction for 39 MFCC
*****
%
*****
function Featur = Feature_Extraction_MFCC(InputWave, Fs)

% Featur = Feature_Extraction(InputWave);
% Return 26 MFCC feature vectors of InputWave

if nargin < 1
    disp('Error: in Feature_Extraction, no wave file. ');
end
if isstr(InputWave),
    [InputWave, Fs, NBits] = wavread(InputWave);
elseif nargin == 1
    Fs = 8000;
end

% ===== Set Parameters.
%           Frame size: N(ms), Overlapping region is M(ms)
%           Generally , M = (1/2)*N , which N = 24.
FrameSize_ms = 24;           % Ex. N=32 = (256/8000)*1000 , each
frame has 256 points.
Overlap_ms = (1/2)*FrameSize_ms;
FrameSize = round(FrameSize_ms*Fs/1000); % 256
Overlap = round(Overlap_ms*Fs/1000); % 86
%No_of_Frames = floor((size(InputWave,1)/(FrameSize-Overlap)) -
1)+2;
% Triangular BandFilter parameters :
StartFreq, CenterFreq, StopFreq. (20 Bank filters)
StartFreq = [1  3  5  7  9  11  13  15  17  19  23  27  31
35  40  46  55  61  70  81]; %Start
CenterFreq = [3  5  7  9  11  13  15  17  19  21  27  31  35
40  46  55  61  70  81  93]; %Center
StopFreq = [5  7  9  11  13  15  17  19  21  23  31  35  40
46  55  61  70  81  93  108]; %End

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```
%      StartFreq =[1      100 200 300 400 500 600 700 800  900  989
1136 1305 1499 1722 1977 2279 2609 2998 3444 3956 4544 5220 6001];
%Start
%      CenterFreq=[100   200 300 400 500 600 700 800 900  1000 1149
1320 1516 1741 2000 2297 2639 3031 3482 4000 4595 5278 6063 6964];
%Center
%      StopFreq  =[200   300 400 500 600 700 800 900 1000 1124 1309
1504 1727 1983 2278 2617 2999 3453 3966 4556 5234 6012 6906 7927];
%End
      Threshold = 0.0001; % for energy test ==> remove fromes with
energy bellow this amount.

% ===== Step 1: Pre-emphasis.
InputWave = filter([1, -0.95], 1, InputWave);

% ===== Step 2: Windowng & overlapping.
Frame = buffer(InputWave, FrameSize, Overlap);

normalize_coff = 10;
energy = sum(Frame.^2)/FrameSize;
index = find(energy < Threshold);
energy(index) = [];
logEnergy = 10*log10(energy)/normalize_coff;
Frame(:, index) = []; % Remove empty frames
Featurrs = [];
for i = 1:size(Frame, 2); % size(Frame, 2)=No_of_Frames

    % ===== Step 3: Hamming window.
    WindowedFrame = hamming(FrameSize).*Frame(:,i);
    %WindowedFrame = hanning(FrameSize).*Frame(:,i);
    %WindowedFrame = blackman(FrameSize).*Frame(:,i);

    % ===== Step 4: FFT: fast fourier transform.
    %      Using FFT function to calculate.
    %      Compute square of real part and imaginary
part.
    FFT_Frame = abs(fft(WindowedFrame));

    % ===== Step 5: Triangular bandpass filter.
    %      Using user defined function
triBandFilter(fftFrame{i}).
    No_of_FilterBanks = 20; %No_of_FilterBanks means
counts of log spectral magnitude.
    tbfCoef =
TriBandFilter(FFT_Frame,No_of_FilterBanks,StartFreq,CenterFreq,StopF
req);

    % ===== Step 6: Logearithm.
    tbfCoef = log(tbfCoef.^2);

    % ===== Step 7: DCT: Discrete Cosine Transform.
    %      Using DCT to get L order mel-scale-cepstrum
parameters.
```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```

No_of_Features = 12;          % generally No_of_Features is 12.
Cepstrums =
Mel_Cepstrum2(No_of_Features,No_of_FilterBanks,tbfCoef);

    Features = [Features Cepstrums'];
end;
Features = [Features; logEnergy];

%=====compute delta energy and delta cepstrum=====
%Calculate delta cepstrum and delta log energy
% get 13 order Features.
Delta_window = 2;
D_Features = DeltaFeature(Delta_window, Features);

%=====compute delta-delta energy and delta
cepstrum=====
%Calculate delta-delta cepstrum and delta log energy
%Combine them with previous features, get 39 order Features.

%Delta_window = 2;
%D_d_Features = Delta_DeltaFeature(Delta_window, Features);
% or
D_d_Features = DeltaFeature(Delta_window, D_Features);

%===== Combine cepstrum,delta and delta-delta
Features = [Features ; D_Features ; D_d_Features]; % 39 features
%Features = [Features ; D_Features]; % 26 features

%===== Sub function
=====
%=====

%          *****
%          *****   Triangular Band Filter   *****
%          *****

function tbfCoef =
TriBandFilter(fftFrame,P,StartFreq,CenterFreq,StopFreq)
    %The function is triangular bandpass filter.
    for i = 1 : P,
        % Compute the slope of left side of triangular bandpass
filter
        for j = StartFreq(i) : CenterFreq(i),
            filtmag(j) = (j-StartFreq(i))/(CenterFreq(i)-
StartFreq(i));
        end;
        % Compute the slope of right side of triangular bandpass
filter
        for j = CenterFreq(i)+1: StopFreq(i),
            filtmag(j) = 1-(j-CenterFreq(i))/(StopFreq(i)-
CenterFreq(i));
        end;
    end;
end;

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Matlab Programı

```

        tbfCoef(i) =
sum(fftFrame(StartFreq(i):StopFreq(i)).*filtmag(StartFreq(i):StopFreq(i))');
    end;

% *****
% *****      Mel-scale cepstrums      *****
% *****

function Cepstrum = Mel_Cepstrum2(L,P,tbfCoef)
%compute mel-scale cepstrum , L should be 12 at most part.
for i=1:L,
    coef = cos((pi/P)*i*(linspace(1,P,P)-0.5))';
    Cepstrum(i) = sum(coef.*tbfCoef');
end;

% *****
% *****      Delta cepstrums      *****
% *****

function D_Features = DeltaFeature(delta_window,Features)
% Compute delta cepstrum and delta log energy.
rows = size(Features,1);
cols = size(Features,2);
temp = [zeros(rows,delta_window) Features
zeros(rows,delta_window)];
D_Features = zeros(rows,cols);
denominator = sum([1:delta_window].^2)*2;
for i = 1+delta_window : cols+delta_window,
    subtrahend = 0;
    minuend = 0;
    for j = 1 : delta_window,
        subtrahend = subtrahend + temp(:,i+j)*j;
        minuend = minuend + temp(:,i-j)*(-j);
    end;
    D_Features(:,i-delta_window) = (subtrahend -
minuend)/denominator;
end;
%Features = [Features ; temp2];

% *****
% *****      Delta-Delta cepstrums      *****
% *****

function D_d_Features = Delta_DeltaFeature(delta_window,Features)
% Compute delta delta cepstrum and delta log energy.

% another way!
% Features1 = DeltaFeature(delta_window,Features);
% Features2 = DeltaFeature(delta_window,Features1);
% Features = [Features ; Features2];

rows = size(Features,1);

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Matlab Programı

```
cols = size(Featur,2);
temp1 = [zeros(rows,delta_window) Featur
zeros(rows,delta_window)];
temp2 = [zeros(rows,delta_window) Featur zeros(rows,delta_window)];
```

EK-2 (Devam)

```
D_d_Featur = zeros(rows,cols);

% Rabiner method
denominator = sum([1:delta_window].^2)*2;
denominator2 =
delta_window*(delta_window+1)*(2*delta_window+1)*(3*delta_window^2+3
*delta_window-1)/15;
for i = 1+delta_window : cols+delta_window,
    subtrahend = 0;
    minuend = 0;
    subtrahend2 = 0;
    minuend2 = 0;
    for j = 1 : delta_window,
        subtrahend = subtrahend + temp1(:,i+j);
        minuend = minuend + temp1(:,i-j);
        subtrahend2 = subtrahend2 + j*j*temp2(:,i+j);
        minuend2 = minuend2 + (-j)*(-j)*temp2(:,i-j);
    end;
    temp1(:,i) = subtrahend + minuend + temp1(:,i);
    temp2(:,i) = subtrahend2 + minuend2;
    D_d_Featur(:,i-delta_window) = 2*(denominator.*temp1(:,i)-
(2*delta_window+1).*temp2(:,i))/(denominator*denominator-
(2*delta_window+1)*denominator2);
end;
% Featur = [Featur ; temp3];
```

<LocalDistance.m>

```
% Speaker Dependent Isolated Word Speech Recognition
% Using DTW for 11th order LPC with autocorrelation method
% Tayfun Ozisbakan Oct2010

function Out2 = LocalDistance(A,B)
% Out = LocalDistance(A,B)
% calculate the local distance between feature matrices A and
B.
% Using inner product i.e. cos(angle between vectors) between
vectors.
% A and B have same #rows.
%

Mag_A = sqrt(sum(A.^2));
Mag_B = sqrt(sum(B.^2));

Cols_A = size(A,2);
```

EK-3 (Devam) Konuşmacıya Bağımlı Ayırık Sözcük Tanıma Matlab Programı

```

Cols_B = size(B,2);
Out = zeros(Cols_A, Cols_B);
for i = 1:Cols_A
    for j = 1:Cols_B
        % normalized inner product i.e. cos(angle between vectors)
        Out(i,j) = (A(:,i)'*B(:,j))/(Mag_A(i)*Mag_B(j));
    end
end

%Out = (A'*B)./(Mag_A'*Mag_B);

Row=size(Out,1);
for i=1:fix(Row/2)
    %tmp=M(Row-i+1,:);
    Out2(Row-i+1,:)=Out(i,:);
    Out2(i,:)=Out(Row-i+1,:);%tmp;
end
if mod(Row,2)~=0
    Out2(fix(Row/2+1),:)=Out(fix(Row/2+1),:);
end

% Use 1-Out2 because DTW will find the *lowest* total cost
Out2=1-Out2;

```

<CMS_Normalization.m>

```

% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009

function Out=CMS_Normalization(Featur)
    % Cepstral Mean Subtraction (CMS) of Features

    [N,M]=size(Featur);

    for i=1:N
        Mean(i)=sum(Featur(i,:))/M;
        Out(i,:)=Featur(i,:)-Mean(i);
    End

```

<Configure5.m>

```

% Speaker Dependent Isolated-Word Speech Recognition
% using Dynamic Time Warping for
% 39-parameters MFCC or 11th order LPC
% with Discrete Wavelet Transformation for denoising
% Tayfun Ozisbakan May2009

clear, clc

```

EK-3 (Devam) Konuşmacıya Bağımlı Ayrık Sözcük Tanıma Matlab Programı

```

    Fs = input('Enter the sampling frequency (Hz) for recording :
');
    N_bit = input('Enter the sampling data length, bit : ');
    Duration = input('Enter the recoding time, sec : ');
    no_max_templates = input('Enter the total number of record
templates : ');
    windowing_time = input('Enter the windowing time, msec : ');
    windowing_overlap_time = input('Enter the windowing overlap
time, msec : ');
    type window_filter_menu
    windowing_filter = input('Enter the window filter selection,
e.g. 1, 2 or 3 : ');
    config_done=1;
%main
Start
<Window_filter_menu.m>

```

```

Window filter selection:
*****
    1) Hamming window
    2) Hanning window
    3) Blackman window
*****

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ÖZİŞBAKAN, Tayfun
 Uyuğu : T.C.
 Doğum tarihi ve yeri : 27.11.1971 İzmir
 Medeni hali : Evli
 Telefon : 0 (532) 242 51 89
 e-mail : ozisbakan@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Orta Doğu Teknik Üniversitesi Elektrik ve Elektronik Müh.	1993
Lise	İzmir Karşıyaka Lisesi	1988

İş Deneyimi

Yıl	Yer	Görev
1994-1999	Arçelik A.Ş.	AR-GE Müh.
1999-2002	Profilo Holding	AR-GE Müh.
2002-2009	STMicroelectronics	Teknik Pazarlama Müh.

Yabancı Dil

İngilizce, Fransızca

Yayınlar

- Proposed Format of Turkish Occupational Standard, “Endüstriyel Bakım-Onarım Elektronikçisi”, 1998

Hobiler

Basketbol, yüzme, yelkencilik, bilgisayar teknolojileri