Bir Türkçe Fonem Kümeleme Sistemi Tasarımı ve Gerçekleştirimi

The Design and Implementation of A Turkish Speech Phoneme Clustering System

Harun ARTUNER

Hacettepe Üniversitesi

Fen Bilimleri Enstitüsü Yönetmenliğinin

Bilgisayar Bilimleri-Mühendisliği Bölümü için Öngördüğü

DOKTORA TEZİ

olarak hazırlanmıştır.

1994

Fen Bilimleri Enstitüsü Müdürlüğüne,

Bu çalışma tarafımızdan BİLGİSAYAR BİLİMLERİ - MÜHENDİSLİĞİ BÖLÜMÜ'nde DOKTORA TEZİ olarak kabul edilmiştir.

Başkan : Prof. Dr. Ünal YARIMAĞAN

Üye : Prof. Dr. Neş'e YALABIK

Üye : Prof. Dr. Selçuk GEÇİM

Üye : Prof. Dr. Emin AKATA

Üye : Prof. Dr. Ali SAATÇİ

ONAY

Bu tez ..../..../1994 tarihinde Enstitü Yönetim Kurulunca belirtilen yukarıdaki jüri üyeleri tarafından kabul edilmiştir.

..../..../1994

Prof. Dr. Gültekin GÜNAY

FEN BİLİMLERİ ENSTİTÜ MÜDÜRÜ

ÖZET

Günümüzde bilgisayar ve insan arasındaki etkileşim, el ve gözü birlikte kullanmayı gerektiren, daha çok *yazıya dayalı* biçimde gerçekleşmektedir. Halbuki insanlar kendi aralarında, daha çok *sese dayalı* iletişim yolunu kullanmaktadırlar. İnsan-bilgisayar arasında, yazının yanı sıra, sese dayalı etkileşimin kurulabilmesi kullanım kolaylığı, doğallığı ve konforu açısından önem taşımaktadır. Bunun yapılabilmesi sesli ifadelerin hızlı ve hatasız çözümlenebilmesini (tanınabilmesini) gerektirir. Sesin bilgisayar-insan arası etkileşim bağlamında yerini alması, sesli ifadelerin ya da konuşmanın tanınmasında ortaya çıkan sorunların çözülebilmesine bağlıdır. Sesli ifadeleri tanımadaki zorluklar bunun, daha çok gerçek zamanlı (hızlı) yapılması gereğinden kaynaklanmaktadır. Sesli ifade tanıma süreci, konuşulan dilin yapısına bağlı olarak da farklılıklar göstermektedir.

Sesli ifade tanımada, genelde iki temel yaklaşım kullanılmaktadır. Bunlar sözcük tanıma ve fonem tanıma yaklaşımlarıdır. Türkçe sözcükler, fiil çekimi, ismin *i* ve *e* halleri gibi nedenlerle sonekler almaları dolayısıyla, kimi zaman batı dillerinde bir ya da birkaç cümleye karşı gelen karmaşıklıkta olabilmektedir. Bu nedenle sınırlı ve genel amaçlı bir sözlük oluşturma olanağı bulunamadığından Türkçe genel bir sesli ifade tanıma kapsamında sözcük tanıma yaklaşımının kullanılamayacağı düşünülmüştür. Bu tez kapsamında, sözcük altı ses birimlerine dayalı sürekli sesli ifade tanıma yaklaşımı benimsenmiştir. Yaratılan deneysel Türkçe korpüs ses birimlerine ve tek konuşmacıya dönük olmuştur.

Bu bağlamda, ilk olarak, tanınması amaçlanan ses birimlerine ilişkin, temsil niteliği yüksek ve uygun özellik vektörlerinin belirleme çalışmaları yürütülmüştür. Daha sonra, tanıma sürecinde kullanılacak ve her fonemi en iyi temsil ettiği varsayılan özellik vektörlerinden, Türkçe *Codebook* oluşturulmuştur. Türkçe Alfabe *fonemik* bir alfabe özelliği taşımaktadır. Bu sayede Türkçe her foneme bir yazı simgesi (harf) atanabilmektedir. Bu tez kapsamında Türkçe sesli ifade *phon*'larından fonemlere, geçiş *(phon to phoneme mapping)* çalışmaları yürütülerek sözcük tanıma ve tanınan sözcüğün yazıya geçirilmesi kapsamında dolaysız yararlanılacak fonem kümeleme amaçlanmıştır. Bu bağlamda Türkçe fonemler için fonotopik dağılımları, *Self Organizing Feature Map* yöntemi kullanılarak elde edilmiştir.

Anahtar Sözcükler: Türkçe sesli ifade tanıma, Konuşmacıya bağımlı Türkçe fonem tanıma, Türkçe özellik vektörü çıkarımı, Türkçe *Codebook,* Nöron Ağları

ABSTRACT

Units such as keyboard, mouse, CRT's, printers and plotters constitute, today the basic tools of communication between man and machine. These units necessitate mainly the use of texts as means of interaction. As far as the ease and comfort of computers usage is concerned, the idea of speech based communication becomes an important issue in the frame work of man-computer interaction which requires speech recognition. The real time aspect (or the speech) of that recognition poses the hardest problem of the subject, so it requires experimentation with new techniques based on new approches.

In speech recognition two main approaches prevail, one based on word recognition and the other on phoneme recognition. As Turkish words may take on suffixes they may show complexities which render them equivalent to as much as several sentences in English. Consequently, the word recognition approach cannot be used within the scope of Turkish speech recognition because a general-purpose dictionary with clear-cut boundaries cannot be built. Within the scope of this thesis, an approach based on the recognition of sub-lexical sound units was adopted. The experimental Turkish word corpus created within this context was accordingly based on sound units and speaker-dependent.

In this thesis, studies were first conducted to determine feature vectors that are highly representative of and suitable for the sound units to be recognised. Then, a Turkish *Codebook* was generated using the feature vectors, assumed as best representatives per phoneme, to be used in the recognition process. The Turkish alphabet is a phonemic alphabet. For this reason, a written sign (letter) can be assigned to each Turkish phoneme. Within the context of Turkish phonem recognition and the mapping of those phone into phonemes in order to pass from spoken to written words a general phoneme clustering is realized. In the framework of that clustering phonotopic maps are obtained for each of the 28 Turkish phonems using the Self Organizing Feature map method.

Keywords: Turkish speech recognition, Speaker-dependent Turkish phoneme recognition, Turkish feature vector extraction, Turkish Codebook, Neural Network.

TEŞEKKÜR

Bu çalışmanın yapıldığı ortamı sağlayan ve bu çalışmanın her aşamasında görüşlerinden yaralandığım, katkı ve yardımlarını gördüğüm tez yönetmenim Prof. Dr. Ali SAATÇİ'ye teşekkür ederim.

Türkçenin fonetik yapısına ilişkin konularda bilgisine başvurduğum Ankara Üniversitesi Dil Tarih ve Coğrafya Fakültesi öğretim üyesi Doç. Dr. İclal ERGENÇ ve çalışma süresince aynı çaılşma ortamını paylaştığım meslektaşım Ferhat Y. SAVCI'ya ve bölümdeki tüm çalışanlara değerli katkılarından dolayı teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

SİMGELER ve KISALTMALAR DİZİNİ

ANN	*Artificial Neural Network.*
ASR	*Automatic Speech Recognition.*
ASSP	*Acoustics Speech and Signal Processing*
AVIOS	*American Voice I/O Society*
CELP	*Code-book Excited Linear Prediction.*
COLING	*COmputational LINGuistics*
DFT	*Discrite Fourier Transformation*
DTW	*Dynamic time warping*
FFT	*Fast Fourier Transformation*
HMM	*Hidden Markov Model*
IEEE	*Institute of Electrical and Electronics Engineers*
IPA	*International Phonotic Alphabet*
JASA	*Journal of the Acoustic Society of America*
LPC	*Linear Predictive Coding*
LVQ	*Learned Vector Quantisation*
NLP	*Natural Language Processing*
NN	*Neural Network.*
TI	*Texas Instruments*
TIMIT	*TI* ve *MIT* de ortak hazırlanan çok büyük sesli ifade veri tabanı
TTS	*Text-To-Speech*

TÜBİTAK	Türkiye Bilimsel ve Teknik Araştırma Kurumu.
VQ	*Vector Quantisation.*

ÇİZİMLER DİZİNİ

Çizim **Sayfa**

ÇİZELGELER DİZİNİ

Çizelge **Sayfa**

1. GİRİŞ

Günümüzde bilgisayar ve insan arasındaki etkileşim araçları klavye, *mouse* gibi giriş, ekran yazıcı gibi çıkış birimlerinden oluşmaktadır. Bu birimler daha çok *yazıya dayalı*, el ve gözü kullanmayı gerektiren birimlerdir. Halbuki insanlar kendi aralarında, daha çok *sese dayalı* iletişim yolunu kullanmaktadırlar. İnsan-bilgisayar arasında, yazının yanı sıra sese dayalı etkileşimin kurulabilmesi kullanım kolaylığı, doğallığı ve konforu açısından önem taşımaktadır. Bunun yapılabilmesi sesli ifadelerin hızlı ve hatasız çözümlenebilmesini (tanınabilmesini) gerektirir. Yazının kolay çözümlenir özelliği sesli ifadelerde bulunmamaktadır. Başka bir deyişle, yazı değişmez kodlarla tanımlı damgalardan (karakterlerden) oluşurken sesli ifadeler, kişilerden kişilere değişiklik gösteren, tanınması ve dolayısıyla kodlanması zor fonemlerden oluşmaktadır. Sesin bilgisayar-insan arası etkileşim bağlamında yerini alması, sesli ifadelerin ya da konuşmanın tanınmasında ortaya çıkan sorunların çözülebilmesine bağlıdır. Sesli ifadeleri tanımadaki zorluklar bunun, daha çok gerçek zamanlı (hızlı) yapılması gereğinden kaynaklanmaktadır. Çoğu kez klasik örüntü tanıma yöntemleri sesli ifadelerin çevrim-dışı (*off-line*) tanınmasında kullanılabilmektedir. Ancak çevrim-içi (*on-line*) ve gerçek zamanlı uygulamalarda yüksek başarımın elde edilebilmesi, klasik örüntü tanıma yöntemlerinin dışında yeni yaklaşımların araştırılmasını ve kullanılmasını gerektirmektedir.

Günümüz sesli ifade tanıma araştırmalarının içinde nöron ağlarının kullanımı oldukça önemli bir yer tutmaktadır. Sesli ifade tanımada nöron ağları, sesli ifade tanıma sistemlerinin çeşitli modüllerinde klasik yaklaşımların yerine kullanılabilmektedir. Bu tez kapsamında Türkçe fonem altı birimlerin nöron ağı yaklaşımı kullanılarak kümelenmesi ve fonemleri temsil etme başarımı çalışması gerçekleştirilmiştir.

Sesli ifade tanıma, genel anlamda iki değişik biçimde düşünülmektedir. İlk yaklaşımda sesli ifadeler ayrışık sözcükler (*isolated words*) olarak ele alınmakta ve sesli ifadelere ilişkin özellik vektörleri, bu nedenle sözcük tabanında tutulmaktadır. Bu yaklaşım özellik vektörlerinden oluşan sözlükleri kullanmayı gerektirmektedir. Sesli ifade tanımada ikinci yaklaşım sürekli sesli ifadeleri (*connected/continiuous speech*) tanıma yaklaşımıdır. Bu yaklaşımda tanıma birimi sözcük olmak yerine, daha çok, (hece, fonem, *phon* gibi) sözcük altı ses birimleri olmaktadır. Türkçe sözcükler, fiil çekimi, ismin *i* ve *e* halleri gibi nedenlerle sonekler almaları

dolayısıyla, kimi zaman batı dillerinde bir ya da birkaç cümleye karşı gelen karmaşıklıkta olabilmektedir. Bu nedenle bu tez çalışması kapsamında sözcük altı ses birimlerine, başka bir deyişle fonemlere dayalı, sürekli sesli ifade tanıma yaklaşımına temel oluşturacak fonem kümeleme çalışması tek konuşmacı için benimsenmiştir.

Bunun için, ilk aşamada sözcük altı ses birimlerini en iyi temsil edecek özellik vektörünün belirlenmesi çalışmaları yürütülmüştür. Bu bağlamda *Fast Fourier Transformation (FFT)* ve *Linear Prediction Coding (LPC)* tekniklerine dayalı 7 değişik özellik vektörü çıkarma yöntemi incelenmiştir. Bu yöntemlerden tanımaya yönelik en yüksek başarımı sağlayan yöntem kullanılarak Türkçe *Codebook* oluşturulması yoluna gidilmiştir. Türkçe *Codebook* etiketlenmiş, başka bir anlatımla başlangıcı ve sonu belirlenmiş fonemlere dayalı olarak kurulmuştur.

Türkçe Alfabe *fonemik* bir alfabe özelliği taşımaktadır. Bu sayede Türkçe her foneme bir yazı simgesi (harf) atanabilmektedir. Bu tez kapsamında Türkçe sesli ifade *phon*'larından fonemlere, geçiş *(phon to phoneme mapping)* çalışmaları fonem kümeleme yöntemlerinin araştırılması biçiminde yürütülmüş, sözcük tanımaya temel oluşturacak altyapının oluşturulması amaçlanmıştır. Ancak bu biçimde gerçekleştirilen sözcük tanıma ve başarım ölçümleri ve başarım yükseltme araştırmaları tez kapsamı içinde ele alınmamıştır.

Bu tez çalışmaları kapsamında söz konusu edilen Türkçe fonem kümeleme sistemi tasarım ve gerçekleştirim çalışmaları TÜBİTAK tarafından desteklenen EEEAG-DPT/14 projesi içinde ele alınmıştır.

Tez ilk bölüm olan Giriş ile birlikte yedi bölümden oluşmaktadır. Sesli ifade tanıyıcıların tasarımında genelde çıkış noktası bunların doğadaki karşılıkları olmuştur. Bu nedenle, ikinci bölümde, sesin üretilmesi, algılanma süreci ve bu süreçle ilgili genel tanımlara yer verilmiştir. Sesli ifade tanıma sürecinin önemli bir kesimini sesli ifadelerin elektriksel sinyallere dönüştürülmesi ve klasik sinyal işleme yöntemleri ile işlenmesi oluşturmaktadır. Sesli ifadelerin tanımaya hazırlanmasında kullanılan teknikler ve aşamaları üçüncü bölümde açıklanmıştır. Dördüncü bölümde, şu ana kadar gerçekleştirilmiş ve başarımları yayınlanmış sesli ifade tanıyıcıları ve bunların ortak özellikleri özetlenmiştir. Beşinci bölümde yaygın olarak kullanılan belli başlı sesli ifade tanıma teknikleri anlatılmıştır. Tez çalışmaları sırasında

açıklanan bu tekniklerin herbiri, ya bizzat gerçekleştirilerek ya da uluslararası yazılım kütüphanelerinden elde edilerek ayrı ayrı denenmiştir.

Türkiye Türkçesinin fonetik özellikleri altıncı bölümde açıklanmıştır. Yine bu bölümde ikinci bölümde ele alınan genel tanımlarım Türkçe'ye uyarlanmış biçimleri verilmiştir. Yedinci bölümde, tez çalışmalarının yürütüldüğü TÜBİTAK projesi ile oluşturulan Türkçe Sesli İfade Tanıma Labaratuvarı tanıtılmıştır. Labaratuvarda oluşturulan birikim ve kaynaklar özetlenmiştir.

Türkçe fonem kümeleme sistemininin tasarımında iki ana kesim bulunmaktadır. Bunlardan ilki sesli ifadelerin temsil nitelikleri yüksek vektörlerle gösterilmesidir. *Vector Quantization* olarak bilinen bu işlem sayesinde, sesli ifadeye ilişkin çok sayıdaki sayısal veri kısıtlı sayıda vektör birleşenine dönüştürülmektedir. *Vector Quantization* adlı işlem 6'ıncı bölümde incelenen ve seçilen *mel* skalasında *FFT*'ye dayalı *cepstrum* hesaplama yöntemiyle gerçekleştirilmektedir. Elde edilen vektörlerin sesli ifadenin tanınmasında kullanılması amaçlanmaktadır. Bu bağlamda tanımaya temel oluşturacak iki aşamalı bir çalışma izlenmiştir. Bu aşamalardan ilki *alıştırma*, diğeri ise *tanıma* aşamasıdır. Alıştırma aşamasında Türkçe *Codebook* üretilmesi ve etiketlenmesi, tanıma aşamasında ise sözkonusu *codebook* kullanılarak fonemlere karşı gelen yazılı simgelerin (harflerin) belirlenmesi gerçekleştirilmektedir.

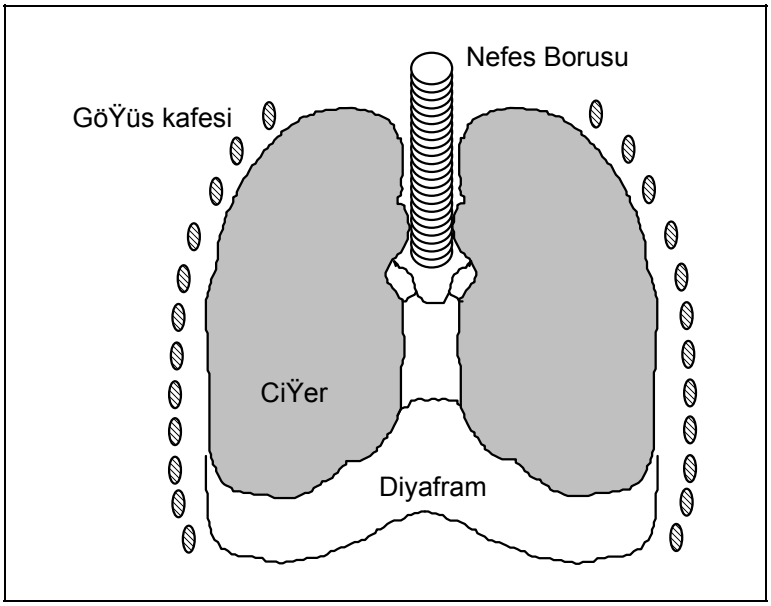
2. SESLİ İFADENİN ÜRETİMİ ve ALGILANMASI

2.1. Sesli İfadenin Üretimi

Sesli ifadeyi üreten ses organları üç kesimde toplanmaktadır. Bunlar:

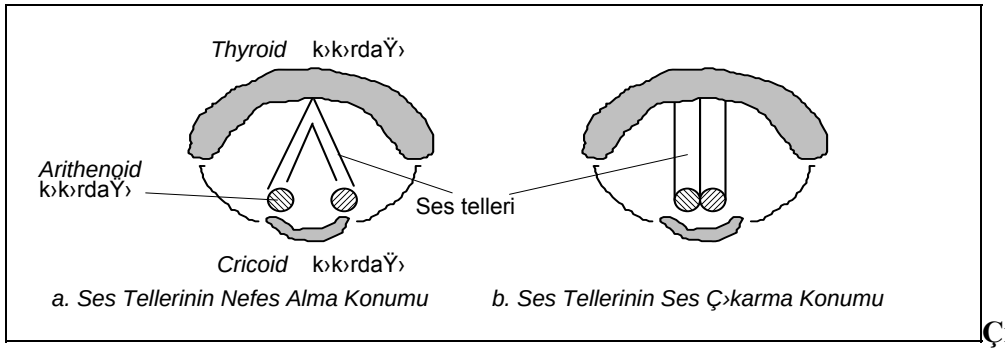
- Akciğerler ve nefes borusu (*lungs and trachea*),
- Ses tellerinin yer aldığı kesim (*larynx*) ve
- Gırtlaktan dudaklara uzanan kesim, ses yoludur (*vocal tract*).

Akciğerler 4-5 litre hava içerebilen süngerimsi bir yapıya sahip olup ses sisteminin güç kaynağıdır. Ciğerler *pleura* denilen bir zar torbanın içinde yer alır. Bu torba kenarlardan omurgalara alt kesimden de diyaframa bağlıdır. Ciğerlerin hava soluyabilmesi, omurgalar ve diyaframın, bunlara bağlı kaslarla hareket ettirilmesi sonucu *pleura* torbasının genişleyip daralması ile mümkün olur. *Pleura* torbasının göğüs kafesi yoluyla genişleyip daralması *thoracic*, diyafram yoluyla genişleyip daralması ise *abdominal* (karından) nefes alma olarak bilinir. Nefes borusu ciğerleri bronşlar yoluyla gırtlağa bağlayan ve kıkırdak halkalardan oluşan bir yapıdır.



Çizim 2.12. Ciğerler ve Nefes Borusunun konumu

Larynx ses tellerinin (vocal cords) bulunduğu kesimdir. Ses tellerinin *larynx* içindeki konumu Çizim 2.2'de gösterilmiştir. Buna göre ses telleri *thyroid* kıkırdağı ile *aritenoid* kıkırdaklarına bağlıdır. *Aritenoid* kıkırdakları da *cricoid* kıkırdağına bağlıdır. Bu yapıda *aritenoid* kıkırdakları açılıp kapanarak ses tellerinin nefes borusunu açıp kapaması sağlanmaktadır. Ses tellerinin tam açık olduğu durum nefes alıp verme durumudur. Ses tellerinin nefes borusunu tam kapadığı durum ise yutma durumudur. Bu iki durumda ses tellerinin titreşmesi sözkonusu olmaz. Havanın ses telleri arasından belli bir süreklilik ve şiddette dışarı itilmesi bu tellerin titreşmesine olanak verir. Bu titreşim sesli ifadede *ünlülere* kaynaklık eden titreşimdir.

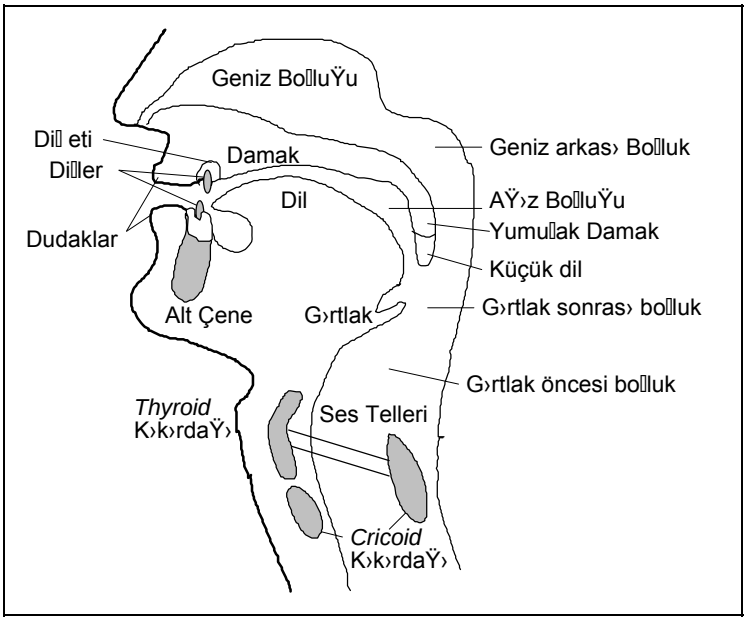


Çizi

m 2.13. *Larynx*'in Görünümü

Gırtlaktan dudaklara uzanan kesim (*vocal tract*):

- Gırtlak öncesi yutak (*laryngeal pharynx*)
- Gırtlak sonrası yutak (*oral pharynx*)
- Geniz arkası yutak (*nasal pharynx*)
- Ağız boşluğu (*Oral cavity*)
- Geniz boşluğu (*Nasal cavity*) kesimlerinden oluşur.



Çizim 2.14. Sesli İfadeyi üreten Ses Organları

Bu kesimlerin yanı sıra sesli ifade üretim sürecine katkı verebilen diğer ögeler:

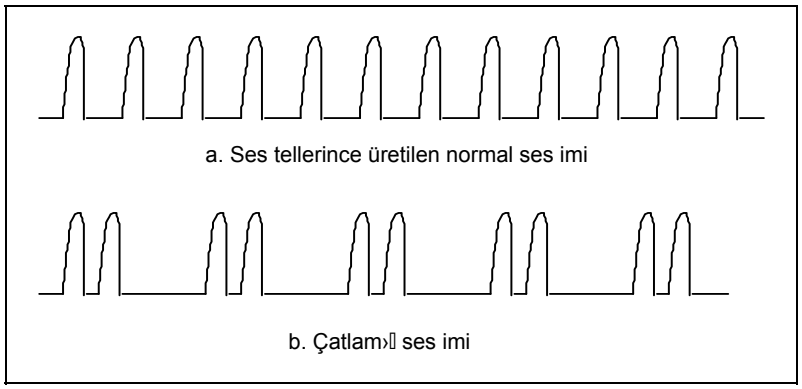
- Gırtlak,
- Alt çene,
- Dil,
- Yumuşak Damak ve Küçük dil
- (Sert) Damak
- Dişler
- Dişetleri ve
- Dudaklardır.

Bunlardan dilde; dilin ucu, ön dil orta dil, arka dil ve dilin kök kesimleri bulunur.

Sesli ifade üretim süreci, titreşimlerin oluşturulması ve bunların modülasyonunu içerir. Titreşimlerin oluşturulması sesin tahriki olarak da bilinir. Sesli ifade üretim sürecinde, sesin tahriki çeşitli biçimlerde gerçekleşir. Bu bağlamda:

- Ses tellerinin *titreştirilmesi (phonation),*
- Ses telleri arasından havanın *fısıltı (whispering)* yaratacak biçimde geçirilmesi,
- Ses tellerinden dudaklara değin herhangi bir konumda, havanın türbülans yaratacak biçimde *sürtüşmesinin (frication)* sağlanması,
- Ağız içi boşluğun hava ile doldurulup basınç yaratılması, sonra ağzın birden açılarak *patlama (explosion)* sağlanması
- Dil ucu, küçük dil, gırtlak gibi ögelerin titreştirilmesi *(vibration)*,

yoluyla sesin tahriki sağlanır.



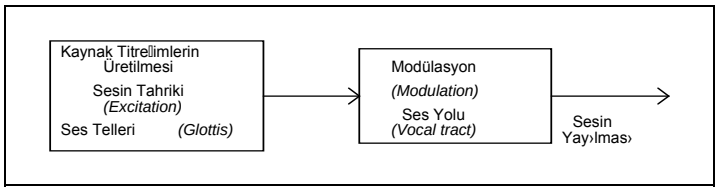
Çizim 2.15. Ötümlü seslere kaynaklık eden ses titreşimleri

Yukarıda açıklanan tahrik yollarından *ses tellerinin titreşimi* ile elde edilen sesler *ötümlü*(*voiced*), diğer yollarla elde edilenlerin tümü ise *ötümsüz(voiceless/unvoiced)* sesler olarak bilinir. Ötümlü seslere kaynaklık eden ses titreşimlerinin belirli bir sıklığı bulunmaktadır. Ötümsüz seslere ise, havanın sürtünme ya da patlamasının yarattığı türbülansa dayalı, yüksek sıklığa sahip *gürültü* türü titreşimler kaynaklık etmektedir. Ötümlü seslere kaynaklık eden ses titreşimlerinin görünümü Çizim 2.4'te verilmiştir.

Ses, ses tellerinden dudaklara uzanan kesimde, kaynak titreşimlerin modülasyonuyla üretilmektedir. *Modülasyon, fizyolojik ve akustik* olmak üzere iki biçimde olabilmektedir. Başta dil olmak üzere, titreşimlerin ses organlarınca duraklatılması, ya da bunlara yüksek sıklıkta (geniş bantlı) gürültü türü ek titreşimlerin bindirilmesi, eklenmesi fizyolojik modülasyon olarak bilinmektedir. Akustik modülasyonda ise

harmonik yönünden çok zengin olan kaynak titreşimlerin, rezonans kutusunda olduğu gibi filtrelenmesi, *formant* olarak adlandırılan temel rezonans sıklıklarının elde edilmesi sağlanmaktadır.

Sesli ifadelerdeki sesler *ünlüler* ve *ünsüzler* olarak da sınıflandırılır. Tüm *ünlü* sesler *(vovels)* ile kimi *ünsüz* sesler *(consonants)* ötümlü olup akustik modülasyon yoluyla oluşturulmaktadır. Akustik modülasyon sonucu oluşan formant sıklıkları, bu sesleri belirlemede önemli bir kimlik bilgisi niteliği taşımaktadır. Duraklama, geniş bantlı ek titreşim bindirme gibi fizyolojik modülasyon ise çoğu ünsüz sesin üretilmesinde etkili olmaktadır (Çizim 2.5).

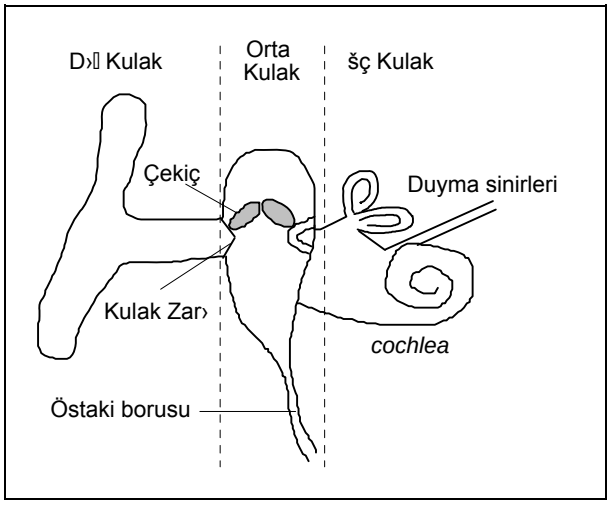


Çizim 2.16. Sesin üretilme süreci

2.2. Sesli İfadenin Duyulması ve Algılanması

Sesli ifadeler, duyma organı olan kulak ile duyulmakta ve beyin tarafından algılanmaktadır. Kulak; dış, orta ve iç kulak olmak üzere üç bölümden oluşmaktadır. Dış kulak dış ortam ile kulak zarı arasında kalan kesimdir. Yaklaşık 2.7cm uzunluğunda 0.7cm çapında borumsu bir bağlantı ile kulak zarına ulaşılır. Bu boyutları ile 3 kHz'lik rezonans sıklığına sahip olup bu değer çevresinde yer alan sıklık bandındaki sesleri, diğerlerine göre daha çok yükseltmektedir. Ses titreşimleri kulak zarı tarafından orta kulağa iletilmektedir.

Orta kulak, ses titreşimlerinin iç kulağa aktarılmasını sağlayan *malleus* (çekiç), *inars* ve *stapes* adlı kemikleri içermektedir. Bunlardan çekiç adlısı zara yapışıktır. Bu kemikler kulak zarı ile orta kulağı iç kulağa bağlayan oval pencere arasında bağlantıyı sağlayarak titreşimlerin enerji yitirmeksizin iç kulağa geçmesini sağlarlar. Bu bağlamda empedans uyumlama ve genlik sınırlama işlevlerinden söz etmek mümkündür.



Çizim 2.17. İnsan Kulağının Kesit Görünümü

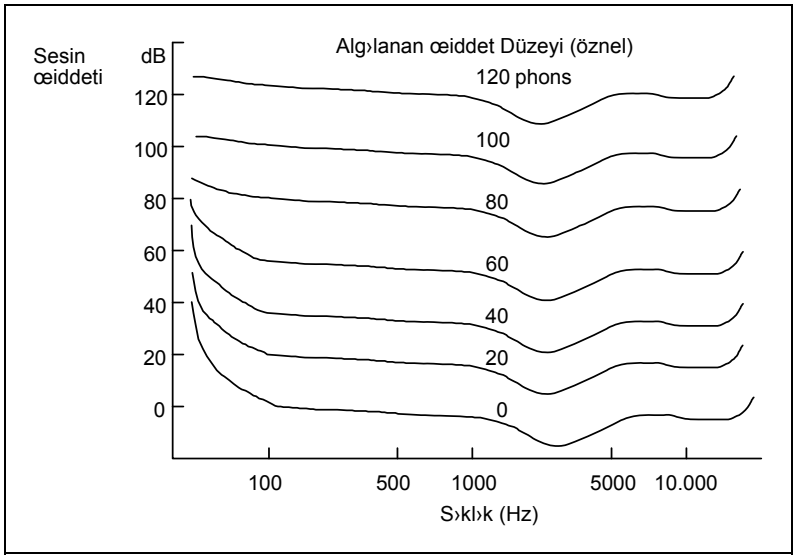
Sıvı ile dolu olan iç kulakta, *cochlea* adlı yapı içinde ses algılama hücreleri ve bunların bağlandığı sinir uçları bulunmaktadır. *Cochlea* adlı yapıya ses titreşimleri oval pencereden girmekte ve yuvarlak pencereden çıkmaktadır. U yapılı bir boruya benzetilebilen *cochlea* adlı yapının her noktasında ayrı bir ses sıklığı algılanmaktadır (Bekezy 1960). Bu durumda *cochlea* adlı yapının bir spektrum çözümleyicisi gibi çalıştığı söylenebilmektedir.

İnsan kulağı, ortalama 16 Hz ile 16 kHz arasında yer alan sıklıktaki sesleri duyabilmektedir. Üst sınır, yaşın ilerlemesi ile düşmektedir. Gençlerde bu sınır 20 kHz'e çıkabilirken yaşlılarda 10 kHz'e kadar düşebilmektedir. Referans düzey 10 mikroW/cm^2 olarak kabul edildiğinde duyulan sesin şiddeti 0 ile 130 dB arasında değişmektedir.

Sesin Şiddeti *(loudness)*

Sesin *şiddeti* hem sıklık hem de genliğin bir işlevi olarak algılanmaktadır. Bu, öznel *(subjective)* bir değer olarak *algılanan şiddet* biçiminde tanımlanmaktadır. Sesin algılanan şiddeti *algılanan şiddet* (*loudness*) ya da *algılanan şiddet düzeyi* (*loudness level*) olarak iki biçimde ölçülmektedir. *Algılanan şiddet,* kişilere duydukları sesin

şiddetini, referans bir sese göre, 1/2'si, 2 katı; 1/10'u, 10 katı gibi sınıflandırmaları istenerek ölçülebilmektedir. *Algılanan şiddet* birimi *sone'*dir. *Algılanan şiddet düzeyi,* algılanan şiddet ölçümü içinde sıklık bağımlılığı ortadan kaldırıldığında elde edilen değere verilen addır. *Algılanan şiddet düzeyi* birimi *phons'*dur. Bu değerler arası ilişki N=0.063x10$^{0.003L}$ formülü ile kurulmaktadır. Burada N algılanan şiddeti, L ise algılanan şiddet düzeyini göstermektedir. İnsanlar sesin şiddetini sıklığa bağımlı olarak algılamaktadır. Bu olgu *eşit algılanan şiddet düzeyi* eğrileri ile Çizim 2.7'de gösterilmiştir. Bu çizimden, 3-4 kHz civarında, dış kulağın fiziksel yapısı gereği, duyarlılığın arttığı görülmektedir.



Çizim 2.18. Eşit Algılanan Şiddet Düzeyi Eğrileri (Fletcher and Munson, 1933)

Sesin Perdesi *(pitch)*

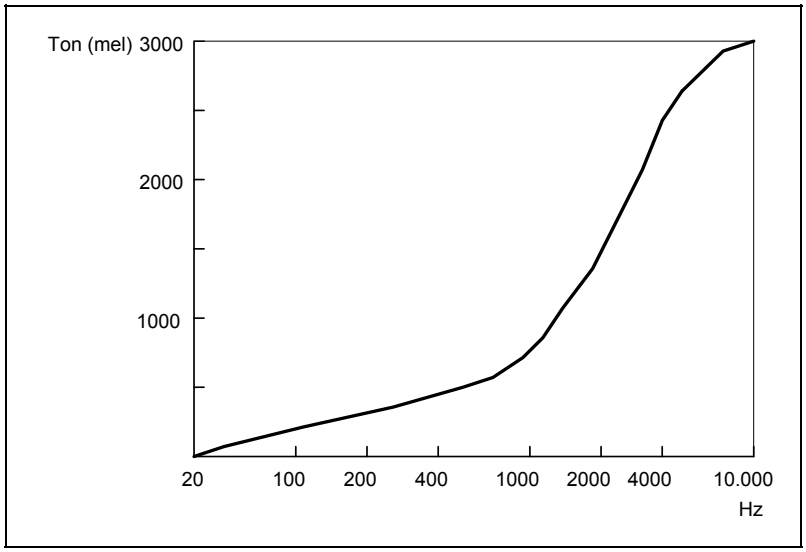
Ses tellerinin titreşmesiyle oluşan vurular dizisinin sıklığı sesin perdesi olarak bilinir(Çizim 2.4). Perde müzikte ve fizyolojide ayrı ayrı tanımlanmaktadır. Müzikte iki ses perdesi arasındaki ayrım, seslerin sıklık oranının 2 tabanlı logaritmasının 1200'le çarpılması sonucu bulunmaktadır ($p_1 - p_2 = 1200\ Log_2\ [f_1 / f_2]$). Fizyolojide kullanılan perde ise

$$p = \left[\frac{1000}{\log_{10} 2}\right] \log_{10}\left[1 + \frac{f}{1000}\right]$$ olarak hesaplanmaktadır.

Ses perdesinin ölçüm birimi *mel* olarak bilinmektedir. Mel Skalası olarak bilinen eğri Çizim 2.8'de verilmiştir.

Sesin temel harmoniği

Sesi oluşturan periyodik titreşimler, *Fourrier* açılımında olduğu üzere, sıklıkları birbirlerinin katı olan sinüssel harmoniklerin (titreşimlerin) toplamı olarak düşünülebilir. İnsan kulağı, herbiri ayrı bir perdeye karşı gelen bu harmonikleri ayrı ayrı duymaz. Ses, temel harmoniğe karşı gelen perdede algılanır. Ancak sesi oluşturan harmonik sayısının çokluğu, özellikle yüksek sıklıklara karşı gelen harmoniklerin varlığı *sesin rengi* olarak anılan niteliği belirler. Bu bağlamda insan, ünlü sesleri, içerdikleri harmoniklere göre ayrıştırmaktadır. Sesin içinden temel sıklık filtrelense bile beyin geriye kalan harmoniklerden filtrelenen bu harmoniği yeniden yaratabilmektedir. Örneğin erkek sesinin perdesi yaklaşık 120Hz'e karşı gelmektedir. Telefon sistemi 300Hz'in altındaki sıklıkları süzmesine rağmen insanlar telefonda erkek sesinin perdesini (120Hz'i) algılamayı sürdürmektedirler.



Çizim 2.19. Mel Skalası, Perde - Sıklık İlişkisi

Sesli İfadelerin Algılanması

İnsan beyninin sesli ifadeleri nasıl algıladığı ve anladığına ilişkin bilgiler tam ve tartışmasız değildir. Ancak yaygın olarak benimsenen ve bilgisayarlı sesli ifade tanımaya ışık tutan noktalar aşağıda sıralanmıştır (Thomas Parsons, 1986):

İnsan beyni sesli ifadelerle diğer sesler arasında tam bir ayrım yapmaktadır. İnsan beyni sesli ifadeleri diğer seslere göre ayrı bir biçimde işlemektedir. Var olduğu sanılan bir *ön işlem* kesimi ile sesli ifadeleri diğer seslerden *kategorik* olarak ayırmaktadır. Bu ön işleme kesimi, sonradan sesli ifadelerin içerdiği temel

harmonikleri diğer harmoniklerden de yararlanarak çıkarmakta ve sesli ifadeyi temsil eden bir dizi özelliği *ana işlem* kesimine aktarmaktadır. Ana işlem kesimi, giriş verileri olarak gelen bu özellikleri, dilbilgisi, konuşmacıyla ilgili bilgiler, konuya ilişkin bilgiler ışığında anlamlara dönüştürmektedir.

İnsan beyninde sesli ifadelerin algılanmasının nasıl gerçekleştiği tam olarak bilinmemektedir. Ancak bu konuda *Cole ve Jakimik* tarafından öne sürülen temel ilkeler, özellikle sesli ifade tanıma sistemlerinde kullanılan yaklaşımlara taban oluşturmaktadır. Bu ilkeler şunlardır:

- Sesli ifadeler, salt akustik sinyallerin çözümlenmesi ile gerçekleşmemektedir. Çözümlenen akustik sinyaller sesbilim *(phonetics)*, dilbilim *(linguistics)* ve kullanıcının genel bilgisinin yarattığı kısıtlamalarla *(constraints)* denetlenmektedir. Özetle sesli ifade tanıma, akustik sinyallerden elde edilen *verilere dayalı ancak güdümlü* olarak gerçekleşmektedir.

- Sesli ifadeler sözcük sözcük ve sözcüklerin geliş sırasında işlenmektedir. Bir sözcük tanındıktan sonra diğer sözcüğe ilişkin akustik sinyalin başlangıcı belirlenmektedir. Sözcüklerin geliş sırasında işlenmesi, bir sözcük tanındıktan sonra gelecek sözcüğün anlamı üzerinde öngörü yapma olanağı vermektedir.

- Sözcükler *ses birimleri* tabanında tanınmakta ve ses birimleri geliş sırasında tanınmaktadır. Tanınan her ses birim gelecek ses birimini daha kısıtlı bir alt kümede arama ya da öngörebilme olanağını vermektedir.

2.3. Sesli İfadelerde Ses, Fonem (Sesbirim) ve Hece (Seslem)

Fonetik bilimi, tüm dünya dillerindeki sesli ifadeleri tam, doğru ve tutarlı bir biçimde tanımlamaya yarayacak sesleri *(phon)* belirler. Dillerdeki seslerin doğal özelliklerini inceler. Bu bağlamda örneğin [k] damgası ile simgelenen ses, dilin damağa değdiği yere (örn. kim), dudağın aldığı biçime (örn. küt) göre birbirine yakın ancak değişik seslere karşı gelir. Fonetik yönünden bu seslerin herbiri diğerinden ayrı düşünülür ve bunların herbiri için özel bir simge vardır. Bu simgeler, ilk kez 1880 yılında Paul Passy tarafından yazılan *(IPA International Phonotic Alphabet)* Uluslararası Fonetik Alfabesinde yer almaktadır. Fonetik bilimi bir yandan seslerin anatomik olarak nasıl

çıkarıldığını *(Articulatory phonetics)* diğer yandan da ses sinyallerinin gözlemlenebilir ve ölçülebilir özelliklerini *(Acoustic phonetics)* inceler.

Fonetik biliminin yanı sıra, sesli ifadeleri fonemler ya da sesbirimler yönünden inceleyen *fonoloji (phonemics)* vardır. Fonem, bir dilde, bir sözcüğü diğer bir sözcükten anlamsal olarak ayırmaya yarayan en küçük ses birimine verilen addır. Eğer bir sesi *(phon)* değiştirmek ilgili sözcüğün anlamını değiştiriyorsa bu ses fonem olarak anılmaktadır. Ancak bir sesi değiştirmek ilgili sözcüğün söylenişini değiştirip anlamını değiştirmiyorsa fonemden söz edilmemektedir. Ses ve fonem arasında her zaman birebir bir ilişki bulunmamaktadır. Fonemler *anlam ayırıcı* ses topluluğu içinde yer almaktadırlar. Bir dilde ses sayısı, fonem sayısından daha çok olmaktadır. Bu bağlamda, örneğin Türkçe'de /l/ fonemi ve ince [l], kalın [l] gibi değişik sesler vardır. Kural olarak fonemler / / işaretleri, ses simgeleri ise [ ] işaretleri arasında yazılarak ayrıştırılır. Fonem anlamaya dayalı okunması itibariyla dillerarası bir birim değildir. Her dilin kendine özgü fonemleri bulunur.

Ses

Sesleri *ünlü(vovel)* ve *ünsüzler(consonants)* olmak üzere iki gruba ayırmak genel bir alışkanlıktır. Bu terimler fonetik biliminde de kullanılır ancak tam bir tanımlarını verebilmek genelde mümkün değildir. Ünlüler ötümlü ve ses tellerinden dudaklara uzanan kesimde, hava akımına bağlı olarak çıkarılan seslerdir. Ünlü sesler, ses telleriyle elde edilen titreşimlerin ses yolunda rezonansa uğratılması sonucu elde edilen sesler olarak da tanımlanabilir. Ünsüzler ise ses yolunun herhangi bir konumunda, akan havaya yaratılan engellerle (tıkamalarla) üretilen seslerdir. Ünsüzlerde ses teli titreşimleri (ötüm) önemli değildir. Bunları tanımlayan özellik daha çok seste duraklama ve türbülansa dayalı yüksek sıklıktır.

Ünsüzler

Ünsüzleri belirleyen özellikler genelde üç kritere bağlanır. Bunlar:

- çıkış noktası
- çıkış biçimi ve
- ses teli titreşimi varlığı kriterleridir.

Çıkış Noktası ünsüzün üretiminde kısılan, akan havayı engelleyen ya da kısıtlayan noktayı tanımlamaktadır. Tüm dillerde, ünsüz seslerin üretimi:

- dudaklar arası (ya da çift dudak) *(bilabial)*
- alt dudak - üst dişler arası *(labiodental)*
- dil ucu - diş ardı arası *(apicodental)*
- dil ucu - diş eti arası *(apicogingival)*
- dil ucu - diş kökü arası *(apicoalveolar)*
- dil ucu - (sert) damak arası *(apicodomal)*
- ön dil - diş kökü arası *(laminoalveolar)*
- ön dil - (sert) damak arası *(laminodomal)*
- orta dil - (sert) damak sonu arası *(centrodomal)*
- arka dil - yumuşak damak arası *(dorsovelar)*
- dil kökü - yutak arası *(pharyngial)*
- ses telleri arası *(glottal)*

kısılma noktalarında gerçekleşebilmektedir.

Çıkış Biçimi, çıkış noktasında hava akımına kısılma ya da engelleme yaratıldıktan sonra serbest bırakılış biçimini tanımlamaktadır. Buna göre, tüm dillerdeki ünsüzler:

- patlamalı *(plosive)* /p/ gibi
- üflemeli *(aspirated, stop)*
- sızmalı *(fricative, spirants, sibilants)* /s/ gibi
- sızıcılaşmalı *(affricative)*
- yan ünsüz *(lateral)* /l/ gibi
- yarı ünlü *(semivowel)* /y/ gibi

- genizden *(nasal)* /m/ gibi
- çarpmalı (açılıp kapanan noktalarda titremeli) *(trill)* /r/ gibi

olabilmektedir.

Ses teli titreşiminin varlığı ünsüzleri diğer ünsüzlerden ayırmada yararlanılan bir özellik olabilmektedir. Ses yolunda akan havayı, bir çıkış noktasında durdurma ya da engellemeye rağmen ses tellerinin titreştirilmesi, kısa bir süre sürdürülebilir. Bu tür ses teli titreşimlerini taşıyan ünsüzler *ötümlü ünsüzler* olarak anılmaktadır.

Bu durumda herhangi bir ünsüzü, *çıkış noktası, çıkış biçimi* ve *ötüm* olmak üzere üç temel özelliği ile tanımlama olanağı bulunur. Bu bağlamda, örneğin ünsüzler:

- {ötümlü, dudaklar arası, üflemeli} *(voiced, bilabial, stop)* (Örn. [b])
- {ötümsüz, dil ucu diş ardı arası, sızıcılaşmalı} *(unvoiced, apicodental, affricate)* (Örn. [p])

biçiminde tanımlanabilmektedir.

Ünlüler

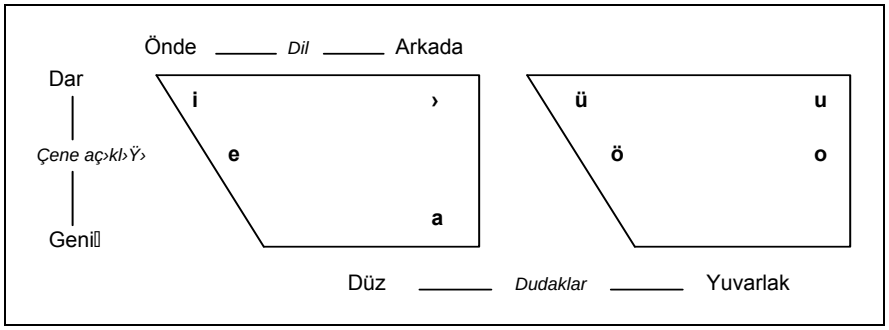
Bilindiği gibi ünlüler ses telleriyle oluşturulan titreşimlerin ses yolunda rezonansa sokulması yoluyla elde edilen seslerdir. Ünsüzlerde yapılana benzer biçimde, ünlüler, ses yolunun sokulduğu biçime, başka bir deyişle; çene açıklığına (dil damak arası açıklık) *dilin ağız içindeki konumuna* (dil önde arkada), *dudakların biçimine* ve *genzin rezonans kutusuna katılıp katılmamasına* göre sınıflandırılmaktadır. Bu bağlamda ünlüler 4 değişik parametre ile tanımlanmaktadır. Bunlar:

1. Çene açıklığı *dar (tongue high), geniş (tongue low)* (kimi durumlarda *orta* açıklıktan da söz edilebilmektedir.)
2. Dil *önde (tongue front), arkada (tongue back)*
3. Dudaklar *yuvarlak (lips rounded), düz* (yuvarlak değil) *(lips unrounded)*
4. Geniz *açık (nasalized), kapalı (unnasalized)* (salt Fransızca ve Portekizce'de anlamlı bir parametreyi oluşturmaktadır.)

parametreleridir.

Bu parametrelerden, dilin konumu göz önüne alınarak ünlü dörtgenleri *(vowel diagrams)* oluşturulmaktadır. Bu dörtgenler iki boyutlu olup yatay eksen dilin arkada ve öndeki konumunu, dikey eksen ise, çene açıklığının dar ve geniş olmasını göstermektedir. Bu dörtgenlere, dudakların biçimine ilişkin boyut, düz ve yuvarlak biçimlerinin herbiri için, iki ayrı ünlü dörtgeni oluşturularak katılmaktadır. Bunun gibi, geniz parametresinin önemli olduğu dillerde, bu dörtgenlerin, bir de geniz açıklığına göre ele alınması gerekmektedir. Örnek bir Türkiye Türkçesi Ünlü Dörtgeni Çizim 2.9'da verilmiştir. Ünlü dörtgenleri ünlü seslerin ancak kaba bir sınıflandırmasına olanak verebilmektedir.

Ünlülerle ilgili önemli bir kavram *diftong (diphtong)* kavramıdır. Bir ünlü çıkarılırken ses yolunun bir biçimden diğer biçime geçmesiyle aynı hece içinde birden çok ünlünün yer alması sağlanır. Elde edilen bu *bileşik ünlü* ses diftong olarak tanımlanır. İngilizce'de *long, who* sözcüklerindeki ünlüler diftongdur. Ancak İngilizce'de diftongların fonem (anlam belirleme) özelliği yoktur. Türkçe'de ise diftong bulunmamaktadır. Türkçe'ye yabancı dillerden geçen diftonglu sözcükler de, genellikle diftongsuz seslendirilmektedir.



Çizim 2.20. Örnek Türkiye Türkçesi Ünlü Dörtgeni (Demircan, 1979)

Fonem (Sesbirim)

Daha önce de belirtildiği gibi, fonem, bir dilde bir sözcüğün anlamını diğer bir sözcüğün anlamından ayırmaya yarayan en küçük ses birimine verilen addır. Eğer bir sesi *(phon)* değiştirmek ilgili sözcüğün anlamını değiştiriyorsa bu ses fonem olarak anılmaktadır. Ancak bir sesi değiştirmek ilgili sözcüğün söylenişini değiştirip anlamını değiştirmiyorsa fonemden söz edilmemektedir. Fonemlerin belirlenmesi için *en küçük çift*'lerden *(minimal pair)* yararlanılmaktadır. *Gel* ve *Kel* de olduğu

gibi, salt tek bir sesi değişik olan sözcükler en küçük çift olarak anılmaktadır. En küçük çift içinde değişiklik gösteren ses, anlam değişikliğine neden oluyorsa sözkonusu sesin fonem olduğu söylenmektedir. *Gel* ve *Kel* örneğinde görüldüğü üzere, /g/ ve /k/ Türkçe'de iki ayrı fonemi oluşturmaktadır. Türkçe, yazıldığı gibi okunan ya da okunduğu gibi yazılan *fonemik* bir dil olarak kabul edilirse 28-29 fonemi olduğu söylenebilmektedir. (yumuşak G üzerinde tartışmalar sürmektedir.)

Ses *(phon)* taban alınarak bir tanım vermek gerektiğinde fonem, (anlam ayırıcı özelliği bulunmayan) benzer seslerden oluşan bir *ses kümesi* olarak tanımlanabilir. Bu durumda kümenin herbir ögesi *allafon (allophone)* olarak adlandırılmaktadır. Fonem allafonlar kümesine verilen addır. Özellikle Türkçe için fonemin yazıya allafonların da telaffuza taban oluşturacağını söylemek yanlış olmaz. Bu gözleme paralel olarak, bir dili ana dili olarak konuşanların allafonlarla, yabancı dil olarak konuşanların ise fonemlerle konuştuklarını söylemek de mümkündür.

Gerek seslerin, gerekse fonemlerin diğer ses ve fonemlerden bağımsız olarak belirlenen özellikleri, hece ve sözcükler içinde, ardarda söylenmeleri nedeniyle değişebilmektedir. Başka bir deyişle, bir fonem için belirlenen özellik, sesli ifade içinde, olduğu gibi bulunamamakta ve diğer fonemlerle örtüştüğünden değişime uğramaktadır. Bu, *coarticulation* olarak bilinen, birlikte seslendirmeden kaynaklanmaktadır. *Birlikte seslendirme (coarticulation)*, foneme dayalı sesli ifade tanımada gözetilmesi gereken önemli bir hususu oluşturacaktır.

Ayırıcı Özellikler

Fonemleri diğerlerinden ayırıp biricik olarak ve tutarlı bir biçimde tanımlayabilmek için *ayırıcı özellikler* kuramı geliştirilmiştir (Jakobson, Fant, Halle, 1951). Bu kurama göre her fonem, 12 değişik akustik öznitelik *(attribute)* üzerinden ikili olarak kodlanmaktadır. Her fonem için, bu özniteliklerin varlığı / yokluğuna göre, 0 ve 1'lerden oluşan, 12 bit uzunluğunda bir kod tutulmaktadır. Sözkonusu öznitelikler şunlardır:

1. formantların varlığı / yokluğu
2. ünlü / ünsüz
3. *compact / diffuse* (spektral enerji ile ilgili)

4. gergin *(tense)* (geniş bantlı, uzun süreli) / gevşek *(lax)*
5. ötümlü / ötümsüz
6. genizden / ağızdan
7. soluğun ağızdan çıkışı sürekli / süreksiz
8. sızmalı *(strident) / mellow*
9. patlamalı *(cheched) / unchecked*
10. bas / tiz *(grave / acute)*
11. dar / geniş *(flat / plain)*
12. keskin / geniş *(sharp / plain)*

Jakobson tarafından tanımlanan bu öznitelikler daha sonra Chomsky ve Halle (1968) tarafından yeniden ele alınmış ve daha geniş sayıda öznitelik tanımlanmıştır. Ancak bu çalışmaların sonunda, tüm dillerdeki özellikleri kodlamaya yarayacak tartışmasız bir öznitelik takımı bulunamamıştır.

Hece (Seslem)

Seslem ya da hece bir *göğüs atışı* olarak tanımlanır. Göğüs atışı kaburgalar arası kasların kasılması ve gevşemesiyle oluşur. Her göğüs atışıyla ses telleri titrer. Göğüs atışına ses tellerinin titreşimi eşlik edince ünlüler çıkar. Atış sırasında kasların gevşemesine, ciğerlerden akan havayı kısıtlayan ya da durduran ünsüz vuruşu eşlik edebilir. Bunun sonucunda ses tellerinin titreşimi sürerek ya da kesilerek ünsüzler çıkarılır. Her hece mutlaka bir ünlü ses içerir. Ünlü, ünsüz ses ya da seslerle birleşerek heceyi oluşturur. Hece içinde ünlüler temel, ünsüzler ise yedek ve ikincil devinimlerdir. Ü ünlüyü, Z de ünsüzü gösterdiğinde Türkçe hecenin yapısı:

Ü (a), ZÜ (ba), ÜZ (av), ÜZZ (art), ZÜZ (nak), ZÜZZ (kart) biçiminde olmaktadır. Yabancı dillerden giren sözcükler de sözkonusu edildiğinde ZÜZZZ (tekst), ZZÜ (ski), ZZÜZ (tren), ZZÜZZ (flört), ZZZÜZ (stres) hece yapıları da kullanılmaktadır.

Akustik olarak ünlüler, ünsüzlere göre daha yüksek enerji taşımaktadır. Bu nedenle kaydedilmiş ses sinyallerinde yüksek genliklere karşı gelen kesimler ünlüleri temsil

etmektedir. Heceler hep bir ünlü içerdiklerinden ses sinyallari içinde, yerel enerji maksimumları heceleri belirlemede kullanılabilmektedir. Sesli ifade tanımada kesimleme algoritmaları bu nedenle genlik maksimumuna dayalı ele alınmaktadır. Burada önemli bir sorun, her hecenin mutlaka bir genlik maksimumuna karşı gelmesi ancak her genlik maksimumunun mutlaka bir heceyi göstermemesi olgusudur.

Kavşak *(Juncture)*

Sözcükler arası sessizlik, duraklar kavşak olarak adlandırılır. İngilizce *night rate* ile *nitrate* sözcüklerinin söylenişi aynı olmakla birlikte ikisi arasında ayrım yapılabilmesi için, seslendirilirken kavşak olarak anılan bir seslendirme biriminin kullanıldığı iddia edilmektedir. Bu seslendirme birimini heceler arasında düşünmek de mümkündür. Heceyle ilgili göğüs atışının başı ve sonu, her zaman ünsüz vuruşları ile çakışmamaktadır. Dinleyen bir kişi için hece sonlarını her zaman doğru algılayamamak bu nedenle doğaldır. Türkçe'de kavşak seslendirme birimi kullanılmamaktadır. Bunun yerine, *kaldır aç* ile *kaldıraç* arasındaki ayrımı vurgulamaya yarayan *durak* kavramı vardır. Durak yerinin seçimi doğrudan bağlama bağlı olarak yapılmaktadır.

Bürün *(prosodic)*

Daha önce de belirtildiği gibi fonemler, Türkçe sesbirim olarak da bilinir. Sesbirim kavramı genişletilerek:

parça sesbirimler ve

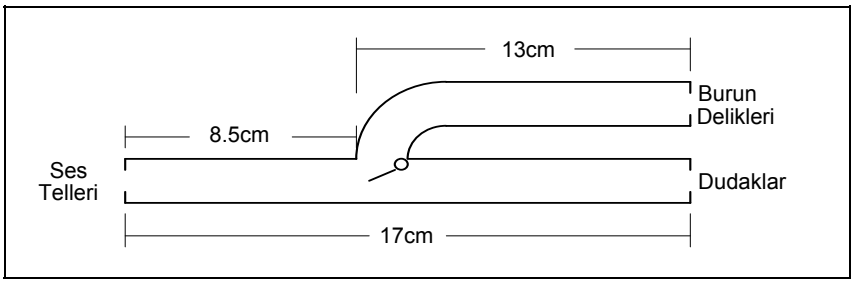
parçalarüstü sesbirimler

olmak üzere ikiye ayrılır. /a/, /d/, /k/ gibi sesbirimler, parça sesbirimlere karşı gelir. Fonemler parça sesbirimlerdir. Fonemlerin ya da parça sesbirimlerin en önemli özellikleri anlam ayırıcı özelliklerinin bulunmasıdır. Anlam ayırıcılık, parça sesbirimleri örten *vurgu (stress), uzatma (length), uyum (harmony), perde değişimi (pitch variation) ve ezgi (intonation)* ile de sağlanabilmektedir. Bunlar parçalar üstü sesbirimler ya da *bürünbirimler* olarak tanımlanmaktadır.

2.4. Ses Yolunun Akustiği

Çizim 2.3'te görüldüğü gibi, ses yolu karmaşık bir yapıya sahiptir. Ancak bu yapının akustiği yalın bir model kullanılarak incelenmektedir. Bu model çerçevesinde ses

yolu, gırtlaktan dudaklara kadar 17cm uzunluğunda bir boru ve gırtlaktan 8.5cm uzaklıkta çatallaşan ve 13cm uzunluğundaki ikinci bir borudan oluşuyor biçimde düşünülür. Sözkonusu uzunluklar erkeklerde raslanan ortalama uzunluklardır. Boruların çatallandığı yer yumuşak damak ve küçük dil ile açılıp kapanan ağız ve geniz yollarının kesiştiği kesimi temsil etmektedir (Çizim 2.10).



Çizim 2.21. Ses Yolunun Yalın Akustik Modeli

Bu boru modeline göre, ses telleri tarafından üretilen ve Çizim 2.4'te gösterilen vuru dizisinin içerdiği harmoniklerden *doğal sıklık* olarak tanımlanan ve:

$$f_n = (2n-1)c / 4l \qquad n = 1, 2, 3,...;\ c = 350m/s;\ l = 17cm$$

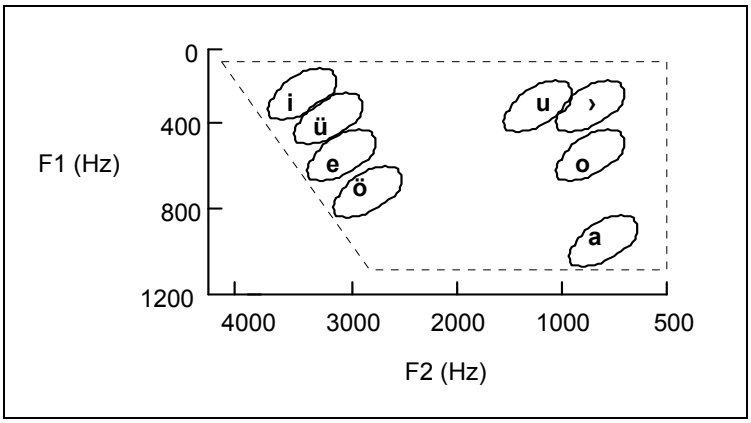
formülü ile belirlenen sıklıktaki harmonikler rezonansa girmektedir. Rezonansa girip yükseltilen bu harmonikler *sesin formantları* olarak bilinir. Yukarıdaki formüle göre rezonans sıklıkları, yaklaşık 500Hz, 1500Hz, 2500Hz, ... olarak belirlenebilmektedir. Gerçekte, ses yolunun tam bir boru gibi olmaması nedeniyle, formantlar arası adımlar her zaman 1000Hz olmayabilmektedir. Sesin içindeki formantlar F_1,F_2,F_3 .. gibi ifade edilmektedir. F_0 ise sesin perdesini temsil etmektedir. Bu formantlardan ilk ikisi (F_1, F_2) akustik incelemede, en az ilk üçü (F_1, F_2, F_3) sesli ifade tanımada, en az beşi (F_1, F_2, F_3, F_4, F_5) ise ses sentezi için gerekli olmaktadır. Erkek ve kadınlar için, ilk üç formanta ilişkin sıklık aralıkları Çizelge 2.1'de verilmiştir.

Çizelge 2.2. Erkek ve Kadında İlk 3 Formant için Sıklık Aralıkları

	Erkek	*Kadın*	*Rezonans Bant Genişliği*
F_1	200-800Hz	250-1000Hz	40-70Hz
F_2	600-2800Hz	700-3300Hz	50-90Hz
F_3	1300-3400Hz	1500-4000Hz	60-180Hz

Formant Sıklıkları ve Sesliler arası İlişki

Daha önce de belirtildiği gibi, ses tellerinin titreşmesi, birkaç istisna dışında ancak ünlülerle söz konusu olmaktadır. Ses tellerinden çıkan titreşimler, ses yolu değişik biçimlere sokularak filtrelenmekte ve kimi sıklıklar için rezonans oluşturularak bunların formantlar olarak belirginleşmesi sağlanmaktadır. Bu durumda, ses tellerinin titreşiminden kaynaklanan ünlüleri F_1, F_2, F_3,.. gibi formant sıklıklarıyla tanımlamak mümkün olmaktadır. Ünlüler, akustik yönünden, genellikle ilk iki formant sıklığı ile parametrelenmektedir. Bu bağlamda İngilizce /i/ sesbiriminin erkekler için, F_1 ve F_2 değerleri, ortalama 255Hz ve 2330Hz olarak ölçülmektedir (Peterson, 1961). Türkçe /a/ sesbirimi için bu değerler 590Hz ve 1430Hz'tir (Ergenç, 1989). Ünlüleri F_1 - F_2 koordinat sisteminde (düzleminde) gösteren çizime *Koenig Skalası* adı verilmektedir. Koenig Skalasının F_1 ve F_2 yer değiştirirek çizilen biçimi, daha önce anılan *ünlü dörtgenini* çizimi vermektedir (Çizim 2.11).



Çizim 2.22. Formant Sıklıklarına göre ünlü Dörtgen Örneği

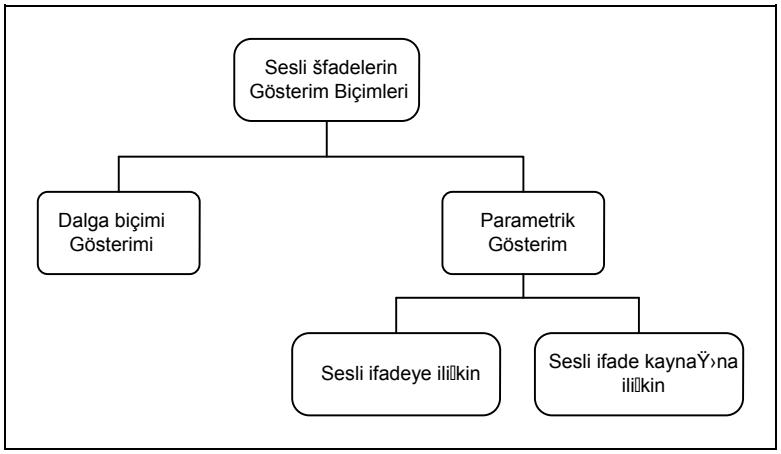
3. SESLİ İFADELERİN TANIMA SÜRECİNE HAZIRLANMASI

Sesli ifadelerin, bilgisayar destekli olarak tanıma sürecine sokulabilmesi için bunların öncelikle bu sürece hazırlanmaları gereklidir. Bu amaçla sesli ifadelerin (bir mikrofon aracılığıyla) örneksel sinyallere dönüştürülmesi, sayısallaştırılması, sayısallaştırılan bu sinyallerin gerekirse filtrelenmesi, (örneğin sesler, fonemler, sözcükler olarak) birimlenmesi ve tanıma işlemlere taban oluşturacak parametrik yapılar ya da yalın modellerle ifade edilen biçimlere dönüştürülmesi gerekmektedir.

Ses sinyalleri, bilgisayar ortamında genelde iki biçimde temsil edilebilmektedir. Bu temsil ya da gösterim biçimleri:

- dalga biçimi ile gösterim *(waveform)*
- parametre tabanlı ya da parametrik gösterimdir.

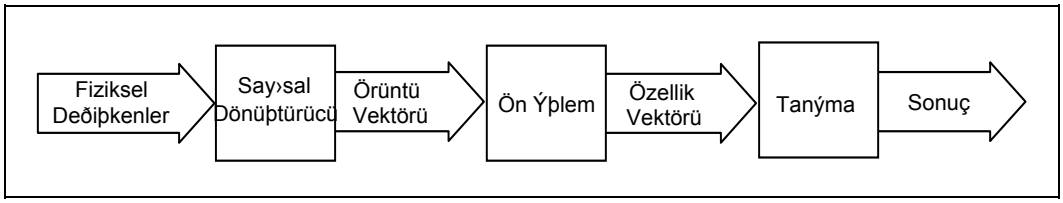
Dalga biçimi ile gösterimde örneksel sesli ifade sinyalleri, örnekleme teoremine uygun olarak örneklenir ve sayısallaştırılır. Sinyal, sayısal örneklemlerin *tümü* ile temsil edilir. Parametrik gösterimde ise sesli ifadenin *tümü* bir dizi parametre ile temsil edilmeye çalışılır. Bu parametreler, ya doğrudan sesli ifadeye ya da sesli ifade üretme modeline ilişkin olabilmektedir. Parametrik gösterime ulaşmak için, önce dalga biçimi gösteriminden geçilmekte ve bu gösterim biçiminden, sesli ifadeyi temsil edecek parametreler elde edilmektedir. Sözkonusu parametreler, örneğin *LPC (linear predictive coding), cepstrum* katsayıları gibi parametreler olabilmektedir. (Çizim 3.1)(Rabiner 1978).



Çizim 3.1. Ses sinyali için kodlama yöntemleri

Sesli ifade sinyalleri, genel bir perspektif içinde bir örüntü *(pattern)* olarak düşünülebilir. Matematiksel olarak, her örüntü bir dizi parametre ile temsil edilebilmektedir. Sesli ifade örüntüleri, *özellik vektörü* (*feature vector*) olarak anılan vektörlerle temsil edilebilmektedir. Özellik vektörü, vektör boyutunu da belirleyen *n* değişik parametreden oluşur. Zaman içinde, bir sesli ifade örüntüsü birden çok vektör ile gösterilir. Uygulamalarda vektör boyutlarının büyük tutulması temsil doğruluğunu artırmakla birlikte bu vektörler üzerinde yürütülecek işlemleri zorlaştırmaktadır. Bu nedenle vektör elemanlarını oluşturan parametrelerin kendi aralarında gruplandırılması, paralellik gösterenlerin birleştirilmesi yoluna gidilebilmektedir. Bu tür işlemler genelde özellik çıkarma (*feature extraction*) ya da *vector quantization* işlemi olarak anılır.

Sesli ifade tanıma süreci, diğer örüntü tanıma yöntemlerine benzer aşamaları içermektedir. Sesli ifadeler, mikrofon aracılığı ile örneksel değerler olarak örneklenip sayısala dönüştürülmekte ve *örüntü vektörleri* elde edilmektedir. Örüntü vektörü sesli ifadeden elde edilen örneklem dizisine verilen addır. Örüntü vektörleri, ön işlem sonucu özellik vektörlerine dönüştürülür. Sesli ifadeler özellik vektörleriyle işleme sokulur. Tanıma, özellik vektörlerine dayalı olarak yürütülür (Çizim 3.2).



Çizim 3.2. Genel bir örüntü tanıma sisteminin ilke çizimi.

3.1. Sesli İfadelerin Sayısallaştırılması

Ses sinyalinin bilgisayar ortamına aktarılabilmesi için öncelikle bu sinyalin örnekselden sayısala dönüştürülmesi gereklidir. Bu işlem, ses sinyalinin temel özellikleri kaybolmadan yapılmalıdır. Bunun için sayısallaştırılmaya taban oluşturan örnekleme işlemi, sinyal içindeki en yüksek frekansın en az iki katı sıklıkta yapılmak zorundadır. Sesli ifade sinyallerinin sayısallaştırılmasında değişik kodlama yöntemleri kullanılmaktadır(Rabiner 1978). Bu yöntemler, İngilizce adlarıyla:

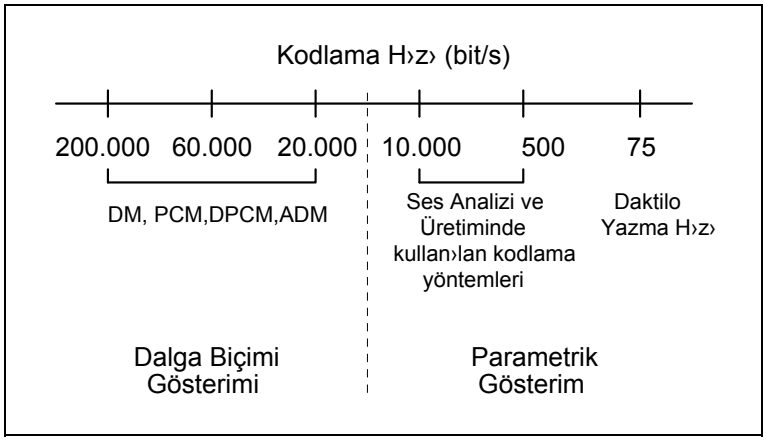
- *PCM Pulse Code Modulation*
- *Log-PCM LogarithmicPulse Code Modulation*
- *APCM Adaptive Pulse Code Modulation*

- *DPCM Differential Pulse Code Modulation*
- *ADPCM Adapted Differantial Pulse Code Modulation*
- *DM Delta Modulation*

yöntemleridir. Bu yöntemlerden en yaygın olarak kullanılanı, *PCM Pulse Code Modulation* yöntemidir. Bu yöntemde sesli ifade sinyalleri en az 8kHz sıklığında örneklenmekte ve her örnek belirli sayıda bit üzerinden kodlanmaktadır. Örneklem genlikleri doğrusal olarak kodlanırsa doğrusal *PCM* yönteminden, logaritmik bir skalada kodlanırsa *Log-PCM* yönteminden söz edilmektedir. *Log-PCM* yöntemiyle düşük genlikli sinyaller daha iyi kodlanabilmektedir.

PCM yönteminin değişik türevleri bulunmaktadır. Bu türevler kodlama sonucu elde edilen bit sayısını düşük tutmayı amaçlamaktadır. *DPCM Differential Pulse Code Modulation* yönteminde örnek genliğini kodlamak yerine bir önceki örnekle arasındaki fark kodlanmaktadır. Bu yaklaşım kodlama için gerekli bit sayısının düşük tutulmasına olanak sağlamaktadır. Genlikler arası farkın salt iki düzeyle (tek bitle) kodlandığı durumda *DM Delta Modulation* yöntemi sözkonusu olmaktadır.

Yukarıda verilen yöntemlerde, örnekleme hızı ve her örneği kodlamak için kullanılan bit sayısı, birlikte, veri aktarım hızını belirlemektedir. Bu nedenle bu yöntemler açıklanırken bunlarla ilgili aktarım hızlarından da söz edilmektedir. Burada anılan aktarım hızı sesli ifade tanıma sistemleri için iki örneklem arasında kalan işlem hızı olarak yorumlanmalıdır. Çizim 3.3'de, sözkonusu kodlama yöntemleri, aktarım hızları ile birlikte verilmiştir. Sesin kodlanmasında kullanılan bu yöntemlerin seçimi maliyet, gösterim biçiminin esnekliği ve sesin kayıt kalitesi gibi faktörler göz önüne alınarak yapılmaktadır.

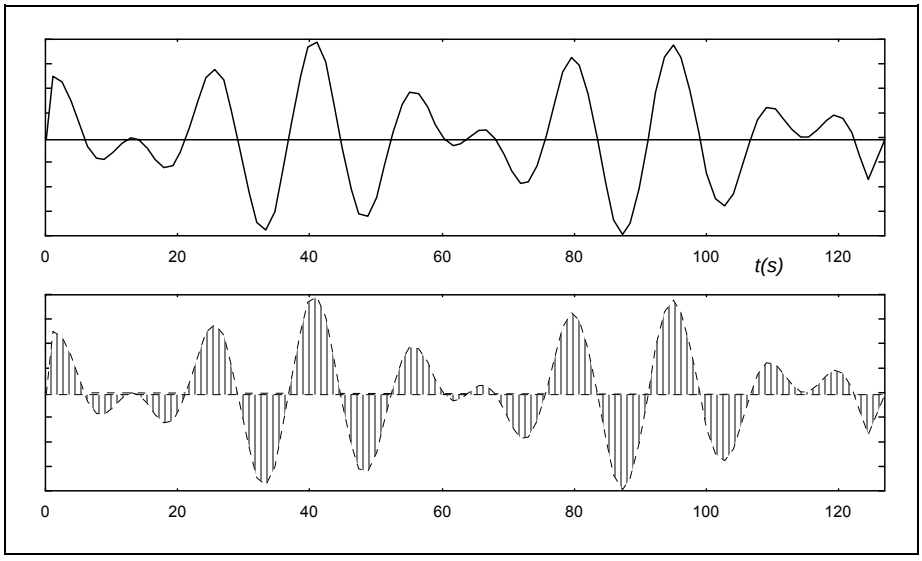


Çizim 3.3. Ses Sinyali Kodlama Yöntemleri ve Aktarım Hızı Örnekleri

Örnekselden sayısala dönüşüm sesli ifade tanımada verilerin bilgisayar ortamına alınmasındaki ilk adımdır. Bu dönüşüm yoğun tümleşik çevrim teknolojisini kullanan birimlerce sorunsuz bir biçimde yapılmaktadır. Sesli ifadelerin sayısallaştırılmasında, örnekselden sayısala dönüşüm hızı ve bit türünden kod uzunluğu önemli iki kriterdir. Sesli ifade tanıma sistemlerinde örnekleme hızı genelde en az 8kHzve üzerinde, sözcük genişliği de genellikle 12 bit ve üzerinde olmaktadır. Örnekselden sayısala dönüştürme iki temel aşamayı içermektedir(Enden 1989): (Çizim 3.4)

- Örnekleme *(sample and hold)*
- Niceleme *(quantization)*

Örnekleme belirli zaman aralıklarında sinyal örneği alma işlemidir. Örnekleme sıklığı *Nyquist* ilkesine bağlı kalınarak belirlenir. Bu ilkeye göre örnekleme sıklığı, örneklenen sinyal içindeki en büyük sıklığın en az iki katı olmalıdır. Bu en büyük sıklık F_n ise, olması gerekli en küçük örnekleme sıklığı $1/T > 2F_n$ formülüyle bulunur.



Çizim 3.4. Ses Sinyalinin Örneksel ve Sayısal Görünümleri

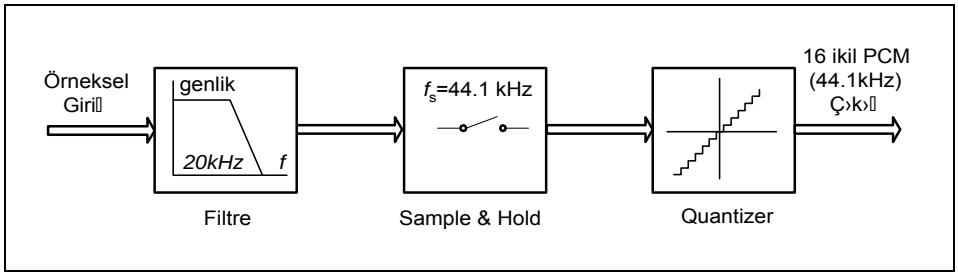
Kodlama aşamasında, örneklenen sinyal değeri ikili koda dönüştürülmektedir. Kod uzunluğu ne olursa olsun kodlama aşamasında son bit için mutlaka bir yuvarlama

yapmak zorunludur. Bu amaçla, genelde üç farklı yol kullanılır. Bunlar ve ilgili duyarlık değerleri aşağıda verilmiştir:

- Yuvarlama *(rounding)* $-q/2 \le x_Q - x < q/2$
- *Value truncation* $-q \le x_Q - x < 0$
- *Magnitude truncation* $\begin{cases} -q \le x_Q - x < 0 & \text{eğer} x > 0 \\ 0 \le x_Q - x < q & \text{eğer} x < 0 \end{cases}$

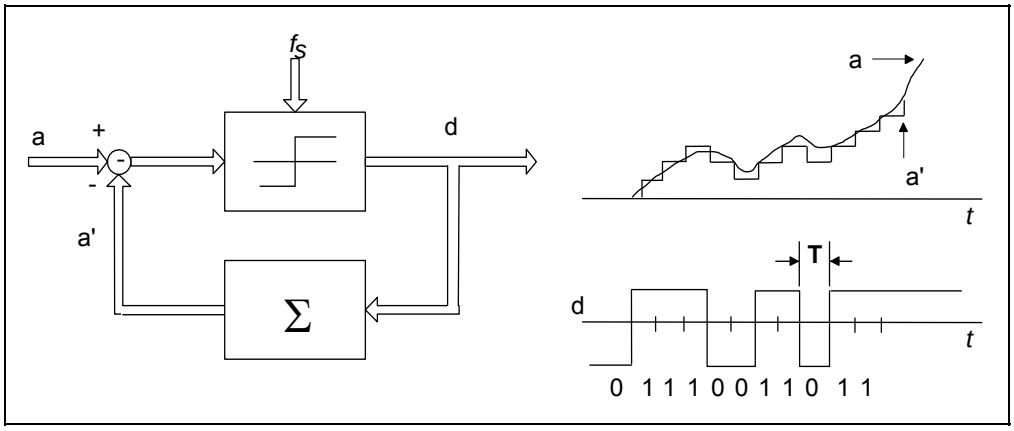
Yukarıda verilen eşitsizliklerde q kodlama adım değerini, x kodlanan örneksel değeri, x_Q ise kodlanmış değeri temsil etmektedir. Verilen eşitsizlikler kodlama sonucu elde edilen değer üzerinde yapılan yanlışın sınırlarını belirtmektedir.

Pulse Code Modulation (PCM) yöntemi yüksek nitelikli dönüşümler için tercih edilen bir yöntemdir. Yüksek nitelikli seslendirme teknojisinde *(HI-FI)* 16 bit üzerinden kodlama seçeneği kullanılmaktadır. Bu kod uzunluğu ile 2^{16}=65.536 değişik düzey kodlanabilmektedir. Çizim 3.5'de *PCM* tekniği kullanılarak örnekselden sayısala dönüşüm süreci verilmiştir. Bu yöntemde sinyal önce *low pass* türü bir filtreden geçirilmekte sonra örneklem alınmaktadır. Alınan örneklem değeri, bir sonraki örneklem alınana değin kodlama biriminin girişlerinde sabit bir biçimde tutulmaktadır *(sample and hold)*.(Enden 1989)



Çizim 3.5. *PCM* tekniği

Yukarıda da belirtildiği üzere, Delta Modülasyon kodlama tekniği yaygın olarak kullanılan bir diğer tekniktir. Delta modulatöründe kodlama tek bit üzerinden gerçekleşir. Örneklem değeri bir önceki örneklem değeri ile karşılaştırılır (Çizim 3.6). Bulunan fark pozitif ise, çıkış olarak 1, değilse 0 biti üretilir.



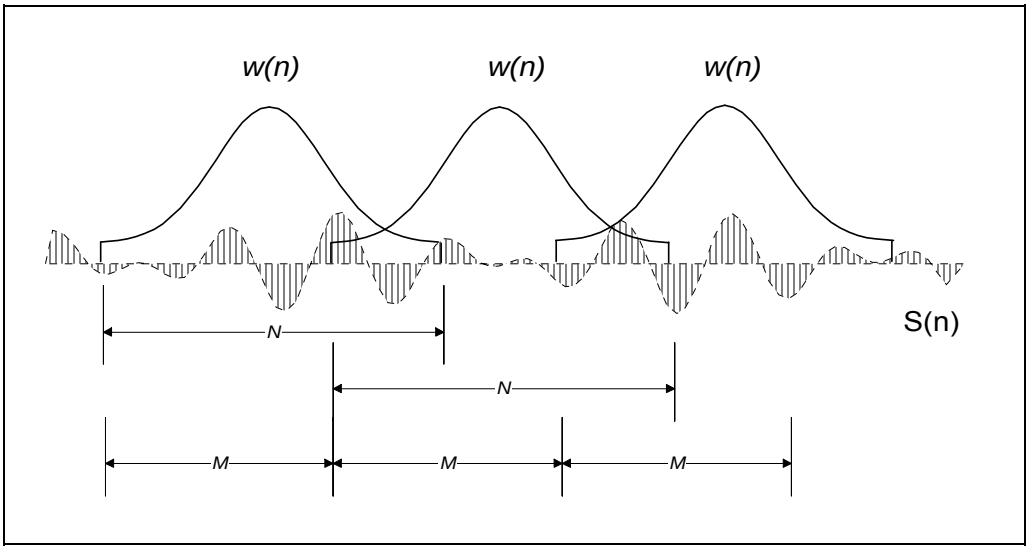
Çizim 3.6. Delta Modülasyon tekniği

3.2. Sesli İfadeler Üzerinde Yürütülen Ön İşlemler

Sesli ifadeler, tanıma sürecine geçilmeden önce kimi ön işlemlere tabi tutulurlar. Bu ön işlemler tanıma sürecine hazırlık olarak bilinir. Sözkonusu ön işlemler aşağıda açıklanmıştır:

3.2.1. Pencereleme

Sayısallaştırılan sinyaller bilgisayar ortamında ikili sözcükler olarak tutulurlar. Sinyal işleme aşamasında, genellikle sesli ifadeleri temsil eden sözcüklerin tümünü bir seferde ele alma olanağı bulunmaz. Bu nedenle, bütünün anlamlı uzunlukta parçalara bölünmesi gerekir. Zira *FFT (Fast Fourier Transformation)* gibi işlemlere ilişkin algoritmalar belirli uzunlukta (belirli sayıda sözcükten oluşan) veri kümeleri üzerinde çalışabilmektedir. Bir seferde işleme alınacak sözcük kümesi, sinyal içinde bir pencere *(window)* olarak tanımlanır. Sözcükler örnekleme sonucu elde edilen değerler olduklarından sözcük sayısının örnekleme periyodu ile çarpılması ile bir zaman birimi elde edilir. Bu nedenle pencere, bir zaman dilimi olarak da düşünülür. Sesli ifadeleri belirli uzunluktaki sözcüklere ayırma işlemi ise pencereleme *(windowing)* olarak bilinir.

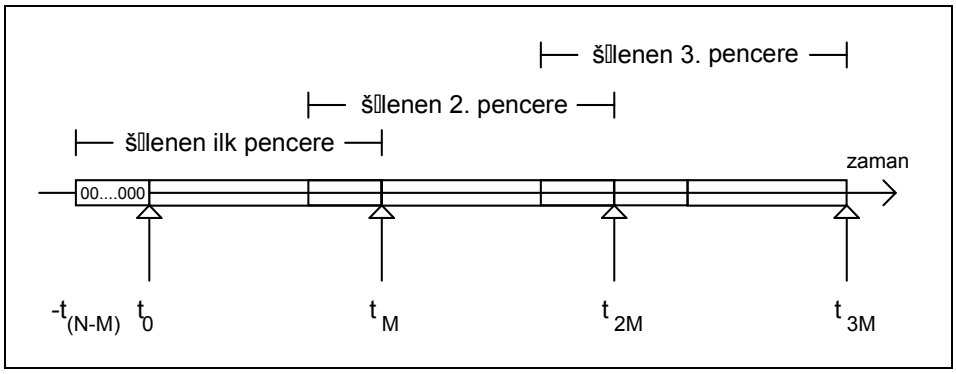


Çizim 3.7. Ses Sinyalinin Pencerelenmesi

Pencereleme işlemi, çoğu kez birbirinden kopuk veri kümeleri yerine üstüste binen (örtüşen) kümeler oluşturulacak biçimde yapılır(Moris 1988). Çizim 3.7'de, sayısal bir sinyal üzerinde oluşturulan pencere örnekleri verilmiştir. Bu çizimde, *N* örneklemlik pencere içinde *(N-M)* örneklem, bir önceki penceredeki örneklemlerle örtüşen örneklemlerden oluşmaktadır. Sesli ifade sinyalleri için, pencerelerin genişliği tek bir sesin özelliklerini içerecek kadar kısa ve birbirini izleyen *pitch* harmoniklerini kaybetmeyecek kadar da uzun olmalıdır. Sesli ifade tanımada pencere genişlikleri yaklaşık 25-75 ms arasında olmaktadır. 10 kHz'lik bir örneklem hızı için 512 sözcüklük bir pencere 51,2 ms'lik bir sinyal zaman dilimine karşı gelir.

Ses sinyalini temsil eden örneklemler pencereler olarak saklanır ve işleme alınır. Pencere boyu, pencere oluşturma periyodundan daha büyük tutularak, ardarda alınan percerelerde örtüşmeler sağlanır. Derlenen pencereler pencere oluşturma periyodunda işleme sunulur. Pencere oluşturma periyodu, pencere boyundan (*N*'den) küçük olduğundan pencere oluşturma periyodu sonunda elde edilen örneklem sayısı (*M*) bir pencereyi doldurmaz. (*i*) inci aşamada işleme sunulan pencere, (*i*) inci pencere oluşturma periyodunda derlenen (*M*) adet örneklemi içerir. Eksik kalan (*N-M*) adet örneklem, bir önceki ((*i* - 1) inci) pencere oluşturma periyodu sonunda elde edilen örneklemlerden sağlanır (Çizim 3.8). Ancak derlenen ilk pencere (*i*=1) için bunu yapma olanağı bulunmaz. Zira bu durumda (*i* - 1) inci pencere diye birşey yoktur. Birinci pencere tümüyle doldurulmadan ikinci pencerenin oluşturulmasına

geçilmek zorunda kalınacağından ilk pencerenin kimi sözcükleri boş kalır. Bu sözcükler 0 ile doldurulur. Bu işleme sıfırla doldurma (*zero padding*) işlemi denir.



Çizim 3.8. Ardarda Oluşturulan Pencereler

Yukarıda açıklanan ilke gereği derlenen pencereler bir dizi (*N* adet) örneklemden oluşur. Bu örneklemler, incelenen sinyalin bir kesimidir. İşlem sırasında bu kesimin önünde ve arkasındaki kesimlerin ele alınmıyor olması sinyalin bir dikdörtgenle maskelenmiş biçiminin incelenmesi gibi bir sonuç doğurur. Bu durumda sinyalin, işlenen pencere dışında sıfır düzeyindeymiş gibi ele alınması, doğal olarak sorun yaratır. Bu sorunu hafifletmek üzere pencereleme işlemi, daha genel bir perspektifle sinyalin uygun bir fonksiyon ile *convolution* işlemine tabi tutulması olarak düşünülür. *Convolution* işlemi, zaman ekseninde, sinyal ile seçilen pencere fonksiyonunun özel bir çarpımıdır.

Convolution işlemi,

$$\begin{aligned} y(t) &= x(t) \otimes w(t) \\ y(t) &= \int_{-\infty}^{\infty} x(t-\tau) w(\tau) d\tau \end{aligned} \qquad (3.1.a)$$

biçiminde ifade edilmektedir. Burada *x* sinyali, *w convolution* fonsiyonunu, ⊗ ise *convolution* işlemini göstermektedir. *Convolution* işleminin kesikli fonksiyonlar için kullanılan biçimi de aşağıdaki gibidir(Oppenheim 1989):

$$\begin{aligned} y[k] &= x[k] * w[k] \\ y[k] &= \sum_{i=-\infty}^{+\infty} x[i] \cdot w[k-i] = \sum_{i=-\infty}^{+\infty} w[i] \cdot x[k-i] \end{aligned} \qquad (3.1.b)$$

Convolution fonksiyonu olarak kimi klasikleşmiş özel fonksiyonlar kullanılmaktadır. Bu özel fonksiyonlar aşağıda görünümleri ile birlikte verilmiştir. Bunlardan *rectangular window* fonksiyonu yukarıda sorun yarattığı belirtilen dikdörtgen pencereye karşı gelmektedir.

Rectangular window:

$$w[i] = \begin{cases} 1 \quad \text{eğer} \quad 0 \le i \le n-1 \\ 0 \quad \text{diğer durumlarda} \end{cases} \qquad (3.2)$$

Bartlett window:

$$w[i] = \begin{cases} \dfrac{2i}{n-1} \qquad \text{eğer} \qquad 0 \le i \le \dfrac{n-1}{2} \\ 2 - \dfrac{2i}{n-1} \quad \text{eğer} \quad \dfrac{n-1}{2} < i \le n-1 \\ 0 \quad \text{değilse} \end{cases} \qquad (3.3)$$

Hanning window:

$$w[n] = \begin{cases} \dfrac{1}{2}\left\{1 - \cos\left(\dfrac{2\pi n}{L-1}\right)\right\}, \quad \text{eğer} \quad 0 \le n \le L-1 \\ 0 \quad \text{değilse} \end{cases} \qquad (3.4)$$

Hamming window:

$$w[i] = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi \cdot i}{n-1}\right), \quad \text{eğer} \quad 0 \le i \le n-1 \\ 0 \quad \text{değilse} \end{cases} \qquad (3.5)$$

Blackman window:

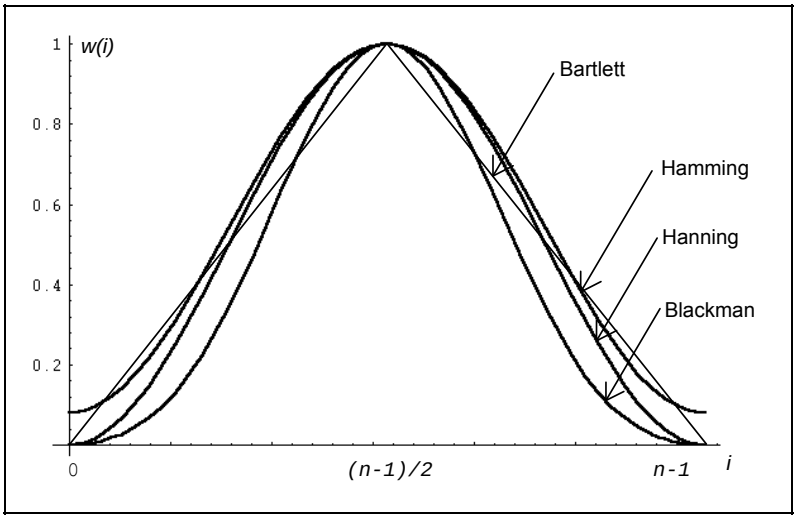
$$w[i] = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi \cdot i}{n-1}\right) - 0.08\cos\left(\frac{4\pi . i}{n-1}\right) \text{ eğer } 0 \le i \le n-1 \\ 0 \text{ değilse} \end{cases} \qquad (3.6)$$

Kaiser window: İçerdiği β parametresi itibarıyla, daha önce açıklanan değişik pencereleme fonksiyonlarına dönüşebilen özel bir pencereleme fonksiyonudur.

$$w[i] = \begin{cases} \dfrac{I_0\left(2\beta\sqrt{\dfrac{i}{n-1} - \left(\dfrac{i}{n-1}\right)^2}\right)}{I_0(\beta)} \text{ eğer } \; 0 \le i \le n-1 \\ 0 \text{ değilse} \end{cases} \qquad (3.7)$$

Burada I_0 Bessel fonksiyonudur.

$$I_0(x) = 1 + \sum_{k=1}^{\infty}\left[\frac{(x/2)^2}{k!}\right]^2 \qquad (3.8)$$



Çizim 3.9. Yaygın olarak kullanılan Pencereleme Fonksiyonları

3.2.2. Filtreleme

Filtreleme işlemi zaman ve sıklık evreninde ayrı ayrı ele alınabilinir. Ön filtreleme işlemi, çoğu kez zaman evreninde uygulanmaktadır. Filtreleme, filtrelenecek sinyal ile filtre fonksiyonu arasında *convolution* işleminin uygulanmasıdır. Kullanılacak

filtre fonksiyonu, sıklık-genlik evreninde anlamlı bir biçimde kolayca belirlenen filtre fonksiyonunun zaman-genlik evrenindeki dönüşümü *(transformation)* olarak seçilmektedir. Yukarıda anılan pencereleme işlemleri de, zaman ekseninde sinyal üzerinde yapılan filtreleme işlemleri olarak düşünülür(Oppenheim 1989).

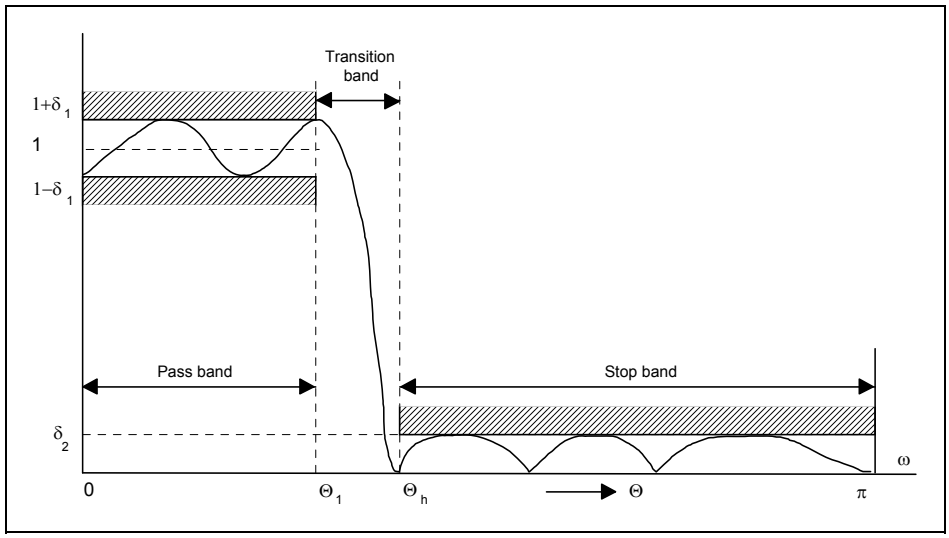
Genel olarak sıklık filtreleri;

- *Low pass*
- *High pass*
- *Band pass*
- *Band stop*

türünde olabilmektedir. Filtreleme fonksiyonu Çizim 3.10'da görüldüğü gibi üç kesim üzerinden incelenebilir. Bunlar:

- *Pass band*
- *Transition band*
- *Stop band*

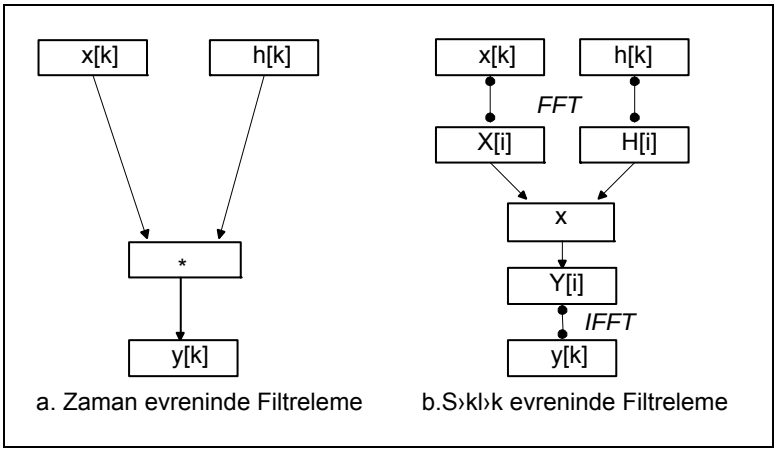
kesimleridir.



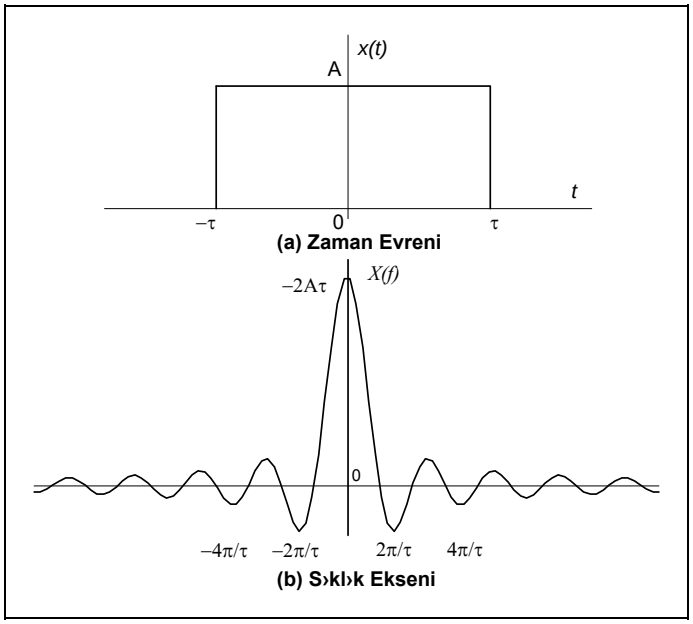
Çizim 3.10. Genlik Sıklık Evreninde Filtre Fonksiyonu Eğrisi

Filtreleme, filtrelenecek sinyalin, tercihe göre, gerek sıklık, gerekse zaman evrenlerindeki görünümleri üzerinde uygulanabilir. Zaman evreninde filtreleme

amaçlananı zorlaştırdığında sinyal, *Fast Fourier Transformation (FFT)* tekniği ile sıklık evrenindeki özdeş görünümüne dönüştürülür. Sıklık evreninde filtreleme işlemleri yapıldıktan sonra, sinyalin zaman evrenindeki görünümüne, *Inverse Fast Fourier Transformation* yöntemi ile geri dönülür. Çizim 3.11'de filtrelemenin zaman ve sıklık evrenlerinde uygulanma ilkeleri verilmiştir. Çizim 3.12'de sıklık evreninde, *low-pass* türünden tanımlanmış bir filtrenin, zaman evrenindeki özdeş fonksiyonu verilmiştir. Ses sinyali üzerinde, zaman ekseninde bu fonksiyon ile *convolution* işlemi yapılır ise, sinyal düşük sıklıkların geçirildiği filtreden geçirilmiş olmaktadır.



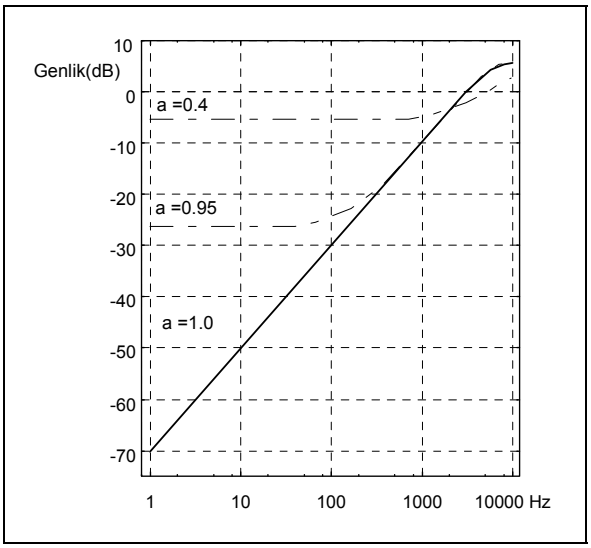
Çizim 3.11. Filtrelemede zaman-genlik ve sıklık-genlik eksenleri



Çizim 3.12. *Low-pass* türü filtrenin zaman ve sıklık evrenindeki özdeş fonksiyonları

***Preemhasis* filtre**

Sesli ifadeler düşük sıklıklar için yüksek enerji değerlerine sahip olmaktadır. Düşük sıklıktaki birleşenlerin yüksek sıklıktakileri maskelememesi için *preemphasis* adı verilen filtre kullanılmaktadır. Bu filtre, $H(z)= 1- az^{-1}$ bağıntısı ile ifade edilebilir. Bağıntıdaki *a* katsayısı 0,0 ile 1,0 arasında bir değere sahiptir. Çeşitli *a* katsayıları için filtre sıklık çıkışı Çizim 3.13'de verilmiştir (Picone 1993).



Çizim 3.13. *Preemphasis* filtresi

3.2.3. *Zero Crossing Rate*

Ses sinyalinin sıfırdan geçiş sayısı, *zero crossing rate* olarak bilinir. Sesli ifade kayıtlarında, ses sinyalinin bulunmadığı kesimlerde bu sayı gürültünün yüksek sıklıkta bir sinyal olmasından dolayı artmaktadır. Sesli ifadelerin yer aldığı kesimlerde ise, *zero crossing rate* olarak bilinen bu değer düşük olmaktadır. Bu özellik, sesli ifadelerin başlangıç ve bitiş noktalarını belirlemede kullanılmaktadır.

Zero crossing rate değeri ile bir sinyalin sıklığını ölçme olanağı bulunur. *Sinüssel* bir sinyalde, her periodda iki sıfırdan geçiş *(zero crossing)* bulunduğundan sinyal sıklığı sıfırdan geçiş sayısının yarısı alınarak hesaplanabilir. Periyodik bir sinyal için sıfırdan geçiş sayısı sıklık değerini elde etmeye olanak verir. Ses sinyali gibi, periodu zaman içinde değişebilen sinyaller için sıfırdan geçiş sayısı kimi sinyal kesimlerine ilişkin, ancak periyod kestirimleri yapmaya olanak verir.

Zero crossing rate matematiksel olarak aşağıdaki gibi ifade edilebilmektedir:

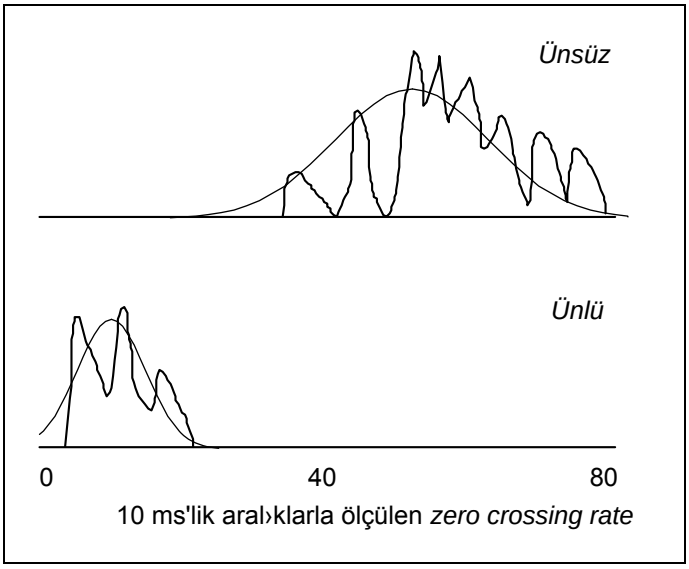
$$Z(k) = \sum_{i=-\infty}^{\infty} \left| sgn\left[x(i)\right] - sgn\left[x(i-1)\right] \right| w(k-i) \tag{3.9}$$

Burada:

$$\begin{aligned} sgn\left[x(i)\right] &= +1 \quad \text{eğer} \quad x(i) \geq 0 \\ &= -1 \quad \text{eğer} \quad x(i) < 0 \end{aligned} \tag{3.10}$$

ve

$$\begin{aligned} w(k) &= \frac{1}{2N} \quad \text{eğer} \quad 0 \leq k \leq N-1 \\ &= 0 \qquad \text{diğer durumlarda} \end{aligned} \tag{3.11}$$



Çizim 3.14.Ünlü ve ünsüz seslerde *zero crossing* değerinin dağılımı (Rabiner 1978)

3.2.4. Enerji

Sesli ifadenin başlama ve bitiş noktasının belirlenmesinde çoğu kez *zero crossing rate* değeri ile birlikte kullanılan bu parametre sesin şiddetine bağımlıdır. Sesin enerjisi, *sort time energy* olarak adlandırılan biçimiyle:

$$E(k)= \sum_{i=-\infty}^{\infty} \left[x(i)\cdot w(k-i)\right]^2 \tag{3.12}$$

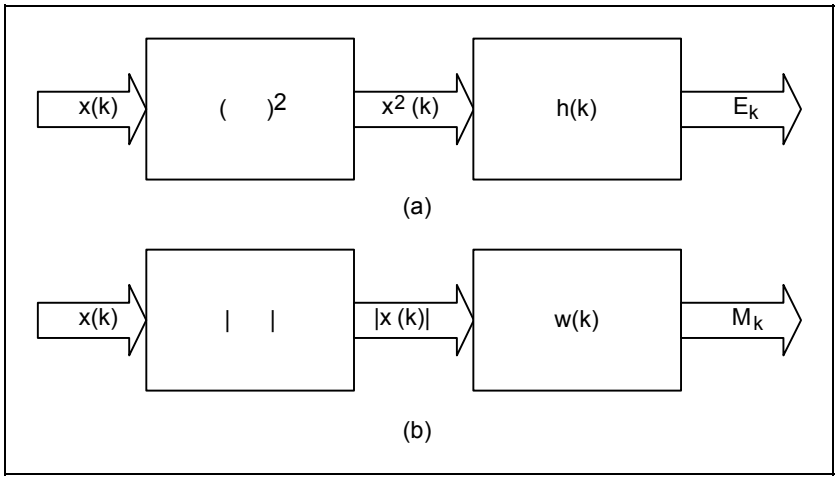
ya da

$$E(k) = \sum_{i=-\infty}^{\infty} x^2(i)\cdot h(k-i)$$
$$h(k) = w^2(k) \tag{3.13}$$

formülleri ile hesaplanmaktadır. (3.13)'te *w(k-i)* enerjinin hesaplandığı pencereyi temsil etmektedir. Enerjiyi, yukarıda verilen formülle hesaplamak yerine, *short time avarage magnitude* olarak bilinen bir diğer parametre ile temsil etmek de olanaklıdır. *Short time average magnitude* aşağıdaki gibi hesaplanır:

$$M(k) = \sum_{i=-\infty}^{\infty} |x(i)|w(k-i) \tag{3.14}$$

Çizim 3.15'de *short time energy* ve *short time avarage magnitude* hesaplama süreçleri verilmiştir.

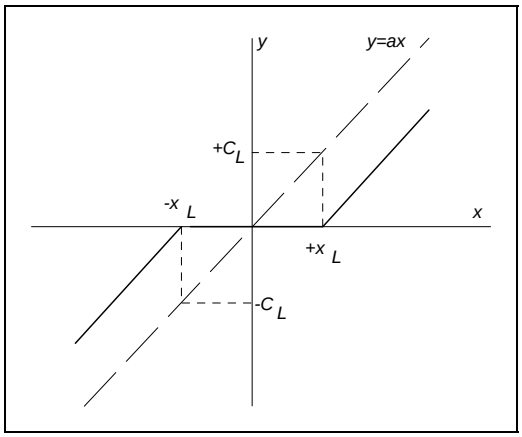


Çizim 3.15. *Short time energy* (a), *Short time avarage magnitude* (b) hesaplaması

3.2.5. *Center Clipping*

Center clipping, sesli ifade sinyalinin gürültüden arındırılması için kullanılan bir yöntemdir. Gürültü, genelde düşük genlik ve enerjili, yüksek sıklıklı bir sinyaldir. Sesli ifade sinyalinin sıfır düzeyine yakın kesimleri çıkarıldığında, gürültü

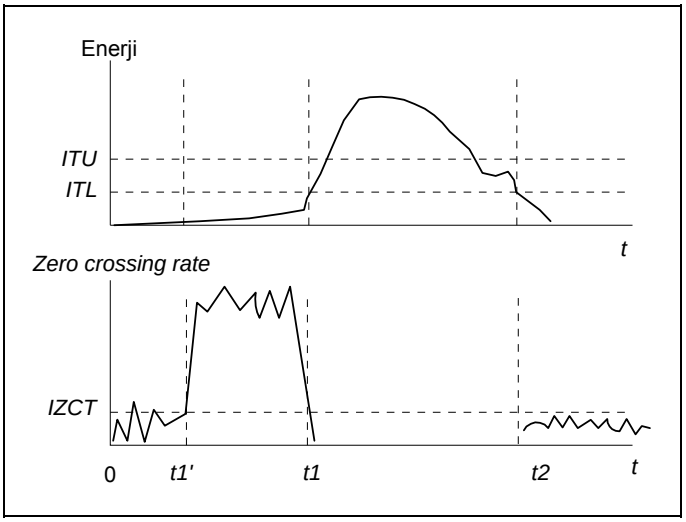
süzülmekte ancak sesli ifade ile taşınan bilgi kaybolmamaktadır. En büyük sinyal genlik değerinin %30'una kadar bir kesimin sinyalden çıkarılması, literatürde söz konusu edilmektedir (Rabiner 1978). Çizim 3.16'da *center clipping*'in $y=ax$ fonksiyonuna uygulanması örneklenmiştir. Burada $-x_L$ ve $+x_L$ arasında fonksiyon sıfırlanmakta, bu aralığın dışında fonksiyon değerinden $-C_L$ ve $+C_L$ (*clipping level*) değerleri çıkarılmaktadır.



Çizim 3.16. Örnek bir *Center Clipping* işlemi

3.2.6. Sinyalin Başlangıç ve Bitiş Noktalarının Belirlenmesi

Sesli ifade tanıma sistemlerindeki önemli sorunlardan birisi de, sesli ifadelerin sürelerinin değişkenliğinden kaynaklanan belirsizlikdir. İki aynı sesli ifade sinyali birbirlerinden farklı sürelere sahip olabilmektedir. Bu tür olumsuzlukları gidermek amacıyla sesli ifadelere ilişkin bilgiler, şablon sözcük ya da alt bileşenlerine (*segment*) ayrılmış biçimleri ile saklanır ve ele alınırlar. Bu soruna, genelde sözcüklerin baş ve sonundaki düşük enerjili fonemler ya da konuşmacıların sesli ifadeleri kısa nefes sesi ile bozmaları ya da uzatmaları neden olmaktadır. Sözcüklerin sonundaki nefes sesi, ilgili fonemin, dolayısıyla ilgili sözcüğün tanınmasını zorlaştırmaktadır.



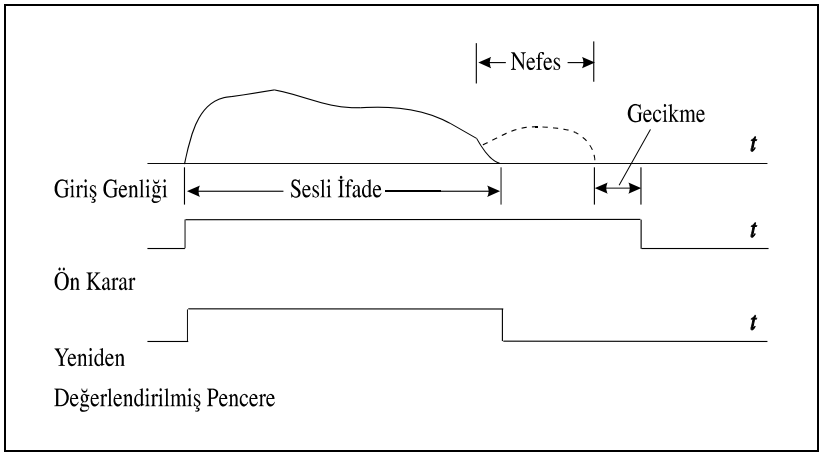
Çizim 3.17. Bir sözcüklük sesli ifade sinyaline ilişkin örnek *enerji* ve *zero crossing rate* ölçümü

Çizim 3.17'de sözcüğün başlangıç ve bitiş noktalarının bulunmasına ilişkin *Rabiner ve Sambur*'un enerji ve *zero crossing rate* fonksiyonlarının değişimlerini kullanan başlangıç-bitiş noktaları belirleme yöntemi verilmiştir (Rabiner 1978). Bu yöntemde incelenen sinyalin enerji ve *zero crossing rate* fonksiyonlarından yararlanılmaktadır. sinyalin, enerji fonksiyonu üzerinde *ITL* (*Interval Threshold Low*) adlı değer aşıldığında t_1 başlangıç noktası, *ITL* değerinin altına düştüğünde ise t_2 bitiş noktası belirlenmiş olur. Ancak sözcüklerin kesin başlangıç noktasının bulunması için aynı zamanda *zero crossing rate* fonksiyonuna da başvurulur. *Zero crossing rate* fonksiyonu üzerinde, *IZCT* (*Interval Zero Crossing Threshold*) olarak adlandırılan düzeyin geçildiği nokta bulunur. Bu nokta (t'_1), sesli ifade sinyalinin, t_1 yerine yeni başlangıç noktası olarak alınır. Ancak bu noktanın kolayca ve bir seferde belirlenmesi, sinyal üzerindeki gürültüden dolayı her zaman mümkün değildir. (t'_1) noktası belirlenirken *IZCT* düzeyinin, belli bir süre içinde, ardarda, örneğin en az üç kez aşılması koşulu aranır.

Zero crossing rate fonksiyonu düşerken geçilen nokta (t_1), (ünlülerin göreli düşük sıklıklı olduğu anımsanırsa) sözcük içindeki ilk ünlünün başladığını göstermektedir. Başlangıç ve bitiş noktalarını belirleme yönteminin iyileştirilebilmesi için, *ITL* olarak adlandırılan tek bir eşik düzeyi yerine, *ITL* ve *ITU (Interval Threshold Upper)* olarak adlandırılan iki değişik eşik düzeyi kullanılabilir. Eğer sesli ifade sinyalinin genlik düzeyi düşükse *ITL,* değilse *ITU* eşik değeri kullanılır. Bu iyileştirmenin ötesinde, *ITL* eşik düzeyinin sesli ifadeye göre uyum sağlayabilir nitelikte olması da

düşünülebilir. Ancak bu yaklaşımla, sinyal içinde sesli ifadenin bulunmadığı kesimlerde gürültü, eşik düzeyinin yanlış belirlenmesine neden olabilmektedir. Bu nedenle sınamaları ardarda *n* nokta için gerçekleştirmek ve bu noktaların hepsi için de eşik değeri aşılıyorsa bunun bir sesli ifadenin başlangıcı olduğuna karar vermek gereklidir. Eşik değerinin ilk aşıldığı *n* ardışık nokta sesli ifade içinde düşünülür. Diğer bir yaklaşım ise iki eşik değeri kullanmaktır. Bu yaklaşımda ilk eşik değeri geçildiğinde sesli ifadenin başladığı kabul edilir. İkinci eşik değeri geçildiğinde ise sesli ifadenin varlığı kanıtlanmış sayılır. Bitiş noktasının belirlenmesinde ise benzer bir yaklaşım kullanılır. Belirlenen noktadan başlayarak belirli sayıda nokta incelenen sinyale katılarak gerekli marj yaratılır.

Sözcük sonunun belirlenmesinde, özellikle sesli ifadelerin patlamalı ünsüzler ile bitişi sorunlar yaratmaktadır. Sözcük sonuyla ilgili olarak, sözcüklerin sonlanmadan önceki sessizlik çerçevelerinin tanınması gerekli olmaktadır. Bu bağlamda Sambur ve Rabiner sesin sonunu, *treated breath* olarak farklı bir fonem olarak ele almış ve ayrıca tanımaya çalışmışlardır. Yaptıkları uygulamalarda sesli ifadenin başlangıcı genlik değerinin artması, sonu ise nefes sesinin bitmesi biçiminde gerçekleşmektedir. Bu kurala ilişkin olarak, Çizim 3.18'de bir sesli ifadenin sınırlarının nasıl belirlendiği gösterilmiştir.

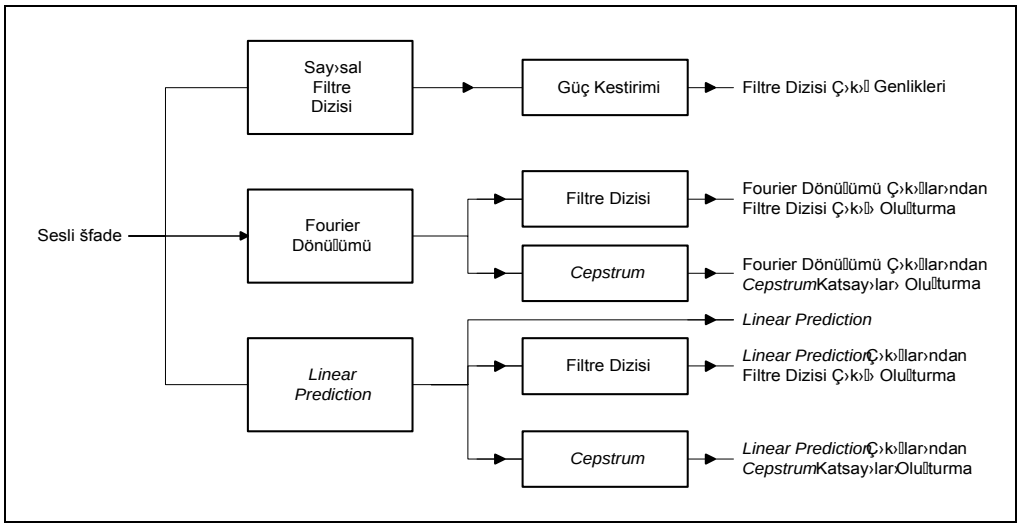


Çizim 3.18. Sesli ifadenin sözcük sınırlarının belirlenmesi

3.3. Sesli İfade Sinyallerinin Modellenmesi

Spektral analiz sinyali oluşturan harmoniklerinin incelenmesine verilen addır. Sesli ifade sinyallerinde spektral analiz, bu sinyallerin parametrik biçime dönüştürülmesinde kullanılır. Günümüzde sesli ifade tanıma sistemlerinde,

parametrik modelleme amacıyla altı *spectral* analiz algoritmasından yararlanılmaktadır. Bunlar Çizim 3.19'da özetlenmiştir.



Çizim 3.19. Önemli spektral analiz algoritmaları

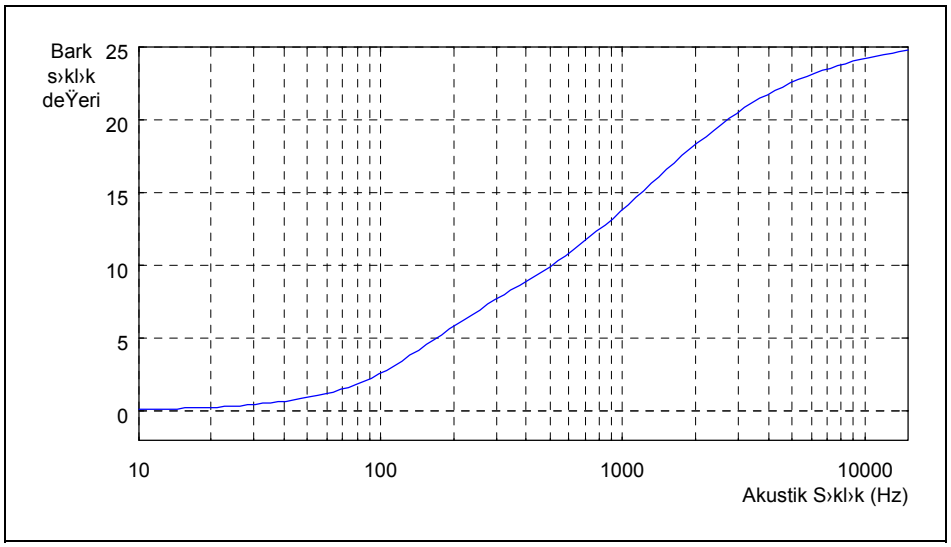
Bu algoritmalardan filtre dizisi (*filter bank*) kullanımı, sesli ifade tanımada özellik vektörü çıkarmada kullanılan eski bir yöntemdir. *Linear prediction* yöntemi ise 1970'li yılların başında ortaya çıkmıştır. *Linear prediction* ve *Fourier* dönüşümü yöntemleri günümüzde sesli ifade tanıma alanında oldukça yaygın biçimde kullanılan yöntemlerdir.

3.3.1. Sayısal Filtre Dizisi Tekniği *(Filter bank)*

Sesli ifade tanımada kullanılan öncü bir tekniktir. Gerçekleştirimi, başta örneksel devreler yardımıyla yapılmıştır. Yapısında kulağın çalışma ilkesi temel alınmıştır. Sesli ifadeyi oluşturan sıklık bölgeleri filtre dizileri ile ayrılmaya çalışılır. Her sıklık bölgesine ilişkin elde edilen değerler dizisi incelenen sinyali parametrik olarak tanımlamada ya da modellemede kullanılır. Yöntem tanımlanan her merkez sıklık değeri için değişmez band genişliğinde filtreler kullanılmasını gerektirir. Band genişlikleri, merkez sıklık değerinin yaklaşık %10 ila %20'si kadarı olarak seçilmektedir. Bu filtreler, *algılanan sıklık* değeri olarak daha önce sözü edilen *phons* değerlerine göre sıralanırlar. Filtrelerin merkez sıklıkları, *akustik sıklık* değerinden *(f), algılanan sıklık* değerine dönüşüm fonksiyonu ile bulunur. Bu bağlamda yaygın olarak, *bark* ve *mel* adlı iki fonksiyon kullanılmaktadır. Bunlardan *bark* sıklığını oluşturan fonksiyon aşağıda verilmiştir(Picone 1993).

$$Bark = 13 \cdot \arctan\left(\frac{0.76f}{1000}\right) + 3.5 \cdot \arctan\left(\frac{f^2}{(7500)^2}\right) \qquad (3.15)$$

Bark sıklığının akustik sıklığa göre değişimi Çizim 3.20'de verilmiştir.

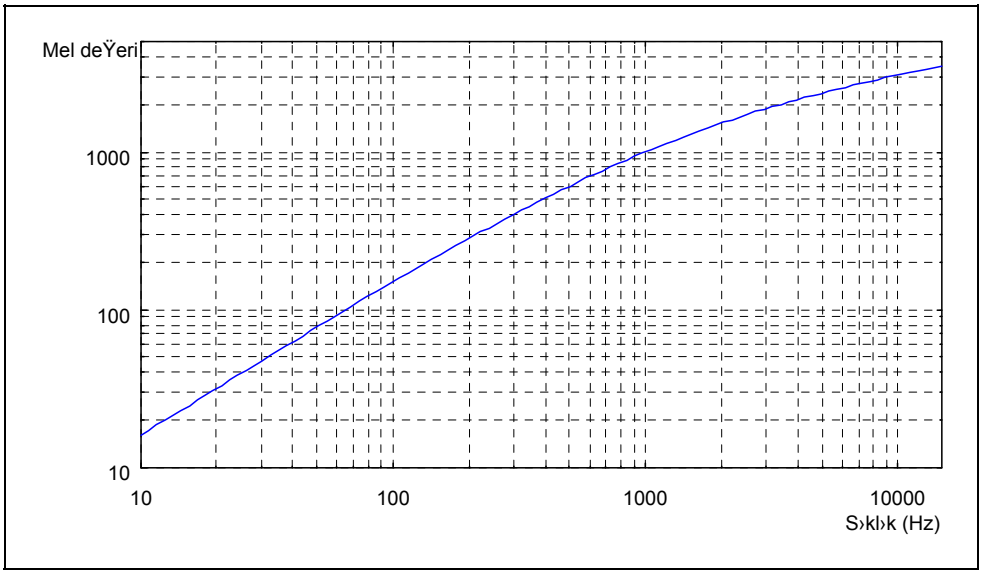


Çizim 3.20. Akustik ve *Bark* sıklık ilişkisi

Sesli ifade tanımada yaygın olarak kullanılan bir diğer fonksiyon da *mel* sıklığına dönüşüm fonksiyonudur.

$$mel = 2595 \log_{10}\left(1 + \frac{f}{700.0}\right) \quad (3.16)$$

Bu dönüşüm fonksiyonu 3.16'da verilen formül ile ifade edilmektedir. Çizim 3.21'de *mel* sıklığının akustik sıklığa göre değişimi verilmiştir. Bu dönüşüm fonksiyonunun özelliği, grafikten de görüleceği gibi, ilk 1kHz'lik kesimin doğrusal, 1kHz'in ötesinde ise logaritmik olmasıdır.

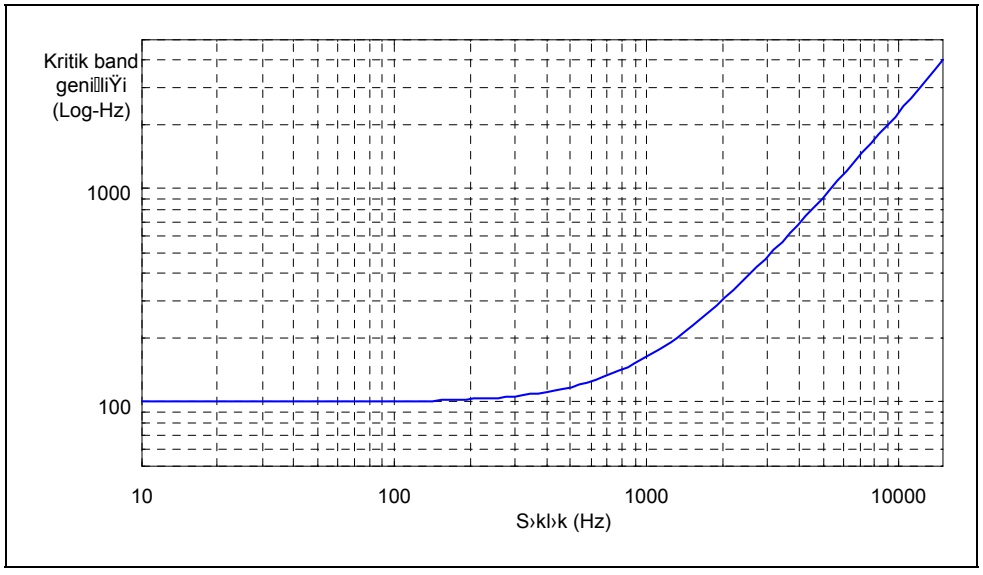


Çizim 3.21. Akustik ve *Mel* sıklık ilişkisi

Dizi içinde yer alan *band pass* filtrelerin *mel* ya da *bark* sıklıklarındaki bant genişlikleri:

$$BW_{CRITICAL} = 25 + 75\left[1 + 1.4\left(\frac{f}{1000}\right)\right]^{0.69} \qquad (3.17)$$

formülü ile bulunur. *Kritik band genişliği* olarak anılan bu değerlerin sıklık eksenine göre grafiği Çizim 3.22'de verilmiştir.



Çizim 3.22. *Mel* ve *bark* sıklıkları için Kritik band genişliği

3.3.2. *Fourier* Dönüşümü

Fourier dönüşümünde, periyodik bir sinyal olarak düşünülen sesli ifade sinyalini oluşturan *sinüssel* harmoniklerin çarpanları bulunmaktadır. Dönüşüm sonucu elde edilen çarpan değerleri sinyalin parametrik tanımında ya da modellenmesinde kullanılır. *Fourier* dönüşümünde sözkonusu olan harmonik sıklıkları, eşit band genişliğine sahip *band pass* filtre dizisine ilişkin merkez sıklıklar olarak da düşünülebilir. *Fourier* dönüşümünde sesli ifade sinyali, sıklık değerleri birbirinin katı bir dizi *sin* ve *cos* fonksiyonunun a_k ve b_k olarak anılan ağırlık katsayıları ile çarpılmış değerlerinin toplamı olarak ifade edilir. *Fourier* dönüşüm fonksiyonu aşağıda verilmiştir.

$$F(n) = \sum_{k=0}^{N-1} (a_k \cos(k\omega n) + b_k \sin(k\omega n)) \qquad (3.18)$$

Burada N incelenmesi istenen sıklık bölgesi (band) sayısıdır. $\omega = 2\pi f$ olup f, örneklem sıklığının $2 \cdot N$'ye bölünmesiyle elde edilir. Uygulamada N değeri, ikinin katları (2^5, 2^6, 2^7, 2^8) olarak alınır. a_k ve b_k katsayıları ise sinyali oluşturan *sin* ve *cos* bileşenlerinin sözkonusu sıklık bölgeleri için *Fourier* katsayılarıdır. Bu katsayıların hesaplanması aşağıdaki formüle göre gerçekleşir:

$$\begin{aligned} a_k &= \frac{2}{N} \sum_{n=0}^{N-1} F(n) \sin(k\omega n) \\ b_k &= \frac{2}{N} \sum_{n=0}^{N-1} F(n) \cos(k\omega n) \end{aligned} \qquad (3.19)$$

Kesikli fonksiyonlar için *Fourier* dönüşümü aşağıdaki biçimde de ifade edilebilir:

$$\begin{aligned} S(n) &= \sum_{k=0}^{N-1} s(k).e^{-j2\pi \frac{n}{N} k} \\ s(k) &= \frac{1}{N} \sum_{k=0}^{N-1} S(n).e^{j2\pi \frac{n}{N} k} \end{aligned} \qquad (3.20)$$

Burada n incelenen sıklık bandını, N ise band sayısını temsil etmektedir.

Fourier dönüşümünün sayısal sinyal işlemede kullanılan biçimi *Fast Fourier Transformation* olarak bilinir. Bu dönüşümde, incelenen sinyal alt kesimlere ayrılarak, dönüşüm işleminin aynı döngüde işlenebilecek kesimleri birleştirilmekte ve bu yolla dönüşüm hızlandırılmaktadır. Bu bağlamda kullanılan algoritma *butterfly* algoritması olarak anılmaktadır (Çizim 3.23). Çizim 3.23-a'daki W_N değeri *twiddle factor* olarak anılır ve

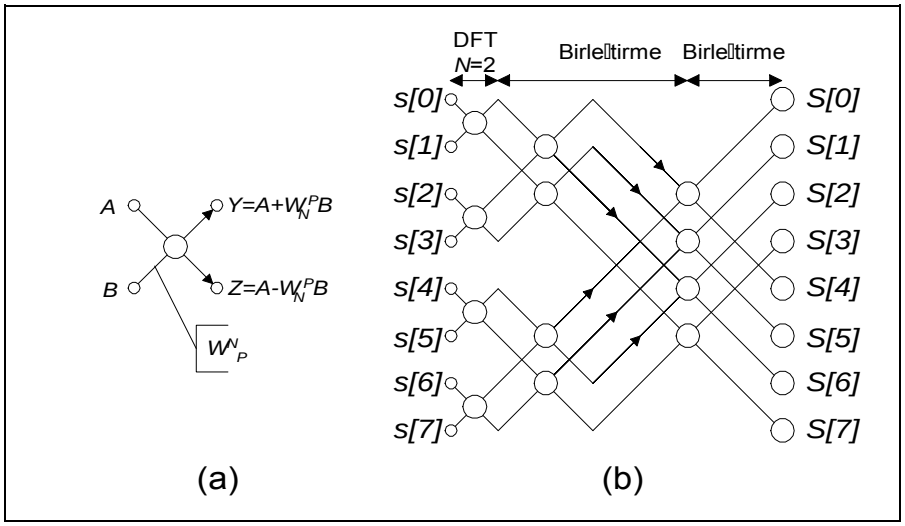
$$W_N = e^{-j(2\pi/N)} \text{ ya da}$$
$$= \cos\left(\frac{2\pi}{N}\right) + j\sin\left(\frac{2\pi}{N}\right) \quad (3.21)$$

formülü ile ifade edilir. W_N değerleri değişmez olup tüm hesaplamalarda kullanılmak üzere bir tabloda tutulabilir. İşlemlerin başında bir kez hesaplanması yeterlidir.

Bu durumda 3.20'de verilen formüller aşağıdaki biçime dönüşürler:

$$S(n) = \sum_{k=0}^{N-1} s(k) W^{-kn}$$
$$s(k) = \frac{1}{N} \sum_{n=0}^{N-1} S(n) W^{kn} \quad (3.22)$$

Fourier dönüşümü, sayısal sinyal işlemede en fazla zaman kaybının olduğu kesimlerden biridir. Orjinal dönüşüm algoritması kullanıldığında, N örnek sayısı için işlemler N^2 adımda tamamlanmaktadır. *Butterfly* algoritması ile aynı dönüşüm $N*\log_2(N)$ adımda gerçekleşmektedir. N sayısının büyümesi ile algoritmanın başarımı da ortaya çıkmaktadır. *Butterfly* algoritması yaygın olarak kullanılan bir algoritmadır. Bu bağlamda *butterfly* algoritmasına ilişkin kimi alt işlemler, sayısal sinyal işleyiciler üzerinde komut düzeyinde tanımlanmıştır. *Fourier* dönüşümünde giriş verisi örneklem sayısının ikinin katları sayılarda seçilmesi de *butterfly* algoritmasının kullanılmasından dolayıdır. *Butterfly* algoritmasının kullanıldığı *Fourier* dönüşümünde giriş verileri, 128, 256, 512 gibi ikinin üstel katı olan sayıdaki sözcük dizisinden oluşmalıdır. Aksi halde *zero padding* olarak adlandırılan ve dizinin boş kalan kısmının sıfır değeri ile doldurulması işlemi gerekmektedir.



Çizim 3.23. (a) *P*'inci FFT Butterfly düğümü, (b) Butterfly ile ayrıntılandırılmış 8 girişli FFT

3.3.3. *Linear Prediction* Katsayıları ile Modelleme

Sesli ifade sinyallerinin parametrik bir biçime dönüştürülmesinde yaygın biçimde kullanılan diğer bir yöntem *linear prediction* yöntemidir. 1970'li yıllarda kullanılmaya başlanan bu yöntem, sesli ifade sinyallerinin sıkıştırılarak saklanmasından tanınmasına kadar pekçok uygulamada kullanılmaktadır. Sinyal bu yöntemlerde *linear prediction* katsayıları olarak anılan katsayılarla *(a_{LP})* temsil edilir. *Linear prediction* yönteminde sesli ifade sinyalinin modellenmesinde, başka bir deyişle parametrik biçime dönüştürülmesinde *Linear Predictive Coding (LPC)* adlandırması altında değişik yöntemler kullanılır. Sözkonusu yöntemler(Rabiner 1978):

- *Covariance* yöntemi,
- *Autocorrelation* yöntemi,
- *Lattice* yöntemi,
- *Inverse filter* yöntemi,
- *Spectral estimation* yöntemi,
- *Maximum Likelihood* yöntemi,
- *Inner product* yöntemi

olarak sıralanabilir.

s(n) incelenen sinyalin *n* inci örneklemedeki değeri ise bu sinyal değeri daha önceki N_{LP} kadar örnekleme dayanılarak öngörülmeye (hesaplanmaya) çalışılır. Bu yaklaşım aşağıdaki biçimde ifade edilir:

$$s(n) = -\sum_{i=1}^{N_{LP}} a_{LP}(i)s(n-i) + e(n) \qquad (3.23)$$

Burada a_{LP} *linear prediction* katsayılarını *e(n)* ise *linear prediction* yoluyla yapılan öngörü hatasını göstermektedir. Başka bir anlatımla sinyal, N_{LP} kadar a_{LP} *linear prediction* katsayısı ile modellenmektedir. *e(n)* hesaplanan ve ölçülen sinyal değerleri arasındaki ayrımdır.

3.23 eşitliği, *z*-dönüşüm notasyonu kullanılarak yeniden yazılırsa aşağıdaki formül elde edilir:

$$E(z) = H_{LP}(z)S(z) \qquad (3.24)$$

Burada, *S(z)* ve *E(z)*, sesli ifade sinyalinin ve hesaplama hatasının *z*-dönüşümleridir. Bu formülde yer alan H_{LP}*(z) linear predictive inverse filter* olarak anılır ve aşağıdaki biçimde ifade edilir:

$$H_{LP}(z) = 1 + \sum_{i=1}^{N_{LP}} a_{LP}(i)z^{-i}$$

ya da (3.25)

$$H_{LP}(z) = \sum_{i=0}^{N_{LP}} a_{LP}(i)z^{-i}$$

Burada a_{LP}*(0)*=1 olarak kabul edilir.

a_{LP} katsayıları:

$$\bar{a}_{LP} = \mathbf{f}^{-1}\bar{\phi}$$
$$\bar{a}_{LP} = \left[a_{LP}(1), \cdots, a_{LP}(N_{LP})\right]^T dir. \qquad (3.26)$$

denklemiyle elde edilir. Burada

$$= \begin{bmatrix} \phi_n(1,1) & \phi_n(1,2) & \cdots & \phi_n(1,N_{LP}) \\ \phi_n(2,1) & \phi_n(2,2) & \cdots & \phi_n(2,N_{LP}) \\ \cdots & \cdots & \cdots & \cdots \\ \phi_n(N_{LP},1) & \phi_n(N_{LP},2) & \cdots & \phi_n(N_{LP},N_{LP}) \end{bmatrix}$$

$$= [\phi_n(1,0), \phi_n(2,0), \cdots, \phi_n(N_{LP},0)]^T \qquad (3.27)$$

ve

$$\phi_n(j,k) = \frac{1}{N}\sum_{m=0}^{N-1} s(n+m-j)s(n+m-k) \qquad (3.28)$$

Yukarıdaki tanımlanan yöntem *Covariance* yöntemi olarak bilinir. $\boldsymbol{\phi}$ *covariance* matrisidir. $\phi(j,k)$, $s(n)$ için *covariance* fonksiyonudur. Bu yöntem *pure least squares* yöntemi olarak da bilinir.

Sesli ifade tanıma kapsamında, yukarıda anılan yöntemlerden, en çok *autocorrelation* yöntemi kullanılır. Bu yöntemde $\phi_n(j,k) = \phi_n(0,|j-k|)$ olarak alınır. Bu durumda $\boldsymbol{\Phi}$ matrisinin elemanları aşağıdaki biçimde hesaplanır.

$$R_n(k) = \frac{1}{N}\sum_{m=0}^{N-1-k} s(n+m)s(n+m-k) \quad (3.29)$$

$R_n(k)$ *autocorrelation* fonksiyonu olarak bilinir. Sesli ifade sinyali, *linear prediction* kapsamında *autocorrelation* yöntemiyle modelleneceği zaman sinyal üzerinde pencereleme uygulanır. Uygulanan pencereleme, genelde *Hamming* pencerelemesidir. Bu bağlamda, *linear prediction* katsayılarının hesaplanmasında *Levinson-Durbin* olarak bilinen özyineli algoritma kullanılır.

3.3.4. *Cepstrum* Katsayıları ile Modelleme

Cepstrum katsayıları ile modellemede sözkonusu katsayılar, hem *Fourier* dönüşümü sonucu elde edilen değerlere hem de *linear prediction* katsayılarına dayalı olarak hesaplanır. İzleyen kesimde, bu hesaplamalar *Fourier* dönüşümüne dayalı olarak açıklanacaktır. Belirli bir tahrik sinyali ve bunun yansımalarının üstüste binmesinden oluşan sinyallerde, sinyalin *Fourier* dönüşümünün logaritmasının ters *Fourier* dönüşümü, yansımaların bulunduğu konumlarda tepe *(peak)* noktalar göstermektedir. Bir sinyalin *Fourier* dönüşümünün logaritmasının ters *Fourier* dönüşüm fonksiyonu

cepstrum olarak adlandırılmaktadır(Oppenheim 1989). Sesli ifade sinyalleri, bir tahrik titreşiminin ses yolu *(vocal tract)* içinde rezonansa uğraması, duraklatılması gibi nedenler sonucu elde edildiğinden *cepstrum* fonksiyonunun sesli ifade sinyallerini parametrelemede kullanılabileceği akla gelir. Bir sesli ifade sinyali *s(n)*

$$s(n) = g(n) \otimes v(n) \qquad (3.30)$$

biçiminde yazılabilir. Burada *g(n)* tahrik sinyalini, *v(n)* ise ses yolunun vuru tepkisini (*impulse response* fonksiyonunu), ⊗ ise *convolution* işlemini ifade eder. Bu durumda sesli ifade sinyalinin *Fourier* dönüşümü aşağıdaki gibi yazılır:

$$S(f) = G(f).V(f)$$

Bu dönüşümün logaritması:

$$\begin{aligned} \log(S(f)) &= \log(G(f) \cdot V(f)) \\ &= \log(G(f)) + \log(V(f)) \end{aligned} \qquad (3.31)$$

olarak yazılır. Bu durumda, logaritma evreninde, tahrik sinyali ile ses yolunun vuru tepkisinin toplanmış biçimde olduğu görülür. Ses yolunun vuru tepki fonksiyonu sesli ifadeye ilişkin özellikleri içerdiğinden bu toplamdan bu özelliklerin çıkarımı yoluna gidilebilir.

Cepstrum katsayıları aşağıdaki formülle ifade edilir:

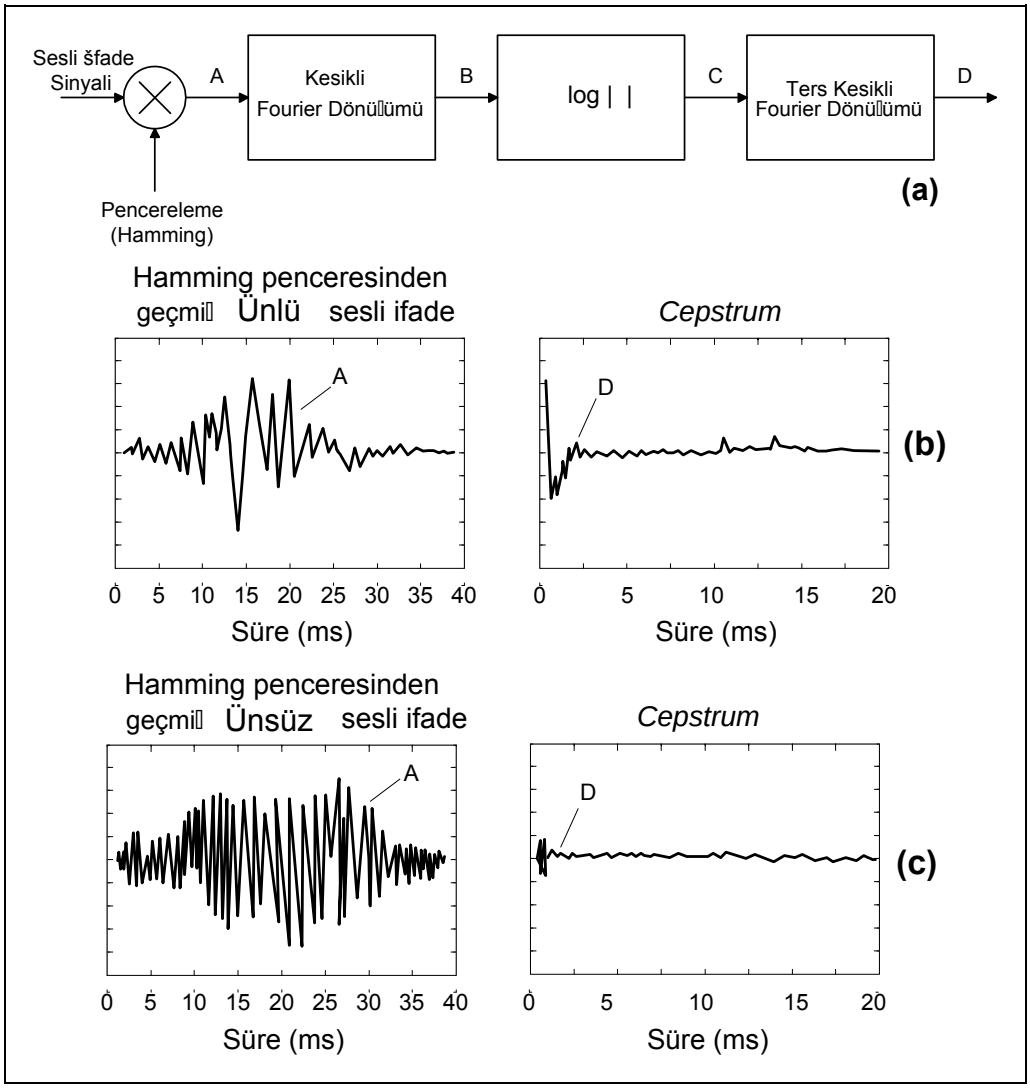
$$c(n) = \frac{1}{N} \sum_{k=0}^{N-1} \log_{10} |S(k)| e^{(2\pi/N)kn} \qquad 0 \le n \le N-1 \qquad (3.32)$$

c(0) sinyalin spektrum ortalamasını ifade eder. Log spektrum fonksiyonunun simetrik ve gerçel bir fonksiyon olduğu düşünüldüğünde *cepstrum* katsayısı fonksiyonu yalınlaşarak aşağıdaki biçimi alır:

$$c(n) = \frac{2}{N} \sum_{k=0}^{N} S(k) \cos(\frac{2\pi}{N} kn) \quad (3.33)$$

Yukarıdaki formülde *S(k)* incelenen sinyalin *Fourier* dönüşümünün *k* sıklığına ilişkin değeridir. *S(k)*'yı tek bir sıklık için hesap etmek yerine, *k*'yı merkez alan *M* sıklık değeri için hesaplamak ve elde edilen değerlerin ağırlıklı bir ortalamasını kullanmak da izlenen bir yoldur. Bunun yanı sıra *Fourier* dönüşümünün yapıldığı sıklık

değerlerini akustik eksen yerine *mel* ya da *bark* sıklık ekseninde ele almak da söz konusu edilir. Bu durumda, yukarıdaki formüllerde yer alan $S(k)$, $S_{ort}(k)$ olarak değiştirilir. Verilen açıklamalar çerçevesinde, $S_{ort}(k)$ özetle aşağıdaki gibi ifade edilebilir:



Çizim 3.24. Sesli ifadelerin *cepstrum* katsayılarının hesaplanması (Oppenheim 1989).

$$S_{ort}(k) = \frac{1}{M}\sum_{i=0}^{M-1} a_i . S_i(f(k)) \quad \forall i \;\; 0 \le a_i \le 1 \quad (3.34)$$

3.33'deki denkleme göre hesaplanan *cepstrum* parametreleri dizisindeki ilk değerler (*k*'nın küçük değerleri için hesaplanan parametreler) sesli ifade sinyalinin içinde, ses yoluna ilişkin özellik bilgilerini; *k*'nın büyük değerleri için hesaplanan *cepstrum* parametreleri ise tahrik titreşimlerinin periyoduna ilişkin bilgileri taşımaktadır (Picone 1993). Bu bilgilerden, sesli ifade tanıma kapsamında, ses yolunun özelliklerine, dolayısıyla sesli ifadelerin kendisine ilişkin olan özellik bilgileri önem taşımaktadır. Bu nedenle sesli ifadeler *cepstrum* katsayıları ile temsil edilirken genelde ilk 20 parametre göz önüne alınmaktadır. Çizim 3.24'de sesli ifade sinyalinin *cepstrum* parametrelerinin hesaplanmasında yer alan adımlar verilmiştir. Bu çizim üzerinde, sesin oluşum sürecine ilişkin, cepstrum fonksiyonu zaman ekseni üzerinde yaklaşık 1,2ms'lik süre içinde 20 parametre kullanılmaktadır.

Fourier dönüşüm değerlerine dayalı olarak hesaplanan *cepstrum* katsayıları genelde *Linear Frequency Cepstrum Coefficients (LFCC)* olarak anılır. Eğer *Fourier* dönüşüm değerleri hesaplanırken kullanılan sıklık değerleri mel sıklıkları olarak ele alınırsa, bu dönüşüm değerlerine göre hesaplanan *cepstrum* katsayıları, bu kez *Mel Frequency Cepstrum Coefficients (MFCC)* olarak adlandırılır. Daha önceden de belirtildiği gibi, *cepstrum* katsayıları *LPC* değerlerine dayalı olarak da hesaplanabilmektedir. Bu durumda hesaplanan *cepstrum* katsayıları *Linear Predictive Cepstrum Coefficients (LPCC)* olarak adlandırılır. Bunun için pencerelenmiş sinyal üzerinden *autocorrelation* ile genelde 10'uncu dereceden elde edilen *LPC* değerleri kullanılmaktadır. *LPCC* değerleri aşağıdaki formüle göre hesaplanmaktadır(Davis 1980):

$$LPCC(i) = LPC(i) + \sum_{k=1}^{i-1} \left(\frac{k-i}{i} \right) \cdot LPCC(i-k) \cdot LPC(k) \;\; i = 1,2,..,10 \quad (3.35)$$

4. SESLİ İFADE TANIMAYA GENEL BAKIŞ

İnsan-makina arasında, insan için en doğalı olan sesli iletişimin kullanılabilmesi, sesli ifadenin makina tarafından tanınabilmesini gerekli kılar. Son 15 yılda çeşitli sesli ifade tanıma stratejileri, farklı dillerde ve bu dillere özgü biçimlerde gerçekleştirilmeye çalışılmıştır. Sesli ifade tanıma; sinyal işleme, örüntü tanıma, yapay anlayış, bilgi ve olasılık kuramları, istatistik, bilgisayar bilimleri, psikoloji, dilbilim, biyoloji gibi değişik bilim dallarını ilgilendirmektedir. Disiplinler arası bir özellik gösteren sesli ifade tanıma konusu günümüzde hala araştırma aşamasındadır. Gerçekleştirilen kimi öncü sistemlerle elde edilen sonuçlar umut verici görülmektedir. Ancak bu sistemler konuşmacıya bağımlılık, sınırlı konuşma birimleri ve sözlük gibi kimi kısıtlamalarla kullanılmaktadır.

4.1. Sesli İfade Tanıma Sistemlerinin Genel bir Sınıflandırması

Sesli ifade tanıma sistemleri ya da sesli ifade tanıyıcılar, artan zorluk sırasında aşağıdaki gibi sıralanabilmektedir:

- Ayrışık sözcük tanıma sistemleri (*isolated word recognition systems*)
- Sözcük yakalama sistemleri *(word spotting systems)*
- Sürekli sözcük / sesli ifade tanıma sistemleri *(connected word / speech recognition systems)*

3 değişik kategoriye ayrılan bu sistemler aşağıda ayrı ayrı tanımlanmıştır.

Ayrışık Sözcük Tanıma Sistemleri (*Isolated Word Recognition Systems*)

Ayrışık sözcük tanıma, en yalın sesli ifade tanıma tekniği olup sözcük yakalama ve sürekli sesli ifade tanımaya temel oluşturmaktadır. Ayrışık sözcük tanımada sözcükler arasındaki duraklar belirgin olduğundan tanıma kolaylaşmaktadır. Zira sesli ifade tanımada en zor olanı, sözcüklerin başlangıç ve bitiş noktalarının belirlenebilmesidir. Bu yöntemde sözcükler birbirlerinden bağımsız olarak ele alındıklarından, birlikte seslendirme (*coarticulation*) sorunu bulunmamaktadır.

Sözcük yakalama sistemleri (*word spotting systems*)

Konuşma içinde aranan sözcüklerin yakalanmasını sağlayan bu tür tanıyıcılar sürekli sesli ifadeler üzerinde çalışmaktadır. *Time warping* ile dinamik programlama tekniklerinin sıkça kullanıldığı bu tür sistemlerde her sözcük bir şablon *(template)* ile ifade edilmektedir. Tanıma işlemi, aranan şablonun sesli ifade içinde çakıştığı bir örüntü arama biçiminde gerçekleşmektedir.

Sürekli Sözcük / Sesli İfade Tanıma Sistemleri (*Connected Word / Continuous Speech Recognition Systems*)

Sürekli sesli ifade tanıma, genelde iki biçimde ele alınmaktadır. Eğer sürekli sesli ifade içinde sözcük sınırları bulunarak sorun sözcük tanımaya indirgenebilirse sürekli sözcük tanımadan; fonem gibi, sözcüğe göre daha alt düzey birimler tabanında bir tanıma gerçekleştiriliyor ise sürekli sesli ifade tanımadan söz edilmektedir. Sürekli sesli ifade tanıma, ayrışık sözcük tanıma ve sözcük yakalamaya göre daha zordur.

Sürekli sesli ifade tanımadaki zorluk üç değişik nedenden kaynaklanmaktadır. Bunlardan ilki, sürekli sesli ifade içindeki sözcük sınırlarının belirgin olmamasıdır. Ayrışık sözcük tanımada, sözcük sınırları bilindiğinden başarım yüksek olmaktadır. Sürekli sesli ifadede, aranan sözcük sınırlarının bulunması zor, hatta kimi zaman olanaksızdır. Sözcük başına gelen /s/ sesinin çoğu kez kaybolmasının oluşturduğu sorun buna örnek olarak gösterilebilir. İkinci sorun, tanınacak ses birimlerinin, ön ve arkasına gelen diğer ses birimleri tarafından etkilenmeleri, birlikte seslendirilmelerinden kaynaklanan sorundur (*coarticulary effects*). Üçüncü sorun vurgulama, duraklama gibi bürünlerden (*prosodics*) kaynaklanmaktadır. Bürünler sözcüklerin söyleyişte kaybedilmesine neden olmaktadır. İsim, sıfat ve fiiller net ve yüksek enerjili seslerle ifade edilirken aradaki bağlaçlar ve kısa süreli sözcükler çoğu kez düşük enerjili olmakta ya da yutulmaktadır. Bu da sistemin, sesli ifadeleri doğru biçimde yakalamasını olumsuz yönde etkilemektedir.

Sürekli sesli ifade tanıma sözcük tabanında ele alınıyor ise, başka bir deyişle sürekli sözcük tanıma yaklaşımı kullanılıyor ise, sözcük yakalamada kullanılan *time warping* tekniğinin geliştirilmiş biçimlerinden yararlanılmaktadır. Sürekli sesli ifade tanıma, fonem gibi daha alt düzey ses birimlerine dayalı olarak da ele alınmaktadır. Bu bağlamda, sesli ifade önce fonemik çözümleme ve kesimlemeye *(segmentation)* tabi tutulmaktadır. İkinci bir aşamada ise fonem kümelemesi ile sözcük çakıştırma gerçekleştirilmektedir.

Sürekli sözcük tanımada, *time warping* tekniğinin geliştirilmiş biçimleri kullanılabilmektedir. Eğer sözcük yakalama algoritması sürekli konuşma içinde belirli sözcükleri yakalayabiliyorsa, daha geniş bir şablon sözlüğü kullanılarak sürekli sesli ifade içinde yer alan sözcükler bulunabilmektedir. Kısıtlı sayıda sözcüğün kullanıldığı sürekli sesli ifadelerde, bu yaklaşımın başarımı yüksek olmaktadır. Örneğin birkaç rakamın birlikte söylenmesinden oluşan, sayılara ilişkin sürekli sesli ifadelerin tanınmasında (*digit recognition*) başarılı sonuçlar alınabilmektedir.

Sürekli sesli ifade tanıma sistemlerinde bir diğer zorluk sesli ifadenin hızıdır. Bu ise birim zamanda söylenen sözcük sayısı ile belirlenir. *IBM* ve *Texas Instruments* Firmalarında, bu amaçla, sırasıyla 1100 ve 1529 İngilizce cümleden oluşan deneysel veri tabanları oluşturulmuştur. Bunlar üzerinde yapılan sözcük söyleme hız ölçümlerine ilişkin sonuçlar, ayrışık olarak seslendirilen sözcük hızları ile sürekli seslendirilen sözcük hızları arasında karşılaştırma yapılabilmeye olanak sağlayacak biçimde Çizelge 4.1'de verilmiştir(Lee 1989).

Çizelge 4.1. Ayrışık ve sürekli sesli ifadede sözcüklerin seslendirilme süreleri.

	Ayrışık Sözcükler	**Sürekli Sözcükler**
Ortalama Sözcük Süresi (s)	0.838	0.352
Standart Sapma	0.171	0.082
Dakikadaki Sözcük Sayısı	71.60	170.45

Fonem tabanlı sesli ifade tanıma sistemleri (*phoneme based recognition*)

Sesli ifade tanımada, sözcük yerine, fonem gibi daha alt ses birimlerine dayanılabilir. Ancak bu ses birimlerinin sayıca kısıtlı olması ve sözcükleri bunların kombinasyonu ile ifade edebilme özelliğinin bulunması gerekmektedir. Sözü edilen alt birimler hece, fonem ve ses *(phon)* olabilmektedir. Sözcükler, herbiri kendine özgü özelliği bulunan ve çok sayıda ögelerdir. Sözcük biriminin tanımaya taban oluşturması durumunda, çok geniş ve hızlı erişimli sözlüklerin kullanımı gerekmektedir. Bu nedenle, sözcüğe göre daha alt birimlerin kullanılması tercih edilebilmektedir. Ancak alt birimlerin kullanılması daha anlamlı görünmesine karşın, kimi durumlarda sözcük tanımaya göre daha karmaşık yapıları da gerektirmektedir. Bu bağlamda, ses birim sınırlarının belirlenmesi ve birlikte söyleme etkilerinin göz önüne alınması gibi

sorunlar ortaya çıkmaktadır. Fonem tabanlı tanıma işlemlerinde önce alt ses birimleri tanınmaya çalışılır daha sonra bu birimler birleştirilerek sözcükler oluşturulur.

Fonem tabanlı sesli ifade tanıma sistemleri iki alt kesimden oluşabilir. İlk kesimde sesli ifadelerin, fonem gibi kesimlere ayrılma (*segmentation*) işlemi yerine getirilir. Kesimlere ayırmada:

- ses sinyal genliği ya da enerjisi,
- ünlüler,
- sıfırdan geçiş oranı (*zero-crossing rate)*,
- perde *(pitch)*,
- fonetik *(f_1/f_2)* ya da spektral değişimler

bilgi olarak, birbirlerini destekler biçimde çoğu kez birlikte kullanılmaktadır.

Fonem tabanlı sesli ifade tanıma sistemlerinde ikinci alt kesim sözcük yakalama kesimdir. İlk kesimin sağladığı fonetik verilerden, fonetik ve sözdizim kuralları ile bir sözlükten yararlanılarak sözcükler elde edilir. Sözdizim kuralları, sözcüklerin ve fonemlerin geliş sırasını belirler.

4.2. Sesli İfade Tanımayı Etkileyen Faktörler

Sesli ifade tanıma sistemlerinin başarımları bir dizi faktöre bağlıdır. Bu faktörler genelde konuşmanın geçtiği ya da kaydedildiği ortamla ilgili olup kısaca şunlardır (Parson 1986):

a. Ses sinyali, fonetik bilginin yanı sıra konuşmacılara özgü özellikleri de içermektedir. Hiç bir konuşmacının sesi bir diğerine benzememekte, sesli ifadelerde *loci* olarak bilinen *f1/f2* ünlü formant sıklık oranları, konuşmacıdan konuşmacıya değişiklikler göstermektedir. Bu nedenle, ses, fonem, hece, sözcük gibi ses birimlerinin ayrıştırılmaları güçleşmektedir. Bu özellik *konuşmacı bağımlığı* olarak bilinmektedir.

b. Sesli ifade tanıma, tek bir konuşmacıya bağlı olarak ele alınsa bile, aynı konuşmacının sesli ifadesi çeşitli nedenlerle değişiklik gösterebilmektedir. Bu nedenler aşağıdaki gibi özetlenebilir:

- Konuşmacılar çoğu kez sözcükleri tane tane telaffuz etmemektedir. Bu durum bilinçli konuşmacılar için bile geçerli olabilmektedir. Sözcüklerdeki kimi sesler, kimi zaman konuşmacı tarafından yutulmakta, yeteri netlikte telaffuz edilmemektedir. Heceler arası duraklar da her zaman aynı özellik ve netlikte olmayabilmektedir. Dikkatli telaffuz, sorunu hafifletmekte ancak tümüyle çözememektedir. Bu sorun *telaffuz* sorunu olarak bilinmektedir.

- Tek bir konuşmacıya ilişkin ünlü formant sıklıkları, *loci* ve formant geçiş süreleri zaman içinde değişiklik gösterebilmektedir. Konuşma örneklerini zaman içinde yineleyerek sorun hafifletilebilmektedir. Bu sorun *zaman içinde fonetik değişim* sorunu olarak bilinmektedir.

- Fonemler, çoğu kez sesli ifadeler içinde yeraldıkları konumlara göre değişik fonetik özellikler göstermektedir. Fonemlerin, sesli ifadelerde bulundukları konumlara göre değişik seslendirilmeleri *birlikte seslendirme (coarticulation)* sorunu olarak bilinmektedir.

- Sözcüklerin seslendirilme süreleri ve sözcüğü oluşturan alt parçaların zamanlaması değişebilmektedir. Bu nedenle, tanıma süreci, zamana bağımlı değişkenleri ele alacak biçimde tasarlanmak zorundadır.

c. Akustik değişkenlerin birebir fonemik değişkenlere karşılık getirilememesi, sesli ifadelerin tanınmasında çelişkili durumlar yaratabilmektedir. İnsan beyni çelişkili durumlardan, konuşulan dile ve konuya ilişkin bilgileri kullanarak çıkmaktadır. Kişi tarafından daha önce duyulmamış yabancı özel isimlerin tam olarak anlaşılamaması bunun belirgin bir örneğini oluşturmaktadır. Bu gözlem sesli ifade tanımada kullanılacak ses birimlerinin seçiminin ve bunlardan üretilecek özellik vektörlerinin önemini ortaya koymaktadır.

d. Gürültü ve girişim, sesli ifade tanımayı güçleştiren diğer etkenlerdendir. İnsanda duyma yeteneği, sesli ifadelerin, zayıf im/gürültü oranları için doğru tanınmasını başarabilecek biçimde gelişmiştir. Bu özellik, büyük oranda iki kulağın varlığından kaynaklanmaktadır. Sesli ifade tanımanın başarımı gürültü düzeyine bağlıdır. Geniş bantlı gürültüler zayıf frikatifleri maskelemektedir. Bunun sonucu olarak, ileride açıklanacak *LPC* gibi tekniklerde doğru olmayan parametre kestirimlerine gidilmekte ve sözcüğün başı ve sonunun belirlenmesi zorlaşmaktadır. Sesli ifade tanıma sistemleri gürültüden elverdiğince arındırılmış

(filtrelenmiş) konuşmalar için gerçekleştirilmektedir. Bu amaçla, çoğu kez sesli ifadeler gürültüye karşı yalıtılmış odalarda kaydedilmekte, kayıtlar filtrelendikten sonra kullanılmaktadır.

4.3. Sesli İfade Tanıma Sistemlerine ilişkin kimi Özellikler

Sesli ifade tanıma sistemleri, türleri ne olursa olsun konuşmacıdan bağımsızlık, ses sinyal niteliği, öğrenme yeteneği, sözlük büyüklüğü, dilbilgisi kullanımı gibi özelliklere sahip olmaktadır. Bu özellikler aşağıda özetlenmiştir:

4.3.1. Konuşmacıdan Bağımsızlık

Sesli ifade tanıma sisteminin işlediği sesli ifadelerin rasgele konuşmacılardan gelebilmesi durumunda konuşmacıdan bağımsızlık özelliğinden söz edilir. Konuşmacıdan bağımsızlık, sağlanması zor ancak önemli bir özelliktir. Genelde sesli ifadeler konuşmacıya bağımlı ögeler içermekte ve bir konuşmacıdan diğerine önemli farklılıklar göstermektedir. Bu farklılıkların tanımayı etkilemeyecek biçimde ortadan kaldırılması konuşmacıdan bağımsızlığı sağlamaktadır. Sesli ifade tanıma sistemlerinde denenen konuşmacı sayısı önemli bir parametre olup tek, sınırlı ve sınırsız biçimde belirlenmektedir.

Konuşmacı bağımsızlığını sağlamak için genelde üç değişik yaklaşım kullanılmaktadır. Birinci yaklaşım, sesli ifadelerdeki konuşmacıya bağımlı ögeleri ortadan kaldıracak *anlayışlı ön işleme birimi* kullanmayı gerektirmektedir. Bu yaklaşımla konuşmacıdan bağımsızlığın sağlanabilmesi için sistemde *uzman spektrogram* kesimi yer almalı ve spektrogramların incelenmesi yüksek duyarlılıkta gerçekleştirilmelidir. İşlem öncesinde bir uzmana danışılarak bu spectrogramlar üzerinde değişmez ortak parametreler saptanmaya çalışılmalıdır. Bu tür parametreler bulunabilirse konuşmacıdan bağımsız tanıma, konuşmacıya bağımlı tanıma kadar kolaylaşır(Parson 1986).

İkinci yaklaşım, konuşmacılar arasındaki farklılıkları yakalayacak gösterim biçimlerini kullanmaktır. Bu bağlamda, konuşmacıların ses özelliklerinin kümelenmesi gerçekleştirilmektedir. Uygulamada sistem sözlüğünde bulunan her sözcük birden fazla konuşmacı tarafından seslendirilmektedir.

Üçüncü yaklaşımda ise konuşmacıya uyum sağlama yöntemi kullanılmaktadır. Yöntemin çalışma ilkesi, daha önceden belirlenmiş parametrelerin yeni konuşmacıdan gelenlerle farklarının bulunması ilkesine dayanır. Sistemde her yeni konuşmacıyla birlikte sözcük parametrelerin değiştirilmesi gerekmektedir. Sonuçta örnekler kümelere ayrılıp her küme için bir örnek küme vektörü üretilmektedir. Ancak sözcük ve konuşmacı sayısının artırılması sistemin genel başarımının düşmesine ve karmaşıklığın artmasına neden olmaktadır. Sistemin konuşmacıya uyumu sağlamak için gereksediği işlem sürelerinin büyüklüğü uygulama alanlarını kısıtlamaktadır. Bu son yaklaşımı kullanan sistemlerde, sistem sözlüğü yeterince küçük tutulursa konuşmacılar arasındaki farklılıklar daha kolay aşılabilmektedir. Aksi halde her konuşmacı ve sesli ifade örneği için bir örüntü oluşmaya başlayacağından sistemin genel başarımı düşmektedir. Bu bağlamda, Sambur ve Rabiner (1975) çalışmalarında sözcük sayısının azaltılmasını fonetik sınıflandırmayı kullanarak gerçekleştirilmişlerdir. Bu kapsamda ses birimleri 6 alt sınıfa ayrılmıştır. Bunlar: Ön Ünlü, Orta Ünlü, Arka Ünlü, Ötüşümlü Ünsüz (*Vowel-like*), Gürültü benzeri Ötüşümlü Ünlü (*Noise-like voiced*), Gürültü benzeri Ötüşümsüz Ünsüz (*Noise-like unvoiced*) dür.

Konuşmacı bağımsızlığını sağlamakta kullanılan yaklaşım ne olursa olsun sistemde alıştırma *(training)* sürecine gerek duyulur. Alıştırma süreci, genelde sözcük tanıma alıştırmalarının çok sayıda yinelenmesi yoluyla gerçekleştirilmektedir. Alıştırma süreci zamana mal olması nedeniyle yüksek işlem kapasiteleri ve bellek kullanımı gerektirmektedir. Levinson (1977), incelediği konuşmacıdan bağımsız sistemlerin başarı oranlarının yaklaşık %65, konuşmacıya bağımlıların ise %88 olduğunu belirtmektedir. Hata oranı açısından bakıldığında da, konuşmacıdan bağımsız sistemlerde hata oranının, konuşmacıya bağımlılara göre 3-5 kat daha fazla olduğu gözlenmektedir (Lee 1989). Bu zorluklar nedeniyle, çoğu kez, sesli ifade tanıma sistemleri, ilk aşamada konuşmacıya bağımlı olarak ele alınmaktadır.

4.3.2. İncelenen Ses Sinyalinin Niteliği

Sesli ifadelerin kayıt edildiği ortam ile kayıt koşullarının nitelikleri sesli ifade tanıma sürecini etkileyen önemli bir özelliktir. Sesli ifadelerin kayıt ortamı:

- Özel yalıtılmış, yansımasız oda ya da
- herhangi bir ortam

olabilir. Rasgele ortamlarda kaydedilmiş sesli ifadelerin tanınması, ortamdan kaynaklanan gürültü nedeniyle daha zordur. Bu bağlamda, *bilgisayar odası* ortamı da, içerdiği gürültünün özelliği nedeniyle ayrı bir kategori kayıt ortamı olarak düşünülebilmektedir. Sesli ifadenin kayıt ediliş biçimi ve kayıt kalitesi de, en az sesli ifadenin kayıt ortamı kadar önemlidir. Kayıtta kullanılan araçlar:

- Yüksek kaliteli elektronik donanımlar (mikrofon ve yükseltici donanımlar),
- Orta kaliteli donanımlar,
- Telefon gibi özel amaçlı ve koşulların zorlandığı donanımlar

olarak sınıflandırılmaktadır.

4.3.3. Alıştırma Gereği

Sesli ifade tanıma sistemlerinde alıştırma *(training)* sürecine gerek duyuluyor olup olmaması da, bu sistemleri sınıflandırmaya yarayan diğer bir özelliği oluşturmaktadır. Sesli ifade tanıma sistemleri:

- alıştırma aşaması gerektirmeyenler *(without training),*
- alıştırmanın başta bir kez uygulandığı sistemler *(fixed training)*,
- alıştırma sürecinin tanıma süreci ile içiçe olduğu ve yeni koşullara uyum sağlayan sistemler *(continuous training)*

olarak sınıflanabilmektedir. Alıştırmanın sisteme başta bir kez uygulandığı şıkta, alıştırma, sistemin kurulma aşamasında gerçekleştirilir. Örneğin, sesli ifadeye dayalı, konuşmacıdan bağımsız rezervasyon sistemlerinde, bu kategoride anılan yaklaşım kullanılır. Bu tür sistemlerin alıştırma aşaması çok sayıda örnek kullanılarak oldukça uzun sürelerde gerçekleştirilmektedir.

4.3.4. Sesli İfadelerin Niteliği

Sesli ifade tanıma sistemlerinde, sesli ifadeyi oluşturan konuşmanın niteliği de önemli bir parametredir. Yazılı metinlere bağlı kalınarak yapılan (okuma biçiminde) konuşmalara ilişkin sesli ifadelerin tanınması daha kolay başarılmaktadır. Bu amaçla bazı araştırmacılar sistemlerinin denenmesinde radyo ya da televizyondaki haber programlarını kullanmaktadırlar. Sesli ifade tanıma sistemlerinin sınanması ve elde edilen sonuçların karşılaştırılabilmesi için ortak veri tabanları da kullanılmaktadır.

İngilizce için gerçekleştirilmiş ve yaygın olarak kullanılan kimi veri tabanları *Compact Disc*'ler üzerinde sayısal ses sinyalleri biçimde araştırmacılara sunulmaktadır. *TIRIM Texas Instruments ve CMU (Carnegie Mellon University)*; *TIMIT* ise yine *Texas Instruments* ve *MIT (Massachusetts Institute of Technology)* işbirliği ile gerçekleştirilmiş, bu tür, bilinen ses veri tabanlarıdır.

4.3.5. Sözlük Büyüklüğü

Sesli ifade tanıma sistemleri için, tanınabilen ayrışık sözcük sayısı sistemin önemli bir özelliğini oluşturur. Bir sistemce tanınabilen sözcüklerin oluşturduğu küme sistem sözlüğü ya da kısaca *sözlük* olarak anılır. Sözcük altı ses birimleri tabanında tanıma yapan sistemlerde sözlük, *ses takımı (phone set)* olarak anılmaktadır. Sözlük büyüklüğü, genelde başarımı olumsuz yönde etkilemektedir. Sözlüğün büyümesi işlem gücü gereksinimini ve hata oranını arttırmaktadır. Sözlük büyüklüğü:

- küçük boy,
- orta boy,
- büyük boy ve
- çok büyük boy

olarak sınıflanmaktadır. 100'den daha az sözcük içerenler küçük boy, 100-1000 arasında sözcük içerenler orta boy, 1000 sözcükten daha fazla sözcük içerenler büyük boy ve 10000 den fazla sözcük içerenler ise çok büyük boy sözlüklü sesli ifade tanıma sistemi olarak tanımlanmaktadır(Lee 1989).

4.3.6. Dilbilgisi Kullanımı

Sesli ifade tanımada, ilgili dilin dilbilgisinden yararlanmak da sözkonusu edilebilmektedir. Bu durumda, sözcükler, dilbilgisi kurallarının getirdiği ek sınırlamalardan yaralanılarak daha az ögeli kümeler içinde aranmaktadır. Başarımı olumlu yönde etkilemesine karşın, bu yaklaşım sistemin genel işlem yükünü arttırmaktadır. Dilbilgisi kullanımına dayalı sistemlerin dilbilgisi karmaşıklığı *perplexity* adlı özel bir parametre ile ölçülmektedir (Lee 1989).

5. SESLİ İFADE TANIMADA KULLANILAN YÖNTEMLER

Özellikleri ve kullandıkları yaklaşımlar birbirinden farklı birçok sesli ifade tanıma yöntemi geliştirilmiştir. Çizelge 5.1'de günümüze değin geliştirilen belli başlı sesli ifade tanıyıcı sistemler ve kullandıkları teknikler özetlenmiştir (Picone 1993). Çizelge dört ana sütundan oluşmaktadır. İlk sütun sesli ifade sisteminin geliştirildiği yer (üniversite, araştırma labaratuvarı) ve sistemin büyüklüğü hakkında bilgi vermektedir. Büyüklük iki kesimde verilmiştir. İlk kesim sesli ifade tanıma sisteminin sözlük kapasitesi hakkında bilgi vermektedir. İkincisi ise kullanıldığı ortamın özelliklerini yansıtmaktadır. İkinci sütun, sesli ifade sinyallerinin kaydına ve önişlem kesimine ilişkin bilgileri içermektedir. Üçüncü sütunda özellik vektörlerini hesaplamada kullanılan yöntemler verilmiştir. Son sütun ise tanıma aşamasında kullanılan model ve sesli ifade tanıma yöntemine ayrılmıştır. Çizelgede kullanılan kimi kısaltmalar çizelgeyi izleyen kesimde açıklanmıştır. Çizelgeden de görüleceği üzere sesli ifade tanıyıcılarında belli tek bir yöntem kullanmak yerine birden fazla yöntem bir arada kullanılmaktadır. Zira tek bir yöntem her şıkta her zaman tatmin edici sonuç üretmemektedir. Günümüzde *DSP* ve benzeri donanımların göreli ucuzlaması bu yola gidebilmeyi olanaklı kılmıştır.

Sesli ifade tanımada kullanılan yöntemler üç değişik yaklaşım çerçevesinde düşünülebilir. Bunlar:

- *Hidden Markov Model,*
- *Time Warping* ve
- Nöron Ağı

yaklaşımlarıdır. Bu yaklaşımlar, izleyen kesimde özetlenmiştir.

5.1. *Hidden Markov Model*

Hidden Markov Model (HMM), ilk olarak 1970 yılında ortaya atılmış (Rabiner 1989) kısa bir süre içinde de sesli ifade tanımada birbirlerinden bağımsız olarak *CMU (Carnegie Mellon University)* ve *IBM*'deki gruplarca kullanılmıştır. *HMM* birçok araştırmacı tarafından 1970'li yıllardan günümüze kadar, yaygın biçimde kullanılmaktadır. Bu modelin ilk türleri sözcük tanımada *(isolated word recognition)* kullanılmıştır.

Çizelge 5.1. Gerçekleştirilmiş sesli ifade tanıyıcıları ve parametreleri (Picone 1993).

Genel Bilgi		Sinyal Ölçümleri					Parametreler	İstatistiksel Model	
Referans	**Büyüklüğü**	**f_s kHz**	**a_{pre}**	**Frame ms**	**Pencere ms**	**Spectral Analiz**	**Parametre Türleri**	**Model**	**Tanıma yöntemi**
ATR	Büyük-Ofis	12	-0.97	3	21.3	LP(12) Power Mel FB, D-FB, D-D-FB	LP Power Mel FB, D-FB, D-D-FB	PWLR VQ(128)	DD-HMM
ATR	Büyük-Ofis	12	0	5	21.3	FFT(128)		-	TD-NN
AT&T	Küçük Telecom	6.67	-0.95	15	45	LP(8) Cep(12) D-Cep D-D-Cep Power D-Power D-D-Power,	Liftered-Cep D-Cep D-D-Cep Power D-Power D-D-Power,	Variance	CD-HMM
AT&T	Orta-Ofis	8	-0.95	10	30	LP(10) Cep(12)	(Benzer)	Variance	CD-HMM
BBN	Büyük-Ofis	20	0.0	10	20	FFT(512) Cep(14)	Mel-Cep D-Cep Power D-Power	VQ	DD-HMM
Brown	Küçük-Ofis	16	0.0	10	40	LP(12) Cep(12)	Cep D-Cep Power D-Power	MS-VQ (256/stage)	HMM-NN
Cambridge	Büyük-Ofis	10	FD	10	10	FFT(128)	FB	-	NN
CMU	Büyük-Ofis	16	-0.97	10	20	LP(14) Cep(12) D-Cep Power D-Power,	BT Cep D-Cep Power D-Power, Mel-FB D-FB D-D-FB	MS-VQ (256/stage)	DD-HMM
CMU	Büyük-Ofis	16	-0.97	10	20	FFT(256)		MS-VQ (256/stage)	TD-NN
CSELT	Orta Telecom	16	0.0	10	20	FFT(256) Cep(12) D-Cep Power D-Power,	Mel-Cep D-Cep Power D-Power, Mel-Cep	Variace MS-VQ	CD-HMM DD-HMM
CSELT	Büyük-Ofis	12	0.0	10	20	FFT(256) Cep(18)		VQ (128)	DD-HMM
Fujitsu	Büyük-Ofis	16	0.0	5	32	FFT(512)	Mel-FB Power	Identity	DP
IBM	Büyük-Ofis	16	0.0	10	20	-	Auditory Model(20)	-	DD-HMM

INRS INRS	Büyük-Ofis	16 16	0.0 0.0	10 10	25.6 25.6	FFT(256) FFT(256)	Mel-Cep D-Cep Power, (Benzer)	MS-VQ (64/stage) Variance	CD-HMM DD-HMM
KAIST	Büyük-Ofis	10	0.0	10	25.6	LP(12) Cep(12)	Cep D-Ceps	MS-VQ (256/stage)	DD-HMM
LL	Orta/Büyük Ofis/Askeri	8	FD	10	20	FFT(256) Cep(14)	FFT(256) Cep(14)	Fixed	CD-HMM
MIT	Büyük-Ofis	16	FD	5	-	FB(40)	FB(40)	PT	CD-CFG
Mitsubishi	Büyük-Ofis	10	-0.95	10	25.6	FFT(256)	FFT(256)	Variance	CD-HMM
NEC	Büyük Ofis	16	0.0	5	32	FFT(512) Cep(10)	FFT(512) Cep(10)	Variance	CD-HMM
NYNEX	Küçük Telecom	8	0.95	5	20	LP(10) Cep(10)	LP(10) Cep(10)	PT	HMM MLP-NN
NTT NTT	Büyük-Ofis Büyük-Ofis	12 12	0.0 0.0	10 8	30 32	LP(16) Cep(16), LP(16) Cep(16)	LP(16) Cep(16), LP(16) Cep(16)	Variance Variance	DD-FSA CD-HMM
Panasonic	Büyük-Ofis	10.6	0.0	9.3	18.6	PLP(8)	PLP(8)	Fixed	CD-HMM
Philips	Büyük-Ofis	16	0.0	10	25	FFT(512)	FFT(512)	MS-VQ	DD-HMM
RSRE	Büyük Ofis/Askeri	20	0.0	10	-	FB(27)	FB(27)	Fixed	CD-HMM
SRI	Büyük-Ofis	16	0.0	8	16	FFT(256)	FFT(256)	MS-VQ	DD-CSG
SSI	Büyük-Ofis	16	0.0	6.6	-	FB(20)	FB(20)	-	CD-HMM
TI	Küçük Telecom	8	-1.0	20	30	LP(10)	LP(10)	PT	CD-HMM
Tohoku	Büyük-Ofis	16	0.0	10	-	FB(29)	FB(29)	-	LVQ2-NN
Waseda	Büyük-Ofis	12	0.0	10	20	LP(16)	LP(16)	MS-VQ (256/stage)	DD-HMM

Küçük: **Küçük boyutlu sözlük (Rakkam ya da tek karakter tanıyıcılar)**

Orta: Orta büyüklükteki sözlük (5000 sözcükten az)

Büyük: Büyük sözlüklü sistemler (5000 sözcükten fazla)

Ofis: Gürültü düzeyi yaklaşık 70dB.

Telecom: Klasik telefon altyapısı

Askeri: Askeri amaçlı, çevre koşulları özelleşmiş.

FD: Sıklık Evreni (*FrequencyDomain*)

LP: *Linear Prediction*

PLP: *Perceptually-Motivated Linear Prediction*

FFT: *Fast Fourier Transform*

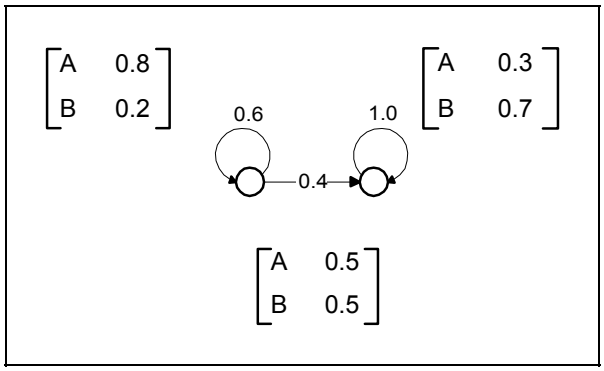
FB: *Filter Bank*

Cep.: *Cepstral* Parametreler

Power:	Sinyalin Gücü (dB)
Mel:	*Mel* skalasında parametreler
Bark:	*Bark* skalasında parametreler
Liftered:	*Liftered* parametreler
BT:	*Mel* skalasında parametreler, ikili dönüşüm kullanılmış (*Bilinear Transform*)
D-FB:	*Delta Filter Bank*
D-D-FB:	*Delta-Delta filter bank* parametreleri
D-Power:	Delta-Güç parametreleri
D-D-Power:	*Delta-delta power*
PWLR:	*Perceptually Weighed log likelihood distance measure*
VQ:	Vektör Kuantizasyonu
MS-VQ:	*Multi Stage* Vektör Kuantizasyonu(Birden fazla *codebook*)
PT:	*Prewhitening* dönüşüm
Variance:	*Varians*-ağırlık parametreleri
Identity:	*Identity* matris ağırlık parametreleri
Fixed:	Değişmez ağırlık matrisi (*pooled, grand* olarak ta adlandırılır)
HMM:	*Hidden Markov Model*
NN:	Nöron Ağları
TD-NN:	*Time Delay Neural Network*
DP:	Dinamik programlama
CD-HMM:	Sürekli yoğunluklu *Hidden Markov Model*
DD-HMM:	Kesikli yoğunluklu *Hidden Markov Model*
FSA:	Sonlu durum otomasyonu
CFG:	Kapsam bağımlı sözdizim
LVQ2:	*Learning Vector Quantizer* (Nöron Ağı)
MLP:	Çok Katmanlı Algılayıcı modeli (Nöron Ağı)

Hidden Markov modeli, doğal konuşma üretiminin stokastik modeli olarak tanımlanabilir. *Hidden Markov* modelinde sesli ifade üretiminin sonlu durumlu bir sistem tarafından üretildiği varsayılır. Bu sistemde her durumun, sonlu sayıda sesli ifade çıktısı üretebileceği düşünülür. Sesli ifade üretilirken, bu sonlu durumlu sistemin bir durumdan diğerine geçtiği ve her durumda belirli bir sesin üretildiği varsayılır. Bir durumdan diğerine geçiş olasılığı geçiş ağırlığı olarak bilinir. *Hidden Markov* modeline göre yürütülen sesli ifade tanıma sistemlerinde her sözcüğün bir sonlu durumlu sistem modeli vardır. *Hidden Markov* modelinde bir sözcüğe ilişkin ardadda gelen özellik vektörleri, tanıma sonlanaan değin bir durumdan diğer duruma geçişleri sağlamaktadır. Çizim 5.1'de *A* ve *B* biçiminde iki çıkışı bulunan basit bir

Markov modelinin çizimi verilmiştir. Çizimde her durum daire, geçişler ve olasılıklar ise oklarla gösterilmektedir (Lee 1989).



Çizim 5.1. Basit bir HMM örneği

Bir *Hidden Markov* modelinde:

- $\{s\}$: durum takımını tanımlar. Bunlardan S_I başlangıç durumunu S_F ise son durumu ifade eder.
- $\{a_{ij}\}$: i durumundan j durumuna geçiş olasılığıdır.
- $\{b_{ij}(k)\}$: Çıkış olasılığı matrisidir. k simgesinin i durumundan j durumuna geçiş sırasında üretilmesinin olasılığını gösterir.

a ve b olasılıksal değerler olmaları itibarıyla aşağıdaki özellikleri sağlamak durumundadırlar:

$$\begin{aligned} &a_{ij} \geq 0, \quad b_{ij}(k) \geq 0, \quad \forall i,j,k \\ &\sum_j a_{ij} = 1 \quad \forall i \\ &\sum_k b_{ij}(k) = 1 \quad \forall i,j \end{aligned} \qquad (5.1)$$

a ve b yeniden yazılsa:

$$\begin{aligned} a_{ij} &= P(X_{t+1} = j \,|\, X_t = i) \\ b_{ij}(k) &= P(Y_t = k \,|\, X_t = i, X_{t+1} = j) \end{aligned} \qquad (5.2)$$

elde edilir. Burada $X_t=j$ ile t zamanında j durumu ve $Y_t=k$ ile t zamanındaki k çıkış sembolünü göstermektedir. X ve Y *HMM* tarafından üretilmektedir. Ancak Y değeri çıkıştan gözlenebilmesine karşın X değeri gizlidir, gözlenemez. *First-order* gizli Markov modelinde iki varsayım yapılır. Bunlardan birincisi *Markov* varsayımıdır:

$$P(X_{t+1} = x_{t+1} \left| X_1^t = x_1^t \right.) = P(X_{t+1} = x_{t+1} \left| X_1 = x_1 \right.) \qquad (5.3)$$

X_i^t ile birbirini izleyen gizli durum dizisi $X_i, X_{i+1}, ..., X_j$ gösterilmektedir. 5.3'te *Markov* zincirinde, $t+1$ zamanındaki durum olasılığı, yalnız zincirin t zamanındaki durumuna bağlı olarak verilmektedir. İkinci varsayım, çıkış bağımsızlığına ilişkin bir varsayımdır:

$$P(Y_t = y_t \left| Y_1^{t-1} = y_1^{t-1}, X_1^{t+1} = x_1^{t+1} \right.) = P(Y_t = y_t \left| X_t = x_t, X_{t+1} = x_{t+1} \right.) \qquad (5.4)$$

Y_i^t ile birbirini izleyen çıkışlar dizisi $Y_i, Y_{i+1}, ..., Y_j$ gösterilmektedir. 5.4'te bir çıkışın t zamanında üretilme olasılığı, salt x_t durumundan x_{t+1} durumuna geçişe bağımlıdır.

Hidden Markov modeli çerçevesinde üç soruna yanıt arayarak bu modelin pratikte kullanılması sağlanır. Bu sorunlar değerlendirme sorunu, kod çözümü sorunu ve öğrenme sorunudur. Değerlendirme sorunu çerçevesinde bir dizi gözlemin verilen bir modelce oluşturulma olasılığı araştırılır. Bu amaçla *forward algorithm* adlı bir algoritma kullanılır (Lee 1993). Kod çözme sorunu eldeki bir dizi gözlemin hangi durum dizisi ile üretildiği sorusuna yanıt arar. Bu sorunun yanıtı *Viterbi* algoritması ile bulunmaya çalışılır. *Viterbi* algoritması sesli ifadelerin kesimlenmesi ve sürekli sesli ifade tanımada uygulama bulmaktadır. Öğrenme aşamasında model parametreleri *forvard-backward* algoritması kullanılarak optimize edilir.

5.2. *Time Warping*

Time warping yöntemi sesli ifade tanımada oldukça sık kullanılan bir diğer yöntemdir. Bu yöntem daha çok diğer yöntemlerle birlikte kullanılan ve daha çok tanıma işlemlerinin verimliliğini artırmak amacıyla kullanılan bir yöntemdir. Bu yöntemde, sesli ifadelerin seslendirme süreleri sıkıştırılarak ya da genişletilerek referanslarla karşılaştırılmaları ilkesi kullanılmaktadır (Parson 1978).

Time warping işleminde sorun A ve B örneklerinin çakıştırılmasıdır. Karşılaştırılacak iki örneğin zaman eksenindeki değerleri,

$$A = a_1, a_2, \cdots, a_i, \cdots, a_m$$
$$B = b_1, b_2, \cdots, b_j, \cdots, b_n \quad (5.5)$$

ise *time wapping* fonksiyonu:

$$C = c(1), c(2), \cdots, c(k), \cdots, c(K) \quad (5.6)$$

olarak yazılabilir. Burada *c,* örneklerin kesişen nokta çiftlerini vermektedir.

$$c(k)=[i(k), j(k)] \quad (5.7)$$

Her *c(k)* için maliyet fonksiyonu:

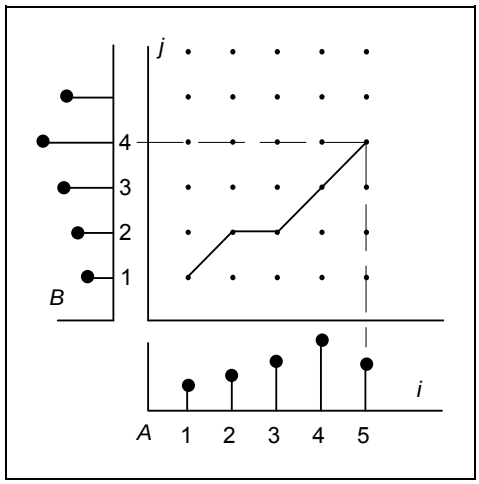
$$d\left[c(k)\right] = \delta(a_{i_{(k)}}, b_{j_{(k)}}) \quad (5.8)$$

denklemi ile bulunur. *d,* burada örnek çiftleri arasındaki zıtlıkdır. Klasik maliyet fonksiyonu ise örnekler arası farkın karesi ile verilmektedir.

$$d\left[c(k)\right] = \delta(a_{i_{(k)}}, b_{j_{(k)}})^2 \quad (5.9)$$

Warping fonksiyonu tüm maliyet fonksiyonlarının toplamı minimize edilerek hesaplanır.

$$D(C) = \sum_{k=1}^{K} d\left[c(k)\right] \quad (5.10)$$



Çizim 5.2. *Dynamic Time Warping* Örneği

Time Warping yönteminin gerektirdiği minimizasyon kimi kısıtlamalarla olanaklıdır. Bu kısıtlamalar şunlardır:

- Maliyet fonksiyonları monotonik özellikte olmalıdır.

$$i(k) \geq i(k-1) \text{ ve } j(k) \geq j(k-1) \qquad (5.11)$$

- Karşılaştırılan *A* ve *B*'nin bitiş noktalarının çakışması gerekmektedir.

$$\begin{aligned} &i(1) = k(1) = 1 \\ &i(K) = M \qquad (5.12) \\ &j(K) = N \end{aligned}$$

- Fonksiyon her noktada tanımlı olmalıdır.

$$\begin{aligned} &i(k) - i(k-1) \leq 1 \\ &ve \qquad (5.13) \\ &j(k) - j(k-1) \leq 1 \end{aligned}$$

- Eğriyi sınırlayacak genel bir üst sınır değeri kullanılmalıdır.

$$|i(k) - j(k)| < Q \qquad (5.14)$$

Q pencere genişliğidir. Karşılaştırmanın sürdürülmesi ya da sonlandırılması için çakışma eğrisinin eğimi kullanılabilir. Diğer taraftan, eğer *M=2N* ya da *N=2M* kuralı sağlanır ya da aşılacak olursa çakışma eğrisi düz bir çizgiye dönüşür ve çakışma gerçekleşmez.

Warping fonksiyonunun hesaplanması, (1,1) noktası ile *(M,N)* noktası arasında en az maliyetli yolu bulmayı gerektirir. Bu bir devingen programlama *(dynamic programming)* problemidir. Genelde bu yöntemde devingen programlama algoritmaları kullanılmaktadır. En düşük maliyetli yol:

$$D(C_k) = d[c(k)] + \underset{\text{legal } c(k-1)}{MIN} [D(C_{k-1})] \qquad (5.15)$$

ile ifade edilir. Geçerli olan *c(k-1)*, tüm *c(k)*lar üzerinde en küçük izin verilebilir yoldur. Eğer *c(k)=(i,j)* ise *(i,j-1),(i-1,j)* ve *(i-1,j-1)* olasılıkları bulunmaktadır. Her nokta için bu üç olasılık geçerlidir.

K adımdan oluşan çakıştırma işleminde w(t) ağırlık fonksiyonu olduğunda toplam maliyet,

$$L = \sum_{k=1}^{K} w(k) \quad (5.16)$$

ile bulunur.

Sesli ifade tanıma kapsamında, sözcüklerin örüntüleri, F_1, F_2 ve F_3 formant sıklıklarından oluşturulmuş matris ile ifade edilebilir. Bu yolla sesli ifade tanımada kullanılacak şablonlar hazırlanmış olur. Bunların herbiri bilinmeyen ya da bulunması istenen sözcüklerle karşılaştırılır.

5.3. Nöron Ağı Yaklaşımlarının Kullanılması

Günümüz sesli ifade tanıma araştırmalarının içinde nöron ağlarının kullanımı oldukça önemli bir yer tutmaktadır. Sesli ifade tanımada nöron ağları, sesli ifade tanıma sistemlerinin çeşitli modüllerinde klasik yaklaşımların yerine kullanılabilmektedir. Bu tez kapsamında Türkçe *phon*'ların nöron ağı yaklaşımı kullanılarak tanınması gerçekleştirilmiştir.

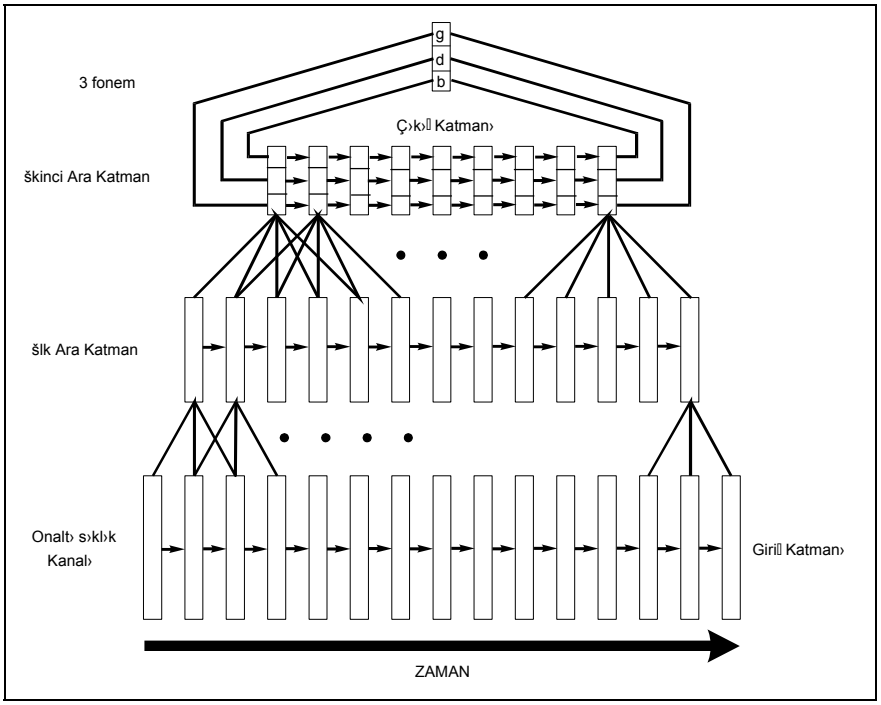
Sesli ifade tanıma sistemlerinde nöron ağı yaklaşımına dayalı çeşitli yöntemler geliştirilmiştir. Bunlar:

- *Time Delay Neural Network*,
- *Self Organizing Feature Map* ve
- *Multi Layer Perceptron*

yöntemleridir. İzleyen kesimde bu yöntemlerin temel ilkeleri verilmiştir.

5.3.1. TDNN (*Time Delay Neural Network*-Zaman Gecikmeli Nöron Ağı)

Sesli ifadeyi oluşturan fonem ve fonem altı ses birimlerinin tanınması amacıyla *Waibel* ve grubunca ortaya atılan *Time Delay* Nöron Ağı yöntemi, nöron ağı kullanan sesli ifade tanıma yaklaşımlarından biridir (*Waibel* 1989). Çizim 5.3'de *TDNN*'in genel yapısı ve buna bağlı çalışma ilkesi verilmiştir. Bu yapıda, sesli ifadenin 10ms'lik kesimlerine ilişkin özellik vektörleri kullanılmaktadır. Sistemin tanıma sonucunu üretebilmesi tüm giriş düğümlerinin dolmasını gerektirir. 15 giriş düğümü, 10 ms'de bir gelen özellik vektörleri ile çıkış en erken 150 ms sonra üretilir.



Çizim 5.3. TDNN'ün *(Time Delay Neural Network)* çalışma ilkesi

İnsan konuşmasındaki değişimleri yakalayabilmek için 10 ms'de bir örneklem alma (özellik vektörü hesaplama) yeterli olmaktadır. 10ms'lik örneklemlerden en az 3 tanesini içeren 30 ms'lik bir pencere içinde, tanımaya taban oluşturacak yeterli sayıda sesli ifade özelliği bulunur. Ses örneklemlerinden 15 tanesi *TD* nöron ağının giriş verisini oluşturur (16x3 düğüm). Örneklemlerin herbiri 16 farklı frekans bölgesi için genlik değerlerine dönüştürülür. Sistemde 30 ms'lik bir percere ile bölgesel *acoustic-phonetic* özellikler fonemlerin tanınmasında kullanılır. *TDNN*'de üç zaman kesitini kullanan giriş düğümlerinin çıkışları, 8'li ilk ara katman düğümlerine girişleri oluşturur. Bu işlem 13 kez pencere kaydırılarak yinelenir. Daha sonra bu düğüm çıkışları, giriş katmandakilere benzer biçimde ikinci ara katmana aktarılır. İkinci ara katman da 5 zaman kesitini pencere olarak kullanmaktadır. Bu defa 9 kez ikinci pencere kaydırılır ve oluşturulan sonuçlar herbir üç düğüme aktarılır. Bu son çıkış düğümlerinin yatay olarak 150 ms'lik ses örneğinin değerlendirilmesi ile en baskın olan *fonem,* çıkış olarak seçilir. Bu yöntemle bir konuşmacı üzerinde yapılan uygulamada */b/, /d/, /g/* ünsüzleri başarıyla tanınabilmiştir. Gerçek sorun pratik uygulamada, boyuta ilişkin olmaktadır. Boyut ya da tanınacak fonem sayısı artığında bunu gerçekleştirecek bilgisayarın ve algoritmaların karmaşıklığı da artmaktadır. Sistemi hızlandırma yöntemi olarak süper ağ yapıları denenmiştir. Bu yapıda paralel çalışan birden fazla ilk ve ikinci ara katman öngörülmektedir. Bu yapı, çalışma grubunca modüler nöron ağı olarak adlandırmaktadır.

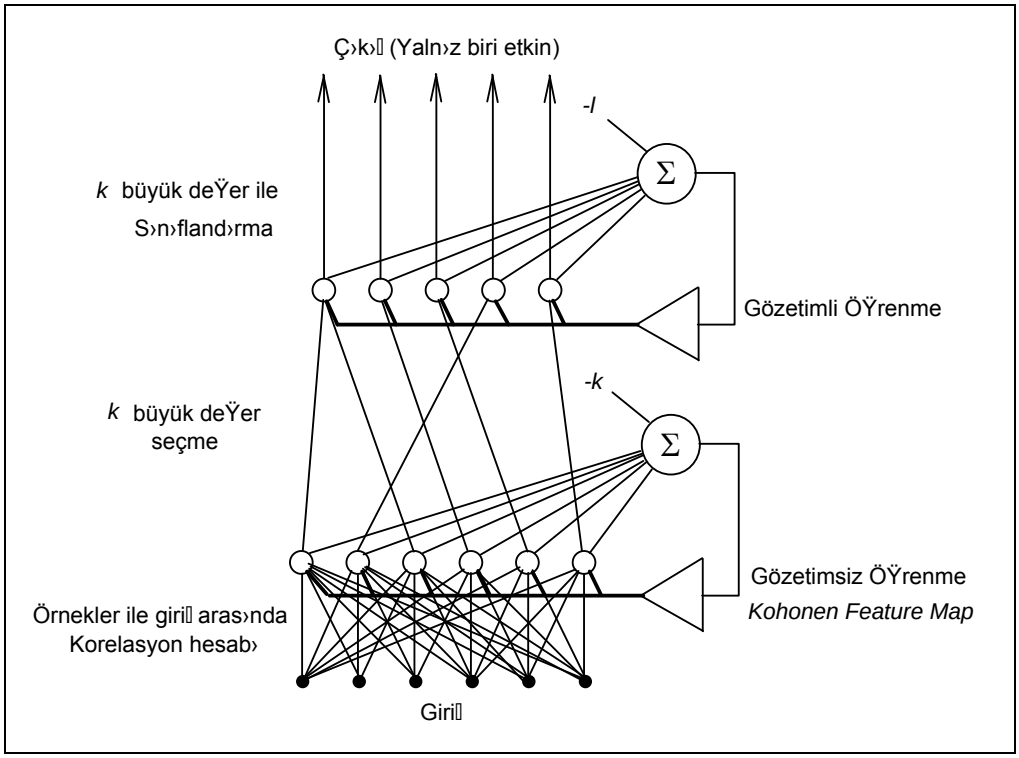
5.3.2. Kohonen *Self Organizing Feature Map*

Fince ve Japonca sesli ifade tanıma için nöron ağı yaklaşımını kullanan uygulamalar *Kohonen* ve grubu tarafından başlatılmıştır. Amaç sesle çalışan bir daktilodur. Kullandıkları yönteme *fonetik* harita ya da *phonotopic map* adı verilmiştir. Kullanılan yöntemde birim olarak fonemler kullanılmaktadır. *Kohonen* ve grubunun kullandığı yaklaşım iki kesime ayrılır. İlk kesim *(acustic preprocessor)* gelen ses sinyallerinin sayısallaştırılması ve özellik vektörü hesaplamaya ilişkindir. Aslında bu kesim her sesli ifade tanıma dizgesinde bulunmaktadır. İkinci kesimde ise iki boyutlu, tek katmanlı nöron ağ yaklaşımı kullanılmaktadır. Yapılan prototip çalışmada, her iki kesim de bir *IBM XT/AT* üzerine takılı iki kart ile gerçekleştirilmiştir.

Kohonen'in *Self Organizing Feature Map* algoritması bu tez kapsamında kullanıldığından 7. kesimde ayrıntılı olarak açıklanacaktır.

5.3.3. Viterbi Çözümleyicisi

Diğer bir yaklaşım ise *Lippmann* ve grubunun yaptığı çalışmadır (Huang 1988). Zaman kavramının kullanıldığı *Viterbi* çözümleyicisinin *HMM* üzerine uygulanması durumunda çok etkili olduğu görülmüştür. Günümüz sesli ifade tanıyıcıların, konuşmacıdan bağımsız ve sürekli konuşma tanımada gösterdikleri başarım düşük olmaktadır. Nöron ağlarının örneksel donanım bileşenlerine dayalı olarak kurulması yüksek işlem gücü yaratabilmektedir. Nöron ağ sistemlerindeki *self organization,* öğrenme, uyarlamanın gerektirdiği paralel algoritmalara dönük geliştirilme çabaları yaygınlaşmıştır.

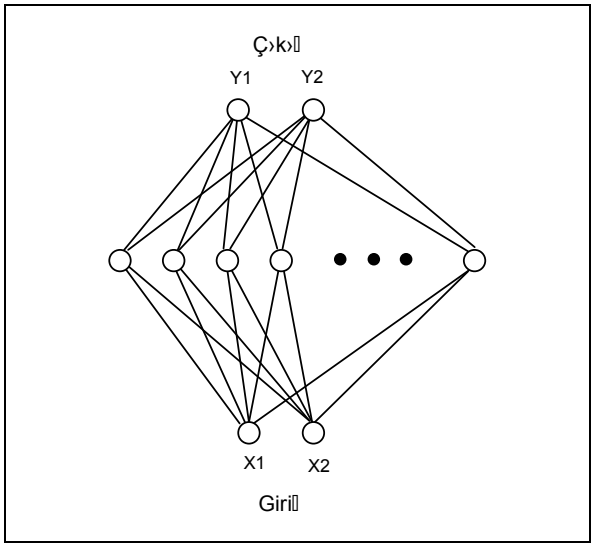


Çizim 5.4. Viterbi Çözümleyicisinin Genel Görünümü

Viterbi topolojisi iki katmandan oluşmaktadır. Birinci katman iki boyutlu tek katmandan oluşan *Self Organizing Feature Map* sınıflandırıcıdır. Bu katman *k-means* kümeleme algoritmasının değiştirilmiş bir biçimini kullanmaktadır. Gözetimsiz öğrenme aşamasında *feature map* katmanında örnekler sınıflandırmaya tabi tutulur. Çıkış katmanı, *feature map* düğümleri arasından en büyük çıkışa sahip olan düğümü göstermektedir.

5.3.4. *Multi Layer Perceptron*

Durgun örüntü sınıflandırıcılarda nöron ağı, çok katmanlı algılayıcıyı kullanmaktadır *(multi layer perceptron)*. Bu tür sınıflandırıcıların ara ve çıkış katmanlarındaki düğümler *sigmoid non-linearities* ögelerini kullanmaktadır. Ağırlıklar ve eşik (*offset*) değerleri etiketlenmiş ya da geri bildirimli algoritma kullanılarak günlenmektedir.



Çizim 5.5. İki katmanlı *perceptron* sınıflandırıcı

Multi layer perceptron yaklaşımına dayalı olarak *Peterson* ve *Barney* sesli *formant* verileri için iki katmanlı sınıflandırıcı ve geribildirim algoritması kullanmışlardır(Lippmann 1986). Bu sınıflandırıcının iki girişi (F_1 ve F_2), 100 ara ve 10 çıkış düğümü bulunmaktadır. Her bir ses birimine bir çıkış düşmektedir. İki düzeyli bir *perceptron, Gaussian* sınıflandırıcıda olduğu gibi geri bildirim özelliğini kullanarak oldukça hızlı ve basit bir biçimde karar bölgelerini belirleyebilmektedir.

6. TÜRKİYE TÜRKÇESİNİN FONETİK ÖZELLİKLERİ

Sesli ifadelerde birim öge, *ses (phon)* dir. Sesli ifadeden yazıya geçiş, her sese bir alfabetik simge verilerek gerçekleşebilir. Ancak ses sayısının çok olması, sesli ifadeden yazıya geçişi karmaşıklaştırır. Bu nedenle, kimi sesleri kümeleyerek her kümeye tek bir simge verme yolu da kullanılabilir. Bu bağlamda seslerin kümelenmesi *fonem* tabanında gerçekleşir. Fonemler, anlam ayırıcı özelliği bulunan ses kümeleridir. Başka bir anlatımla anlam ayırıcı özelliği bulunmayan sesler tek bir fonem kümesi altında toplanır. Ses, diller üstü bir birimdir. Fonem ise anlam ayırıcı özellik içermesi nedeniyle belli bir dile özgü birimdir. Sesler tüm dillere ortak ögeler olarak düşünüldüklerinden tüm diller için geçerli olan bir fonetik alfabe *IPA (International Phonetic Alphabet)* tanımlanmaya çalışılmıştır. Bu tür alfabelerin oluşturulmasındaki amaç, her sesin bir karakter ile temsil edilmesidir. Bu biçimde, sesli ifadelerin, ses tabanında, dillerden bağımsız yazıya dönüştürülmesi amaçlanmaktadır. Yukarıda da belirtildiği gibi bu tür alfabeler çok sayıda simge içermeleri nedeniyle yaygın olarak kullanılamamaktadır. Bu durumda her dilin kendine özgü alfabeleri ve sesli ifadeden yazıya geçiş kuralları bulunmaktadır. Bu bağlamda Türk Alfabesi, *fonemik* bir alfabe olarak benimsenebilir. Türkçenin her fonemine tek bir alfabetik simge (harf) atandığı söylenebilir. Bu sayede sesli ifadeden yazıya, yazıdan sesli ifadeye geçişin kuralları oldukça yalındır.

Seslerin fonemler olarak kümelenmesinde izlenen yol, yukarıda da belirtildiği gibi anlam ayrımına dayanmaktadır. Bir sözcük içerisinde yer alan bir ses, *bAl* ile *bEl* örneğinde olduğu gibi, diğer bir ses ile değiştirildiğinde sözcüğün anlamı değişiyorsa sözkonusu iki sesin ayrı kümelerde yer aldığı, dolayısıyla ayrı fonemlere ilişkin olduğu söylenir. Bunun tersine ses değiş tokuşu anlam ayrımına yol açmıyorsa sözkonusu sesler aynı foneme ilikin sesler olarak düşünülür. Örneğin: Türkçe'de /e/ ile simgelenen üç değişik ses vardır. Bu bağlamda, Türkçe sesli ifade tanıma sistemleri, birim olarak ses tabanında çalışmak ancak yazıya geçişte fonemleri taban almak durumunda olmalıdır. Sesleri fonemler olarak kümelemek için *yalın çift (minimal pair)* olarak anılan sözcük çiftleri kullanılmaktadır.

Türkçe'de diğer dillerde olduğu gibi, fonemleri parçalı *(segmental)* ve parçalarüstü *(suprasegmental, prosodics)* olarak iki grupta incelemek mümkündür. Parçalı fonemler harflerle simgelenebilen fonemlere verilen addır. Parçalar üstü özellik ise,

vurgu, uzatma, uyum, ezgi gibi simgelerle ifade edilemeyen sessel özelliklere verilen addır.

6.1. Türkçe Parçalı Sesbirimler/ Fonemler

Türkçede parçalı sesler, genel sınıflandırmaya paralel olarak:

- ünlü,
- ünsüz
- kayan ünlü

seslerden oluşur.

6.1.1.Ünlüler

Fonetik açıdan incelendiğinde Türkçede bulunan ünlü seslerin sayısı 16'dır(/i/, /I/, /e/, /ɛ/, /æ/, /a/, /y/, /Y/, /ø/, /oe/, /ï/, /u/, /U/, /O/, /ɔ/, /ɑ/). Her ünlünün bir *açık* bir de *kapalı* türü vardır. Ünlülerin açık ya da kapalı biçimde olmaları içinde bulundukları sözcüklerin anlamlarını değiştirmediğinden tüm ünlü sesler 8 ünlü fonem ile temsil edilir. Bilindiği gibi bunlar:

/a/, /e/, /o/, /ö/, /u/, /ü/, /ı/, /i/

dir. Bilindiği gibi ünlüler ses telleriyle oluşturulan titreşimlerin ses yolunda rezonansa sokulması yoluyla elde edilen seslerdir. Ünlüler, *çene açıklığına*, *dilin ağız içindeki konumuna*, *dudakların biçimine* ve *genzin rezonans kutusuna katılıp katılmamasına* göre sınıflandırılmaktadır. Bu bağlamda ünlüler 4 değişik parametre ile tanımlanmaktadır. Bunlar:

1. Çene açıklığı *dar (tongue high), geniş (tongue low)* (kimi durumlarda *orta* açıklıktan da söz edilebilmektedir.)
2. Dil *önde (tongue front), arkada (tongue back)*
3. Dudaklar *yuvarlak (lips rounded), düz* (yuvarlak değil) *(lips unrounded)*
4. Geniz *açık (nasalized), kapalı (unnasalized)*

parametreleridir. Türkçede bu parametrelerden ilk üçü önem taşımaktadır. Türkçe ünlü fonemlerin bu kıstaslara göre sınıflandırılmaları Çizelge 6.1'de verilmiştir.

Çizelge 6.1. Türkiye Türkçesindeki ünlülerin çıkış yeri ve biçimi.

			a	e	o	ö	ı	i	u	ü
Çene Açıklığı	F2 artar	Geniş Ünlüler	✓	✓	✓	✓				
	F1 azalır	Dar Ünlüler					✓	✓	✓	✓
Dudak Biçimi		Düz Ünlüler	✓	✓			✓	✓		
		Yuvarlak Ünlüler			✓	✓			✓	✓
Dilin Konumu	F2 azalır	Arka	✓		✓				✓	
		Orta					✓			
		Yuvarlak Ön				✓				✓
	F2 artıyor	Düz Ön		✓				✓		

Bilindiği gibi, sesler fonemler olarak kümelenirken, bir sesin diğer bir sesle yer değiştirmesi sonucu anlam değişikliği olup olmadığını sınamak üzere yalın sözcük çiftleri kullanılmaktadır. Türkçe ünlü fonemler, yukarıda açıklanan kıstaslar çerçevesinde, bu fonemleri belirlemede yararlanılan yalın sözcük çiftleri ile Çizelge 6.2, Çizelge 6.3 ve Çizelge 6.4'de verilmiştir.

Çizelge 6.2. Çene açıklığına göre yalın çift örnekleri.

Dar\Geniş	a	e	o	ö
ı	kar / kır	kes / kıs	koş / kış	söz / sız
i	kar / kir	tez / tiz	şoş / şiş	söz / siz
u	kar / kur	bez / buz	koş / kuş	son / sun
ü	sar / sür	ses / süs	son / sün	söz / süz

Çizelge 6.3. Dudakların biçimlerine göre yalın çift örnekleri.

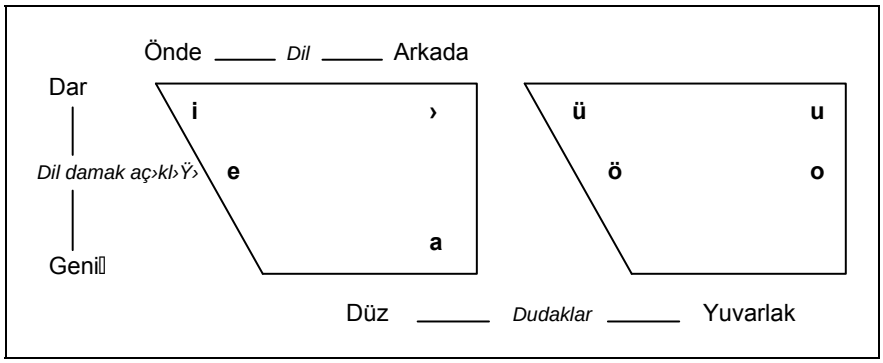
Yuvarlak\Düz	a	e	ı	i
o	kar / kor	kel / kol	kış / koş	kir / kor
ö	kar / kör	gel / gör	sır / sör	kir / kör
u	kar / kur	ser / sur	tır / tur	kir / kur
ü	sar / sür	kel / kül	tır / tür	kir / kür

Çizelge 6.4. Dilin konumuna göre yalın çift örnekleri.

Arka\Ön	a	o	u
e	kas/kes	kol /kel	sur/ser

ö	kar/kör	kor/kör	sun/sön
i	kar/kir	kor/kir	kur/kir
ü	sar/sür	kor/kür	kul/kül

Ünlü fonemleri sınıflandırarak göstermede kullanılan bir yöntem de *ünlü dörtgeni (vowel diagrams)* yöntemidir. Yukarıda verilen üç kıstastan dilin konumu ve çene açıklığı göz önüne alınarak ünlü dörtgenleri oluşturulmaktadır. Bu dörtgenler iki boyutlu olup yatay eksen dilin arkada ve öndeki konumunu, dikey eksen ise, çene açıklığının dar ve geniş olmasını göstermektedir. Bu dörtgenlere, dudakların biçimine ilişkin boyut, düz ve yuvarlak biçimlerinin herbiri için, iki ayrı ünlü dörtgeni oluşturularak katılmaktadır. Örnek bir Türkiye Türkçesi Ünlü Dörtgeni Çizim 6.1'de verilmiştir. Ünlü dörtgenleri ünlü fonemlerin ancak kaba bir sınıflandırmasına olanak verebilmektedir.



Çizim 6.1. Örnek Türkiye Türkçesi Ünlü Dörtgeni (Demircan, 1979)

Türkçe ünlü fonemlerin sözcükler içinde yer aldıkları konumlar incelendiğinde aşağıdaki sonuçlar elde edilmiştir (Demircan 1979). Fonemin sözcük içindeki konumu ön, iç ve son hecede olarak belirlenir. Bu bağlamda:

/a/ sözcüklerin ön, iç ve son hecelerinde bulunur.

/e/ sözcüklerin ön, iç ve son hecelerinde bulunur.

/o/ sözcüklerin ön ve iç hecelerinde bulunur. Bu fonemi son hecesinde bulunduran sözcükler yabancı kökenlidir.

/ö/ sözcüklerin ön ve orta hecelerinde bulunur. Son hecede bulunamazlar.

/u/ sözcüklerin ön ve iç hecelerinde bulunur. Son hecede bulunamazlar.

/ü/ sözcüklerin ön, iç ve son hecelerinde bulunur.

/ı/ sözcüklerin ön, iç ve son hecelerinde bulunur.

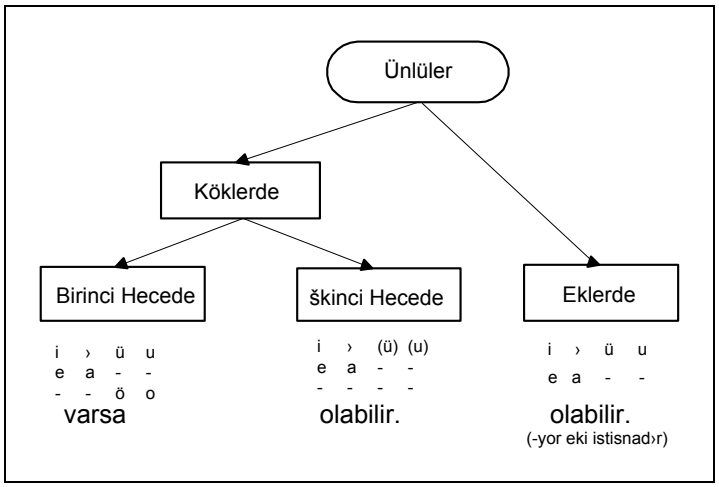
/i/ sözcüklerin ön, iç ve son hecelerinde bulunur.

Türkçe'nin en önemli özelliklerinden biri de ünlü uyumudur. Ünlü uyumu sözcüklerin ilk hecesindeki ünlünün izleyen hecelerdeki ünlüleri etkilemesidir. Sözcüğün ilk hecesindeki ünlü, dilin ağız içinde önde konumunda çıkarılan bir ünlü ise, başka bir deyişle öndil ünlüsü (/ö/,/ü/,/e/,/i/) ise izleyen hecelerdeki ünlüler de öndil ünlüsü olmak zorundadır. Sözcüğün ilk hecesindeki ünlü, dilin ağız içinde arka konumunda çıkarılan bir ünlü ise, başka bir deyişle arkadil ünlüsü ise izleyen hecelerdeki ünlüler, ortadil (/ı/) ya da arkadil ünlüsü (/a/,/o/,/u/) olur. Bu biçimde sözcüğün ilk hecesinden sonraki hecelerde gelebilecek ünlülerin sayıları kısıtlanmış olur.

Ünlü uyumu, ilk hecedeki ünlünün düz ünlü (dudaklar düz iken çıkarılan ünlü) olması durumunda (/ı/,/i/,/a/,/e/) diğer hecelerdeki ünlülerin de düz ünlü olmasını zorunlu kılar. İlk hecede yer alan ünlü yuvarlak (/o/,/ö/,/ü/,/u/) ünlü ise, diğer hecelerdeki ünlüler düz ya da yuvarlak ünlü olabilmektedir.

Bu kuralların yanı sıra iki yuvarlak ünlü arasına düz ünlünün girmesi durumunda düz ünlünün dar olan ile değişmesine neden olur. Bunun gibi geniş ünlü ile biten bir köke ünlü ile başlayan bir ek geldiğinde geçiş sesi olan /y/ eklenerek bu ünlü dar ünlüye dönüştürülür.

Ünlülere, içinde bulundukları konumlara göre getirilen varolma kısıtlamaları Çizim 6.2'de özetlenmiştir.



Çizim 6.2. Ünlülerin sözcük içinde yer alma kuralları (Demircan, 1979)

6.1.2. Ünsüzler

Türkçede ünsüzler 20 tanedir:

/b/, /c/, /ç/, /d/, /f/, /g/, /h/, /j/, /k/, /l/, /m/, /n/, /p/, /r/, /s/, /ş/, /t/, /v/, /y/, /z/

Ünsüzler çıkış noktası, çıkış biçimi ve ses teli titreşimlerinin varlık ve yokluğuna göre sınıflandırılırlar. Çizelge 6.5'de Türkçenin ünsüzlerinin sözü edilen bu özelliklere göre sınıflandırılmaları verilmiştir.

Çizelge 6.5. Ünsüzlerin çıkış noktası, biçimi ve titreşime göre sınıflandırılması

Özellikler	**b**	**c**	**ç**	**d**	**f**	**g**	**h**	**j**	**k**	**l**	**m**	**n**	**p**	**r**	**s**	**ş**	**t**	**v**	**y**	**z**
Patlamalı	✓					✓			✓				✓				✓			
Genizden											✓	✓								
Çarpmalı														✓						
Yarı Ünlü																			✓	
Yan Ünsüz										✓										
Sızmalı		✓	✓		✓		✓	✓							✓	✓		✓		✓
Çift Dudak	✓										✓		✓							
Alt Dudak-Üst Diş					✓													✓		
Dilucu-Dişardı				✓													✓			
Dilucu-Dişeti												✓		✓	✓					✓
Dilucu-Sert damak										✓										
Dil-Sert damak		✓	✓					✓								✓			✓	
Dil-Damak sonu						✓			✓											
Ses telleri arası							✓													
Ötümlü	✓	✓		✓		✓		✓		✓	✓	✓		✓				✓	✓	✓
Ötümsüz			✓		✓		✓		✓				✓		✓	✓	✓			

Türkçedeki ünsüzlerin, sözcükler içinde bulundukları konumlara göre özellikleri aşağıda verilmiştir:

/b/ Sözcüklerin ön ve iç sesleri olabilir. Son seste ötümsüzleşerek /p/ sesine dönüşür. Ünlü ile başlayan ek aldığında ötümsüzleşir.

/c/ İki sesten oluşan bileşik bir sestir. İç seste birisi düşer. Son seste bir iki sözcük dışında bulunmaz.

/ç/ /t/ ve /ş/ seslerinden oluşan bileşik sestir. İç ses olması durumunda bir ünsüz ile yanyana geldiğinde /t/ sesi kaybolur.

/d/ Sözcüklerin ön ve iç sesleri olabilir. Son seste ötümsüzleşerek /t/ sesine dönüşür.

/g/ Sözcüklerin ön ve iç sesleri olabilir. Son seste ötümsüzleşerek /k/ sesine dönüşür. Kimi zaman ön ve iç seslerde /ğ/'e dönüşür.

/ğ/ Ses niteliği taşıyıp taşımadığı tartışma konusudur. Ön ses olamaz. Kayan ünlü oluşumunu sağlar.

/h/ Sözcüklerin ön, iç ve son sesleri olabilir.

/j/ Sözcüklerin ön, iç ve son sesleri olabilir. Türkçede bu sesin bulunduğu tüm sözcükler alıntı olduğundan Türkçe fonem alfabesinde bulunup bulunmadığı tartışılmaktadır.

/k/ Sözcüklerin ön, iç ve son sesleri olabilir.

/l/ Sözcüklerin ön, iç ve son sesleri olabilir. Dilin önde ve arkada tutularak çıkarılan iki türü vardır.

/m/ Sözcüklerin ön, iç ve son sesleri olabilir. Ön ses olarak bu fonemi içeren sözcükler Türkçe değildir.

/n/ Sözcüklerin ön, iç ve son sesleri olabilir.

/p/ Sözcüklerin ön, iç ve son sesleri olabilir.

/r/ Ön seste birden çok çarpmalı kullanılır. /r/esim, /r/af. İç seste tek çarpmalıdır. ı/r/mak, so/r/u. Son seste sızmalıdır. pına/r/, bi/r/ gibi.

/s/ Sözcüklerin ön, iç ve son sesleri olabilir.

/ş/ Sözcüklerin ön, iç ve son sesleri olabilir.

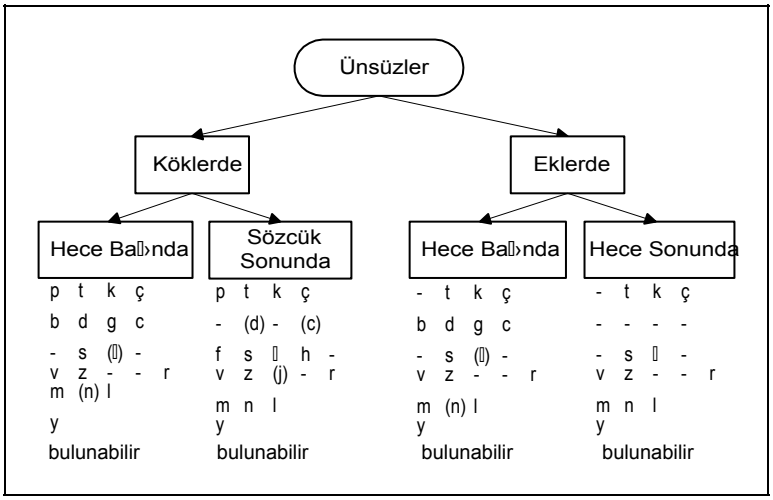
/t/ Sözcüklerin ön, iç ve son sesleri olabilir.

/v/ Sözcüklerin ön, iç ve son sesleri olabilir. Bazı durumlarda ünlü kaymasına neden olur.

/y/ Sözcüklerin ön, iç ve son sesleri olabilir. Yarı ünlüdür. /i/ fonemine yakın yerden çıktığından ünlü kayması olur. Son ses olması durumunda ötümsüzdür.

/z/ Sözcüklerin ön, iç ve son sesleri olabilir. Son ses olması durumunda ötümsüzleşir. ka/z/.

Türkçe ünsüzlerin sözcükler içindeki konumlarına göre varlık özellikleri Çizim 6.3'de verilmiştir.



Çizim 6.3. Ünsüzlerin sözcük içindeki varlık özellikleri (Demircan, 1979)

6.1.3. Kayan ünlüler

Aynı hece içinde yanyana bulunan ve tek sesmiş gibi çıkarılan ya da başlangıçtaki sesin değişmesi ünlü kaymasıdır. Türkçede özellikle konuşma dilinde kayan ünlülerin bulunduğu kanıtlanmıştır. Türkçenin hece yapısı, iki ünlünün yan yana gelmesini kabul etmemektedir. Ancak yabancı dillerden alıntı sözcüklerde /y/ ve /v/ ünsüzlerinin araya girdiği ve yazıya da geçtiği gözlenmektedir (fiat- fiyat örneğinde olduğu gibi). Ayrıca ses niteliği taşıyıp taşımadığı tartışma konusu olan /ğ/ ünlü kaymasına neden olmaktadır. Özellikle kendinden önce gelen ünlünün uzunluğunu değiştirme gibi bir etkisi bulunmaktadır. Ünlü kaymasına neden olan diğer sesler /y/ ve /v/ dir(Ergenç 1989).

6. 2. Türkçe'de Parçalar Üstü Sesbirimler (Bürün)

Fonemlerin ya da parça sesbirimlerin en önemli özellikleri anlam ayırıcı özelliklerinin bulunmasıdır. Anlam ayırıcılık, parça sesbirimleri örten *vurgu (stress), uzatma ya da süre (length), uyum (harmony), perde değişimi (pitch variation) ve ezgi (intonation)* ile de sağlanabilmektedir. Bunlar parçalar üstü sesbirimler ya da *bürünbirimler* olarak tanımlanmaktadır. Türkçede genelde bunların anlam ayırt edici özellikleri bulunmadığı kabul edilmektedir. Ancak Japonca ve Çinçe gibi yapısı vurguya dayanan dillerde bu kavramlar tanıma açısından oldukça önemli özelliklerdir. Bürünbirimler sesli ifade tanıma kapsamında, sesli ifade örüntüsünün yakalanmasını etkilemekten çok anlam çıkarma aşamasında önem taşımaktadırlar.

6.2.1. Süre

Sesin çıkarılışındaki süre, onların uzun ya da kısa söylenişleri ile ilgilidir. Her dilde kendine özgü biçimdedir. Bir dildeki uzun süreli fonem bir diğer dilde kısa süreli kabul edilebilmektedir. Türkçede sürenin anlam ayırd edici özelliği bulunmamaktadır. Ancak Türkçeye yabancı dillerden geçen sözcüklerde uzatma anlam ayırd edici özelliğe sahip bir bürünbirim olarak ortaya çıkmaktadır. Bununla birlikte, iç ve son seste kullanılan /ğ/ ve /h/ seslerinin yutulması durumunda uzayan ünlülerden söz edilebilmektedir (dün-düğün). Türkçe'de ünsüzler için süre, seslenme ve buyurma biçimlerinde ortaya çıkmaktadır (sakın! gibi).

6.2.2. Perde Değişimi

Dilin bürünsel özelliğinden sayılan perde değişimi heceyi ilgilendirir. Hecenin tiz ya da bas söylenmesi olarak tanımlanır. Çince, Nijerya dilleri gibi kimi dillerde sözcük anlamlarını ayırmada kullanılan perde değişimi, Türkçede ezgi birimine bağlıdır ve genellikle tek sözcüklük bildirilerde anlam ayırıcı özelliği bulunmaktadır. Örneğin aferin sözcüğü ile ya beğeni belirtme ya da yaptığını beğendin mi türünden serzeniş dile getirme perde değişimi ile sağlanır.

6.2.3. Vurgu

Bir sözcükte bulunan bir hecenin diğer hecelere göre daha baskılı biçimde çıkarılması vurgu olarak bilinir. Bir de tümce içinde sözcük vurgusu vardır. Türkçede vurgunun yeri genellikle son hecedir. Ancak Türkçenin bağlantılı,

sözcükleri ek alan bir dil olması nedeniyle vurgunun yerinin değişken olduğunu düşünmek daha doğrudur. Vurgu Türkçede anlam ayırıcı özellik içerir.

6.2.4. Ezgi

Sesli ifadedeki hece, durak ve vurguya bağlı olarak ortaya çıkan perde değişimlerine ezgi denilmektedir. Tümüyle konuşmacının, konuşma biçimine bağlı olduğundan kesin kurallar tanımlanması zordur. Bu nedenle çok az incelenmiştir.

6.2.5. Kavşak ve Durak

Ünsüz ile bitmiş kök, ünlü ile başlayan bir ek aldığında hece düzenin değişmesidir (*bi-lim* / *bi-li-min* de olduğu gibi). Aynı zamanda Türkçenin ünlüyle başlayan sözcüklerinde de kavşak vardır. Bu tür sözcüklerde ünlü sesin önüne, gırtlak çarpması olarak adlandırılan ve gırtlaktan gelen ötümsüz, patlamalı bir sesin eklendiği varsayılır. Bu nedenle, örneğin, *ulak* sözcüğü *ul-ak* değil *u-lak* biçiminde söylenir. Türkçede varlığı sözcük başında olan bu ses, Arapça kökenli sözcüklerde iç ve son seste de bulunmaktadır.

Bir tümce içinde anlam birlikleri verilen kısa duraklar ile ayrılır. Durağın Türkçede anlam ayırıcı özelliği büyüktür (*Kara, deniz, hava yolları* ile *Karadeniz havayolları* arasındaki farkta olduğu gibi).

6.3. Türkçe'de Hece Türleri

Sesli ifade üretme kaburgalar arası kaslarca biçimlendirilen göğüs atışlarıdır. Her göğüs atışıyla birlikte ses telleri de titreşmeye başlar. Bu biçimde ünlü sesin çıkarılması gerçekleşir. Her göğüs atışı periodunda çıkarılan sesler hece olarak anılır. Her atış sürecinde belirli basınçta soluk ses yolundan dışarı çıkar. Bu basınçlı havaya ses tellerinin titreşimleri eklenmesiyle ünlü sesler çıkarılır. Kasların gevşemesiyle birlikte geçen havanın azalması ya da durdurulması sırasında ses tellerinin titreşmesi ya da durması ünsüz seslerin çıkarılmalarını sağlar. Ünsüzlerin dağılımına göre hece türleri belirlenmektedir(Ünlü:Ü -ünsüZ:Z)

Çizelge 6.6. Ünsüz dağılımlarına göre hece yapıları

	Hece	**Örnekler**
engelsiz	Ü	O
gevşeme engelli	ZÜ	de ,ne,şu
duruş engelli	ÜZ	et, is, in

gevşeme ve duruş engelli	ZÜZ	sis, düş, gel

Genelde hece başında ve sonundaki bu ünsüz sesler anlaşmazlıkları oluşmaktadır. Özellikle hece başı ve sonunda birden fazla ünsüz kullanan dillerde bu olay yaygındır. Bu nedenden ötürü İngilizce konuşmada kavşak gibi bir özelliğin dile katılması gerekmiştir. *I scream* ile *ice cream* ya da *night rate* ile *nitrate* arasında seslendirme açısından bir fark olmamasına karşın yazılış ve anlam farkı bulunmaktadır. Türkçede kavşak özelliği bulunmamaktadır. Ancak duraklar ve kullanıldıkları yerler anlam farkını oluşturmada kullanılmaktadır(Demircan 1979).

Çizelge 6.7. Türkçe Hece Türleri

Hece Türü	**Örnek**
Ü	O
ZÜ	bu, su
ÜZ	al, at, an
ÜZZ	üst, alt, ilk
ZÜZ	gir, bul, yen
ZÜZZ	Türk, çark, sırt

Çizelge 6.7'de Türkçede kullanılan hece türleri verilmiştir. Türkçede en fazla kullanılanlar ZÜ ve ZÜZ türü hecelerdir. Türkçede hece türlerinin kullanım sıklığı Çizelge 6.8'de verilmiştir.

Çizelge 6.8. Türkçe hece yapısı ve sözcük içindeki kullanım sıklıkları.

	Ü	**ÜZ**	**ÜZZ**	**Z Ü**	**Z ÜZ**	**Z ÜZZ**
Sözcük içinde hece yapısının geçiş sıklığı(Demircan 1979)	başta %4,9 ortada %1,4	%4,2 %0,6	%0,4 %18,3	%52,9 %38,9	%37,1 %28,3	%0,5 %12,5
Tek Heceli / Sözcük Başında	✓	✓	✓	✓	✓	✓
Ortada	Alıntı	✓	Alıntı	✓	✓	✓
Sonda	✓	✓	Yok	Yok	✓	✓
Olasılık Sayısı	✓	160	3200	160	3200	64000

Türkçede bu çizelge 6.7'de verilen hece türleri tek başlarına ya da sözcük başında kullanılırlar. Sözcük ortasında ÜZ, ZÜ, ZÜZ, ZÜZZ, sözcük sonunda ise Ü, ZÜ,

ZÜZ, ZÜZZ türü heceler kullanılmaktadır. Ayrıca yabancı dillerden geçen sözcükler ile birlikte türlerde farklılık göstermektedir. Bunlara örnek olarak,

ZÜZZZ tekst

ZZÜ spiker

ZZÜZ tren, stok

ZZÜZZ tröst, flört

ZZZÜZ stres

verilebilir.

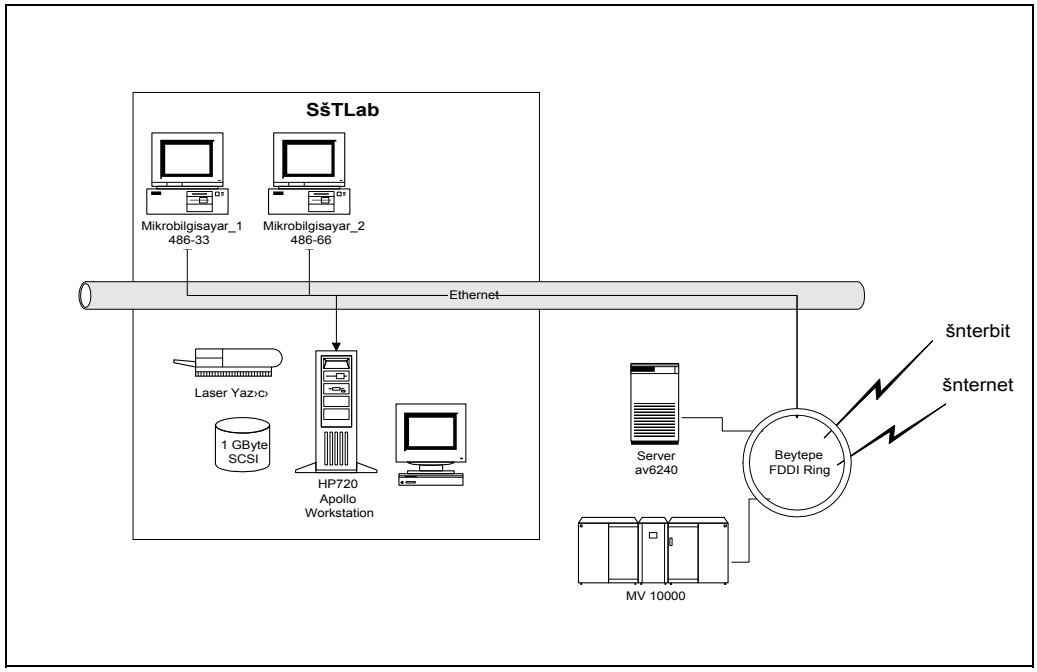
7. UYGULAMA ORTAMI ve TÜRKÇE SESLİ İFADE TANIMA

Sesli ifade tanıma süreci iki kesimde ele alınmaktadır. İlk kesim, sesli ifade sinyallerinin sayısallaştırılması, işlenmesi ve bunun sonunda tanınmaya hazır, temsil niteliği yüksek özellik vektörleri olarak anılan biçime dönüştürülmesidir. Daha sonraki kesim ise üretilmiş olan bu sesli ifade özellik vektörlerinin kullanıldığı tanıma kesimidir.

Bu çalışmada, sesli ifade tanıma süreci, sesli ifade sinyallerinin yazılı metine dönüştürülmesine kadar uzanan süreç olarak ele alınmaktadır. Sözcükler, tanınan her foneme karşı gelen harflerin yanyana dizilmesiyle bütünlük kazanmaktadır. Bu çalışmada elde edilen sözcüklerin yazım ve anlamsal doğrulukları sınanmamaktadır. Bunun, sözdizimsel ve biçimsel *(syntactic and morphological)* çözümleme çalışmalarının yer alalacağı bir sonraki aşamada ele alınması doğru olacaktır. Türkçe için fonem tanıma aşaması sonrası doğrudan sözcükleri oluşturma ve bunların geçerliliğini bir sözlükten yararlanarak sınama, Türkçe sözcük sayısının bir üst sınırı olmaması nedeniyle mümkün değildir. Bu gerekçeye dayalı olarak Türkçe'de, tanıma aşamasında önce hecelerin elde edilmesi ve sınanmalarının hece düzeyinde ele alınmasının daha elverişli olacağı görülmektedir. Türkçe'de, hecelerin yapılarının yalın ve sayılarının kısıtlı olması bu yaklaşımı anlamlı kılmaktadır. Ancak Türkçe'ye yabancı dillerden giren sözcüklerin bu yaklaşımın geçerliliğini ne kadar bozduğunun incelenmesi gereklidir.

Bu tez çalışması TÜBİTAK desteği ile Hacettepe Üniversitesi Bilgisayar Bilimleri Mühendisliği Bölümünde kurulan Sesli İfade Tanıma Labaratuvarında (SİTLab) yürütülmüştür. Sesli İfade Tanıma Labaratuvarı sesli ifade tanınmaya ilişkin birikim oluşturma ve bu birikimin Türkiye Türkçesi üzerine uygulanması amacıyla kurulmuştur. Bu birikim yukarıda sözü edilen iki eksende oluşturulmaya çalışılmıştır. İlk eksende, Türkçe'ye en uygun sesli ifade özellik vektörlerinin belirlenmesi ve elde edilmesi, ikinci eksende ise, çeşitli tanıma tekniklerinin araştırılması ve uygulanması amaçlanmıştır. Bunlara paralel bir diğer çalışma ise, Türkiye Türkçesinin sesbilimsel özelliklerinin incelenmesi olmuştur. Deneysel Türkçe ses veri tabanı, bu incelemeden elde edilen bilgilerden yararlanılarak oluşturulmuştur. Bu çalışmaların sonucunda Türkçe deneysel bir gerçek zamanlı sesli ifade tanıma sistemi modelinin oluşturulması amaçlanmıştır.

SİTLab'ın donanımsal görünümü Çizim 7.1'de verilmiştir. Labaratuvar, birbirlerine *Ethernet* ağı üzerinden bağlı bir *HP-720 Apollo Workstation* ve iki mikrobilgisayar ile bunlara bağlı çevre birimleri ve kimi ek donanımlardan oluşmaktadır. Labaratuvar ayrıca Bilgisayar Bilimleri-Mühendisliği ve HÜ Beytepe Kampüsü yerel ağına bağlanmıştır. Bu biçimde dış dünya ile iletişim olanağı kazanılmıştır. Bu altyapı sayesinde sesli ifade tanıma konusunda güncel program ve belgeler elde edilebilmektedir. Özellikle sesli ifade tanıma için geliştirilen kimi yazılımlar, örnek veri tabanları, nöron ağ benzetim yazılımları ve teknik raporlar bu yolla edinilmiştir. SİTLab'a ilişkin donanım ve yazılım altyapısı aşağıda özetlemiştir:



Çizim 7.1. Sesli İfade Tanıma Labaratuvarının (SİTLab) genel görünümü

7.1. Donanım altyapısı

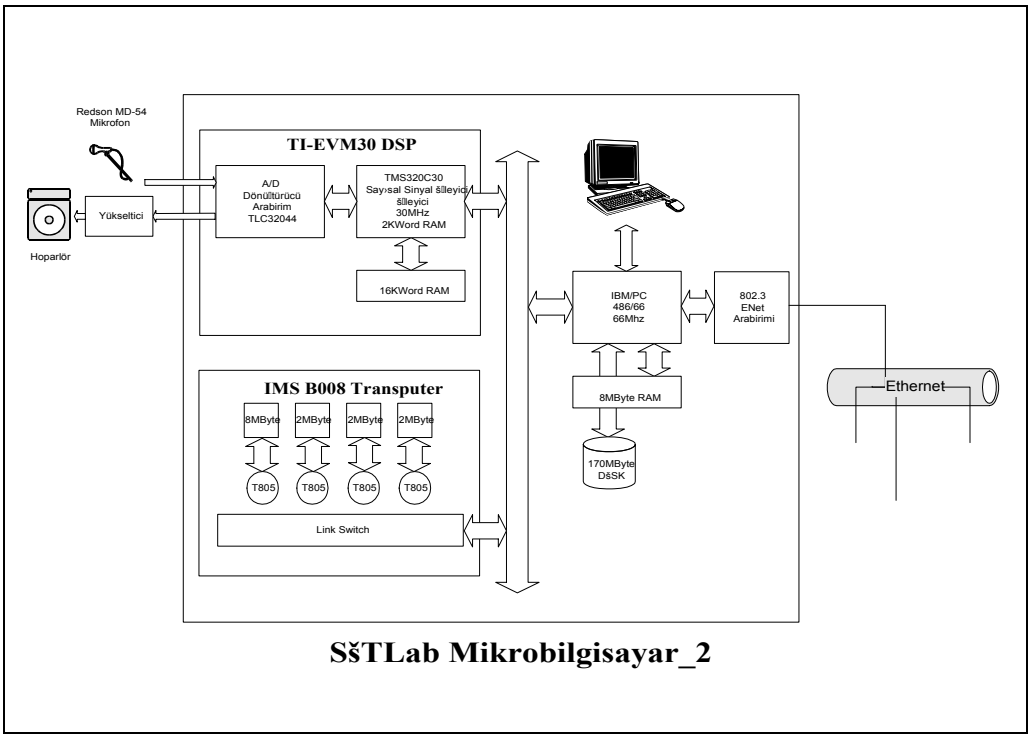
SİTLab'ın donanımsal altyapısı şunlardan oluşmaktadır:

- *Intel486* tabanlı *DOS* ve *LINUX* işletim dizgelerinin kullanıldığı kişisel bilgisayarlar. Bu sistemler *MS-Windows* ve *X-Windows* grafik arabirimi ile desteklenmektedir.

- *PA-RISC (HP 720)* tabanlı iş istasyonu. Bu sistem *Unix (HP-UX)* işletim sistemini kullanmaktadır. Üzerinde grafik kullanıcı arabirimi olanağı sağlayan *X-Windows* grafik arabirimi bulunmaktadır.

- *TI TMS320C30* Sayısal Signal İşleyici kartı ile sesli ifadenin sayısallaştırılması, işlenmesi ve özellik vektörlerinin elde edilmesi sağlamaktadır. Bu tür işleyicili sistemlerde kullanılmak üzere *SPOX* gibi özelleştirilmiş bir işletim sistemi kullanılabilmektedir. Ancak bu çalışma kapsamında *DOS* ve *LINUX* işletim sistemi ortamı program geliştirmede ve işleyicinin yerel belleğine programın aktarılıp çalıştırılmasında kullanılmaktadır.

- *Inmos T805* işleyicilerden oluşan *Transputer* kartı, Bu donanım *çok görevli ortamı* sağlamaktadır.



Çizim 7.2. Gerçek zamanlı sesli ifade tanıma geliştirme sisteminin genel görünümü.

SİTLab'da iki mikrobilgisayardan birisi üzerinde sesli ifadelerin sayısallaştırılması ve işlenmesi amacıyla Sayısal Sinyal İşleme kartı bulunmaktadır. Başlangıçta bu kart yalnız sesli ifadelerin sayısallaştırılıp saklanması amacıyla kullanılmıştır. Bir sonraki aşamada, başarımları ve uygunlukları sınanan kimi sinyal işleme programlarının kartın üzerindeki işleyicide çalıştırılması ile oldukça hızlı bir çalışma ortamı elde edilmiştir. Diğer mikrobilgisayar üzerinde ise paralel işleme taban oluşturacak yapı kurulmuştur. Bu nedenle ikinci mikrobilgisayar sistemi üzerine 4 *Inmos-T805 Transputer* sistemi bulunan kart takılmıştır. Toplam 20MBayt'lık yerel belleği ile

paralel işleme olanak sağlayan bu alt yapı üzerinde deneyim elde edilmeye ve bu ortamın daha sonraki bir aşamada sesli ifade özellik vektörlerini kullanan tanıma programların işletilmesinde kullanılması amaçlanmıştır. Çizim.7.2'de, planlanan sesli ifade tanıma sisteminin genel görünümü verilmiştir.

7.2. Yazılım Altyapısı

SİTLab'da oluşturulan yazılım birikimi sesli ifade özellik çıkarımı ve tanıma programları, sesli ifade örnek veri tabanları, sayısal sinyal işleyici, *transputer,* nöron ağ benzetim programları ve diğer yardımcı programlar başlıkları altında toplanabilir. Bunlardan kimi önemli sayılabilecekler şunlardır:

7.2.1. Sesli ifade özellik vektörü çıkarımı ve tanıma programları

Mathematica 1.0 ve 2.0 için Sayısal Sinyal işleme algoritmaları

MS-Windows Apple ve Unix altında çalışan *Mathematica 2.0* ve üzeri için *The Georgia Tech Research Corporation*'da yazılan sayısal sinyal işleme modeli oluşturmada kullanılabilinen simgesel sinyal işleme kitaplığıdır (Brian L.Evans ve James H. McClellan, *A Brief Introduction to the Signal Processing Packages*.).

spctools.tar.Z

MS-Windows, Apple ya da *Unix* altında çalışan *MathLab* için geliştirilmiş sayısal sinyal işleme algoritma paketidir. *Recnet* ile uyumlu olması nedeni ile önemlidir.

Recnet

Unix altında çalışan *DARPA TIMIT* ve *RM (Resource Management)* adlı sesli ifade veri tabanları üzerinde çalışan sesli ifade tanıma yazılımıdır. Yazılım dört ana kesimden oluşmaktadır. Bunlar:

- Ön işleyici. Birçok standart ve ek ön işlem kesimini içermektedir.
- *Recurrent* Ağ türü nöron ağı tanıyıcı kesimi ve parametre kütükleri (İngilizce için nöron ağı ağırlık matrisi),
- *Markov* model tabanlı *phon* ve sözcük tanıyıcı kesim,
- Sonuç oluşturma kesimi (*Dynamic programming*)

CookBook

Dr. Tony Robinson tarafından *recnet* adı ile anılan yazılımın genel amaçlı kütüphaneye dönüşmüş biçimidir. Bu kütüphanede klasik özellik çıkarma işlemlerinin yanı sıra;

- *dtw* (*Dynamic Time warping*)
- *endpoint* (Sesli ifadenin başlangıç ve bitişini bulmaya yönelik yazılım)
- *lpccoder* (*linear predictive coder*)
- *pitch tracker* adlı yazılım kesimleri bulunmaktadır.

Sürekli üzerinde çalışılan ve sıkça günlenen bir yazılımdır.

hmm-1.0.tar.Z

Hidden Markov modelinin gerçekleştirildiği program paketidir. Kullanıma kolaylıkla geçirilebilecek türden bir yazılımdır.

OGI Speech Tools (Oregon Graduate Institute Speech Tools)

Sesli ifade verileri üzerinde analiz yapmaya yönelik olarak geliştirilmiş bir programdır. Yaygın bir biçimde sesli ifade araştırmalarında veri işleme amacıyla kullanılmaktadır. *X-Windows* alt yapısını kullanan *LYRA* adlı grafik arabirimi bulunmaktadır. Program sesli ifadelerin görüntülenmesine ilişkin üç gösterim biçimi sağlamaktadır. Bunlar:

- Sesli ifadenin kendisi,
- Sıklık ekseninde *spectrum*,
- *Phon*-Fonem-sözcük etiket sınırları.

Program etkileşimli olarak sesli ifadenin etiketlenmesinde kullanılmaktadır. Bunun dışında,

- *PLP* Analizi (*Perceptual Linear Predictive Analysis*),
- *RASTA PLP* Analizi,
- *Linear Predictive Coding*,
- *Mel Cepstrum Coding,*
- *Fast Fourier Transform*

gibi işlemler C dilinde yazılmış yordam kütüphanesince sağlanmaktadır. Program farklı formattaki kütükleri kullanabildiği gibi bu kütükler arasında dönüşümü de sağlayabilmektedir. Bu formatlar şunlardır:

- *ADC*, *WAV,*
- *NIST SPHERE* (*National Institute of Standards in Technology SPeech Header Resource function*),
- *mu-law*,
- İkili değerler(*Binary*), ASCII dir.

Programda *find-phone* adlı ses veri tabanı için gerçekleştirilen arama programı da bulunmaktadır. *PEARL* adlı bir dil ile *Unix* ortamında diğer programlar içinden de kullanılabilmektedir.

pitch-tr.tar.Z

İngilizce fonem sınırlarının bulunmasında kullanılan bir yazılımdır.

rasta.tar.Z

Sesli ifadelerden özellik vektörü oluşturmada kullanılan yazılımdır. Sesli ifadenin özellik vektörüne dönüştürülmesinde kullanılan pencere genişliği, kayma süresi, örneklem sıklığı gibi parametreleri denetlenebilmektedir.

findphone.tar.Z

İngilizce sesli ifade içinde fonemleri bulmayı amaçlayan bir yazılımdır. Diğerlerine göre, açıklama ve başarım açısından zayıf bir yazılımdır.

lutear.tar.Z

Sesin duyma sürecinin benzetiminin yapıldığı bir programdır (kulak modeli).

tdnn.tar.Z

Time Delay Neural Network için *Fortran* ve C ile yazılmış algoritmaların bulunduğu sözcük tabanlı sesli ifade tanıma programıdır.

vowel.c

Tony Robinson tarafından *Single Layer Perceptron*, *Multi Layer Perceptron*, *Modified Kanerva Model*, *Gaussian Node Network*, *Square Node Network*

tekniklerinin kullanıldığı, konuşmacıdan bağımsız İngilizce ünlü ses tanıma ve başarım ölçme programı olarak yazılmış bir programdır.

xml-demos.tar.Z

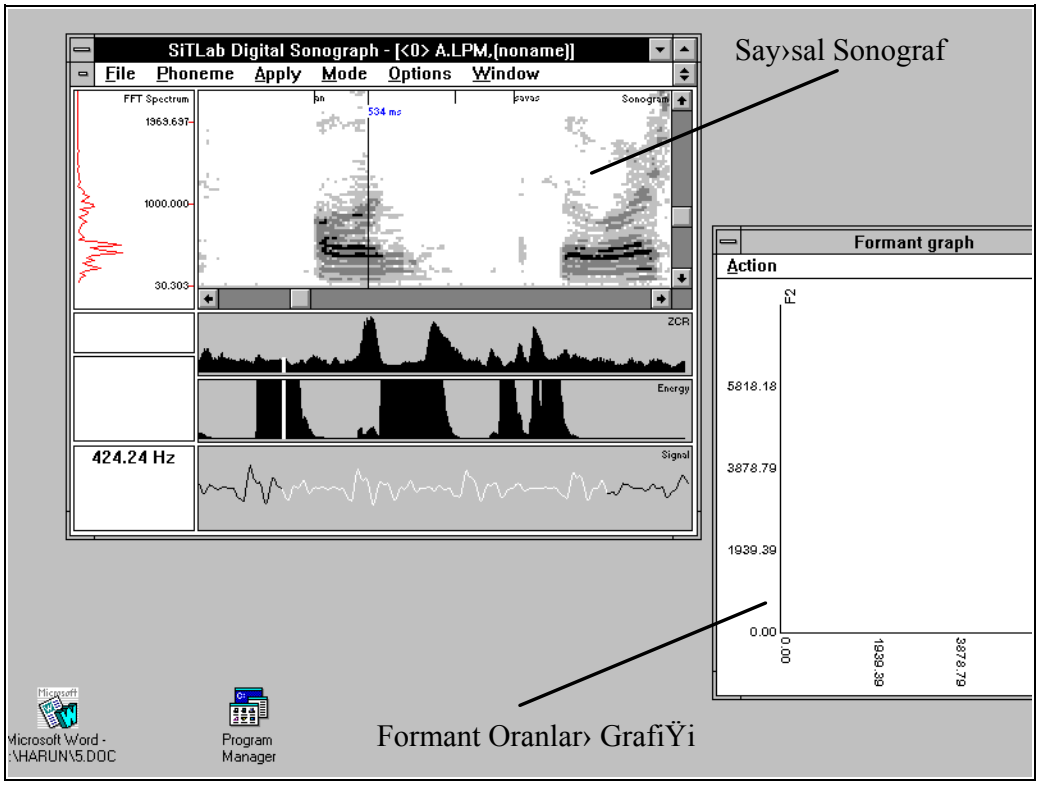
Lisp dilinde sinyal işleme programları bulunmaktadır.

Sayısal Sonograf

Sayısal Sonograf yazılımı *MS Windows* grafik ortamda çalışabilecek biçimde, SİTLab'da Ferhat SAVCI tarafından Yüksek Mühendislik tezi olarak tasarlanıp gerçekleştirilmiştir (Savcı 1994). Yazılım mekanik sonografların tüm işlevlerini içermektedir. Kullanım esnekliği ve hız açısından mekanik sonograflardan üstündür.

Bu yazılımın Proje içinde iki temel amacı bulunmaktadır. Bu amaçlardan ilki, sesli ifade sinyallerinin bilgisayar ortamında etkileşimli olarak incelenebilmesi, ikincisi ise, gerçek zamanlı sesli ifade çözümlemeye altyapı sağlanmasıdır. Sayısal sonograf, *Microsoft Windows 3.1* ortamında çalışacak biçimde geliştirilmiştir. Yazılımın menü tabanlı ve kullanımı kolay olması amaçlanmıştır. Yazılımın kullanıcı arabiriminin tasarımında sonograf kullanmış uzman sesbilimcilerin görüşleri etkili olmuş, sıradan dilbilimciler tarafından kullanılır bir yazılım niteliği içermesi amaçlanmıştır.

Yazılım, sesli ifadeleri zaman ve sıklık boyutunda, karşılaştırmaya olanak verecek biçimde görüntüleyebilmektedir. Bunun yanı sıra sesli ifadelerin zamana göre, *zero crossing,* enerji ve sıklık eksenindeki grafikleri çizdirilmektedir. Çizim 7.3'de sayısal sonografın genel görünümü, Çizim 7.4'de ise sayısal sonograf yazılımının ekran kesimleri verilmiştir.

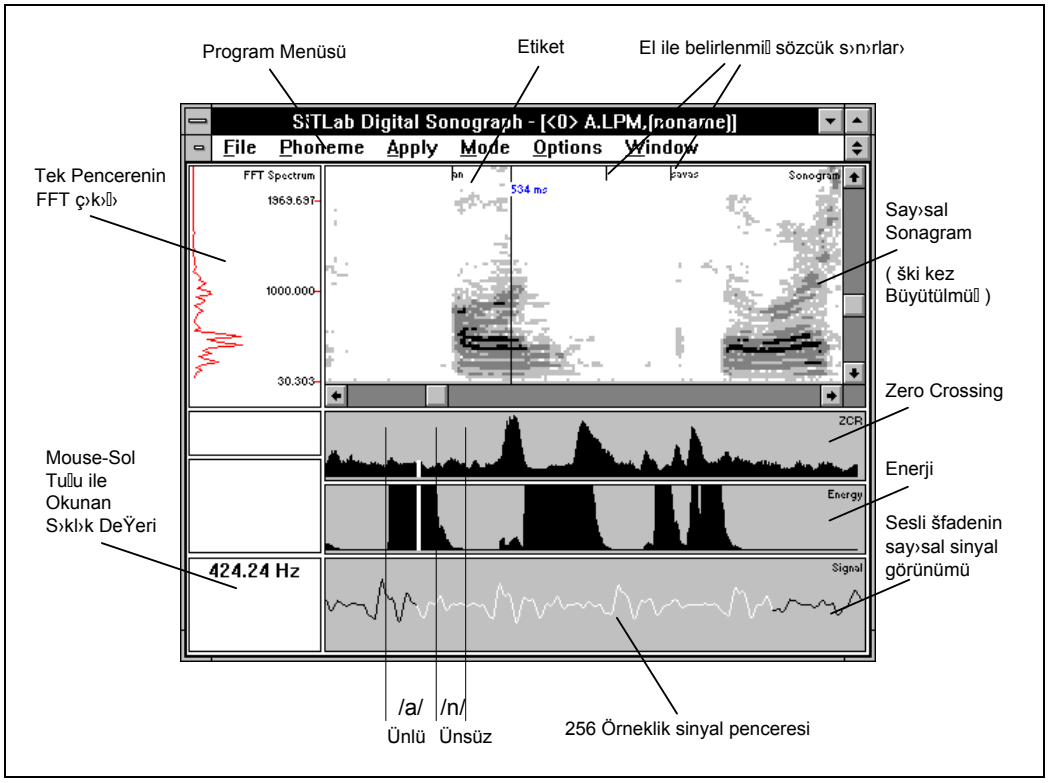


Çizim 7.3. Sayısal Sonograf'ın genel görünümü.

Programda kimi denetlenebilir özellikler şunlardır:

- Pencere genişliği 128, 256 ya da 512 örneklem olarak seçilebilmektedir.
- Pencerelerin örneklem sayısı türünden örtüşme uzunluğu, 8 - pencere boyu arasında bir değere kurulabilmektedir.
- Sinyalin sıklık boyutuna dönüştürülmeden önce filtrelenebilmesi ve filtre seçimi yapılabilmektedir. *(Bartlett, Blackman, Hamming, Hanning, Kaiser* ve dikdörtgen filtrelerinden biri seçilebilmektedir.)
- Sıklık ekseninde çalışmanın daha ayrıntılı biçimde yapılabilmesini olanaklı kılan sıklık penceresinin büyütülebilme özelliği bulunmaktadır (2 ya da 4 katı büyütme olanağı sağlanmaktadır).
- Sesli ifadelerin temel formant sıklık oranlarını saptamada ve gösteriminde kullanılan program kesimi bulunmaktadır.
- Sinyale ilişkin sıklık ve zaman bilgilerinin sayısal biçimde okunabilmesi sağlanmaktadır.

- Fonem sınırlarının elle belirlenebilmesi özelliği sağlanmıştır. Belirlenen sınırlar içinde kalan fonemlere ilişkin etiketlerin verilmesi ve kütüklere bu etiketler ile birlikte yazdırılması sağlanmaktadır.
- Sesli ifadelerin elle belirlenen sınırlar içinde kalan kesimlerinin, *Fast Fourier Transformation* ya da *Linear Predictive Coding* işlemleri gerçekleştirildikten sonra *FFT, LPC* ya da *Mel* skalasında *Cepstrum* değerleri ile saklanabilmesi olanaklı kılınmıştır.



Çizim 7.4. Sayısal Sonografın örnek ekran görünümü

Sayısal sonograf yazılımının bütününe yakın kesimi gerçek zamanlı uygulama amacıyla *EVM-30 DSP* kartı üzerine aktarılmış ve daha önceden belirlenen uygun parametre değerleri ile çalışması sağlanmıştır. Bu biçimde sesli ifadelerin, yaklaşık her 17milisaniyede bir özellik vektörlerinin oluşturulabilmesi sağlanmıştır.

rsynt.0.9.tar.Z

İngilizce sesli ifade oluşturma algoritmalarını içermektedir.

klatt.3.01.tar.Z

Parametrik sesli ifade oluşturma programıdır. Metinden ses üretmeyi amaçlayan bir programdır. İngilizce için gerçekleştirilen bu programda yazılı metin önce fonemlere dönüştürülür. Seslendirmede *coarticulation* etkisi de ele alınmaktadır.

7.2.2. Sesli ifade veri tabanı örnekleri

Sesli ifade veritabanlarına ilişkin bilgi *LDP* (*Linguistic Data Consortium*)'dan sağlanmıştır. Bu veri tabanlarından Türkçe korpus hazırlamada yararlanılmıştır. İncelenen veri tabanları şunlardır:

- *DARPA Resouce Management RM1, RM2*,
- *DARPA Acoustic Phonetic Cont. Speech. Comp*. *TIMIT, NTIMIT*
- *Texas* Sayı korpüsü *TIDIGITS (The TI Connected Digits speech corpus)*
- Hava Trafik Sistemi (*The Air Travel Information System*) *ATIS0, ATIS2, ATIS3*, *(The Multi-site ATIS speech corpus ATIS2*)
- *ARPA Wall Street Journal WSJ0, WSJ1*,
- *Texas* 46 sözcüklük *TI-46* korpus, *(The TI 46-word Isolated Word speech corpus*)
- Kredi kartı ve telefon danışma hizmetleri ses veritabanları
- Sürekli sesli ifade - *Wall Street Journal* sesli ifade korpüsü (*WSJ-CSR*).

DFKI-NLSR.ps.Z

Sesli ifade tanımaya ilişkin kimi kaynakların listesi ve kısa özetlerinin toplandığı kütüktür. *National Language Software Registy*'nin 2.1 inci raporundan alınmıştır. Raporda Sesli ifade tanımanın ileri adımlarına ilişkin (sözdizim, anlamsal) yapılan çalışmalardan ve uygulamalardan da söz edilmektedir.

mitll.csrtools.tar.Z

Çeşitli ses veri tabanı yapılarının ve genel bilgilerin bulunduğu kütüktür.

7.2.3. TI TMS320C30 Sayısal Sinyal İşleme Geliştirme kartı için programlar

Texas Instrument firmasının *TMC320C30 DSP* işleyicisi için yazılmış çeşitli sinyal işleme programları *ti.com:/pub/tms320bbs/mirrors* adlı kaynaktan edinilmiştir. Bu programlar çoğunlukla *DSP* üzerinde gerçek zamanlı uygulamalarda kullanılmak

üzere simgesel dilde yazılmış (*FFT*, *IFFT*, *LPC* ve Filtre dizisi gibi) yordamlar biçimindedir .

COFF (*Common Object File Format*) yükleyici, *COFF Dump* gibi sınama ve geliştirmeye yönelik programlar.

TMS320C30 için Simgesel ve C dili için derleyiciler. Bu derleyicilerin yanısıra kimi yardımcı ve örnek programlar.

Texas Instrument firması tarafından özellikle derleyicilerine yönelik hata raporları.

Sözkonusu yazılımların kullanımı ve tasarım biçimleri üzerine bilgi birikimi oluşturulmuştur. Bu birikimden Türkçe sesli ifade tanıma sisteminde büyük ölçüde yararlanılmıştır.

Bu yazılımlar dışında her iki mikrobilgisayar üzerinde *Microsoft-Windows 3.1* grafik tabanlı çalışma ortamı kurulmuş ve sesli ifadelerden elde edilen verilere ilişkin grafikler, analizler hazır kimi paket programlar kullanılarak yapılmıştır.

7.2.4. Nöron Ağı benzetim programları,

NeuralShell

Version 2.1-3.5 SPANN Lab. (*Signal Processing and Artifical Neural Networks*) The *Ohio State University* 1992'de yazılmış *X-windows* ortamında kullanılabilen genel amaçlı nöron ağı yazılımıdır.

RCS.v4.2 Rochester Connectionist Simulator.

Unix ortamında *X-Windows* arabirimini kullanan Nöron ağı benzetim programıdır. Türünün en eskilerinden olması nedeni ile önemlidir.

xerion

Unix ortamında, *X-Windows* arabirimini kullanan nöron ağı benzetim programıdır.

genesis

Unix ortamında, *X-Windows* arabirimini kullanan nöron ağı benzetim programıdır. Programda, daha çok, biyolojik nöronların çalışma ilkelerinin benzetimi

amaçlanmıştır. Nöronlar arası veri iletişimindeki kimyasal ve fiziksel değişimlerin benzetimi gerçekleştirilmektedir.

aspirin

Nöron ağı benzetim yazılımıdır. Yazılım *Unix* ortamlarında yaygın kullanılan kimi grafik ve çizim programlarını kullanmaktadır. Bu nedenle, özel bir grafik arabirimine sahip değildir.

planet

Kullanımı oldukça kolay nöron ağı benzetim programıdır. *X-Windows* grafik arabirimi ile oldukça kolay nöron ağı geliştirme ortamını sağlar.

art1(Adaptive Resonance Theory)

Rosenblat'ın *Adaptive Resonance Theory* adlı algoritmasının uygulamasıdır.

lvq (Learning Vector Quantizations)

Kohonen'in Gözetimli öğrenme algoritmalarının toplandığı yazılım paketidir.

som_pak-1.2tar.Z(Self Organizing Feature Map)

Kohonen'in Gözetimsiz öğrenme algoritmalarının toplandığı yazılım paketidir.

Bu tez kapsamında, sözü edilen yazılımlardan, özellikle *Planet* ve *Aspirin* üzerinde *Self Organizing Feature Map* ve *Time Delay Neural Network* benzetimleri gerçekleştirilmiştir. Ancak sesli ifade tanımada öğrenim aşamasının gerçek verilerle bu yazılımlar üzerinde oldukça yavaş gerçekleşmesi nedeni ile sözkonusu algoritmaların özel amaçlı olarak yeniden yazılması ve kullanılması yoluna gidilmiştir.

7.2.5. Diğer programlar

pvm3.2.6.tar.Z

Paralel Virtual Machine Yerel ağ üzerinden paralel çalışma ortamını yaratmada kullanılabilecek yazılım paketidir. Oldukça yalın yapısı ve açıklamaları ile ileriki aşamalarda sesli ifade tanımanın bir parçası olarak kullanılması düşünülmektedir.

Ptolemy

Sun, Mips, HP ve *Linux* üzerinde derlenmiş olarak bulunabilinen esnek, genel amaçlı bir prototip hazırlama ve benzetim yazılımıdır. Özellikle sinyal işleme, iletişim ve süreç denetim üzerindeki çalışmalarda yardımcı araç olarak kullanılabilecek özelliktedir. Oldukça büyük oylumlu bir program paketidir. Tüm paket açıldığında kullanım kılavuzları dahil yaklaşık 80MBayt yer kaplamaktadır. Bu yazılımın diğer önemli bir özelliği *DSP* işleyicilerine kod üretebilmesi ve kullanabilmesidir. Özellikle *DP56000* için bu olanak sağlanmıştır. SİTLab'da kullanılan *TMS320Cxx* işleyicisi için de bu olanağın sağlanması beklenmektedir.

Khoros

Genel amaçlı görüntü işleme yazılım paketidir. Ancak kütüphanesi sayısal sinyal işleme için gerekli tüm yordamları içermektedir. *Unix* ve *Linux* için kaynak ya da amaç program biçiminde elde edilebilmektedir. Oldukça kapsamlı ve büyük yer tutan bir programdır (yaklaşık 75MBayt).

Delaunay.tar.Z

Birleşen sayısı fazla olan vektörlerden oluşan kümelerin iki ya da üç boyutlu biçimde görüntülenmesinde kullanılan programdır. Bu gösterim biçiminin bir *benzeri Voronoi diagram* adı ile anılmaktadır. Bu program, *HP-UX* üzerinde *X-Windows* ortamında çalıştırılmıştır.

gavisual.tar.Z

Kohonen self organizing feature map sonuçlarının görüntülenmesinde kullanılan *X-Windows* uyumlu, *SGI* (*Silicon Graphics*) *sgx* kütüphanesi kullanılarak geliştirilmiş ancak daha sonra SİTLab'da *HP-UX* üzerinde kullanılır duruma getirilmiş program paketidir.

7.3. Türkçe Sesli İfade Verilerinin Hazırlanması

Türkçe sesli ifade tanıma sisteminde iki veri kümesi ya da Türkçe ses veri tabanı (korpüs) oluşturulmuştur. Bunlardan ilki, yalnız Türkçedeki ünlü fonemleri incelemek amacıyla ZÜZ (ünsüZ Ünlü ünsüZ) türü tek heceli 47 sözcükden oluşan

küçük boyutta bir kümedir (Çizelge 7.1). Bu korpüs ile ünlülerin ayrımı ve genel olarak fonem sınırlarının saptanması çalışmaları yapılmıştır.

Çizelge 7.1. Tek heceli sözcüklerden oluşan korpüs

Ünlü	Tek heceli sözcükler
/a/	kar, sar, baş, kas
/e/	tez, kes, bez, ses, kel, gel
/ı/	kır, kıs, sız,tır, sır, kış
/i/	tiz, kir, siz, şiş
/o/	kor, kol, son, koş, şoş
/ö/	dök, sök, söz, sör, sön, kör, göl
/u/	kuş, kur, buz, sun, tur, sur, kul
/ü/	tür, tüp, süz, sür, süs, sün, kül, kür

İkinci korpüs hazırlanırken Türkçenin fonemik dil olma özelliği (her fonemin bir harf ile temsil edilmesi) göz önünde bulundurulmuştur. Bunun için korpüste yer alan sözcükler, incelenecek ünlü ve ünsüz fonemlerin sözcükler içindeki konumlarını en iyi yansıtacak ve yeterli sayıda olmalarını sağlayacak biçimde seçilmeye çalışılmıştır. Bu korpüsün hazırlanmasında uyulan ilkeler şunlardır:

- Türkçe 139 ayrık sözcük kullanılmıştır (Bkz. EK.1). Sözcükler belli bir konu ya da amaca yönelik olarak seçilmemiştir. Örneğin telefon ile sayı girme örneğinde salt rakam adlarınının tanınması sözkonusu edilmektedir. Burada bu tür kısıtlamalar sözkonusu değildir.

- Sözcükler mümkün olduğu kadar kısalar arasından seçilmiştir. Genelde sözcükler bir ve iki heceden oluşmuştur. Bu yolla elle yapılan fonem sınırlarını belirleme işlemlerindeki hata oranı en aza indirilmeye çalışılmıştır.

- Türkçe sesli ifade tanımayı zorlaştıran ve diğer dillerden yaklaşım olarak farklılaştıran çekim ve hal ekleri korpüse katılmamıştır.

- Ünlü ve ünsüz fonemlerin hece yapısı içindeki, başta, ortada ve sonda gibi konumları gözetilmiştir. Ünlülerin başta, ortada ya da sonda olmasından dolayı genelde herhangi bir sorunla karşılaşılmamıştır. Ancak ünsüzler de başa ve sona gelme durumlarında aynı fonem değişik *phon*'larla seslendirilebilmektedir. Ayrıca ünsüzlerin, bir iki örnek dışında ortaya

gelmeleri de mümkün olmamaktadır. Bu nedenle ünsüzlerin hece ortasında olup da farklı bir *phon* oluşturması olasılığı da yoktur.

- Türkçe'nin hece türleri göz önüne alınmıştır. Türkçede bu hece türleri, ünsüzlerin ünlüler ile birlikte Ü, ÜZ, ÜZZ, ZÜ, ZÜZ, ZÜZZ biçiminde kullanılmaması ile oluşur.

- Korpüste sözcükler belirlenirken bunların Türkçede geçen kullanım oranları da göz önünde tutulmuştur. ÜZZ ve ZÜZZ türü heceler Türkçe sözcüklerde çok az kullanılmaları nedeni ile bunlara ilişkin örneklerin sayıları da haliyle az olmuştur. Çizelge 7.2 de, korpüs oluşturulurken göz önünde bulundurulan, Türkçedeki hece türlerinin kullanım oranları verilmiştir.

- Fonem olup olmadığı tartışma konusu olan *ğ*'nin yer aldığı örnekler korpüs dışında tutulmuştur. Bir başka çalışmada *ğ*'nin başlı başına bir fonem olup olmadığı, ya da diğer fonemlerin *phon*'larından mı oluştuğu ele alınmalıdır.

Çizelge 7.2. Türkçede hecelerin sözcük içindeki kullanım sıklıkları (Demircan 79)

Sözcük içindeki konumu	**Ü**	**ÜZ**	**ÜZZ**	**Z Ü**	**Z ÜZ**	**Z ÜZZ**
başta	%4,9	%4,2	%0,4	%52,9	%37,1	%0,5
sonda	%1,4	%0,6	%18,3	%38,9	%28,3	%12,5

Çizelge 7.3. Türkçede hecelerin sözcükler içinde yer alabildikleri konumlar(Demircan 79)

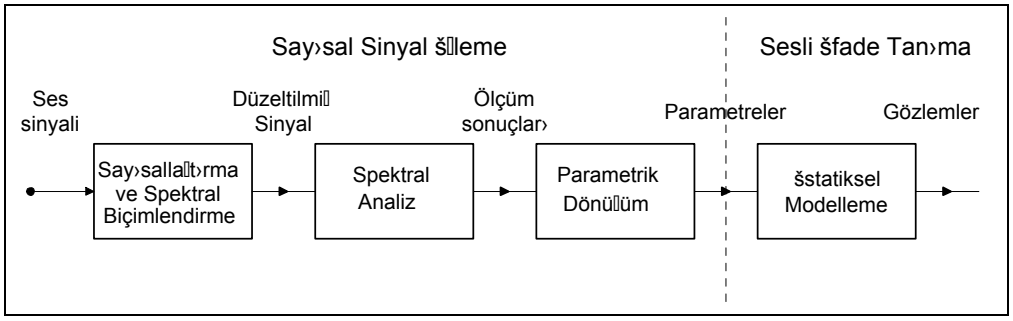
Sözcük içindeki konumu	**Ü**	**ÜZ**	**ÜZZ**	**Z Ü**	**Z ÜZ**	**Z ÜZZ**
Başında	BULUNUR	BULUNUR	BULUNUR	BULUNUR	BULUNUR	BULUNUR
Ortasında	Alıntı	BULUNUR	Alıntı	BULUNUR	BULUNUR	BULUNUR
Sonunda	BULUNUR	BULUNUR	Yok	Yok	BULUNUR	BULUNUR
Olası Hece Sayısı	8	160	3200	160	3200	64000

Çizelge 7.4. Korpüsün içerdiği sözcüklerin hecelerinin dağılımı

Ü	ÜZ	ÜZZ	Z Ü	Z ÜZ	Z ÜZZ
aşı, abajur, acı, efe, emek, erek, evet, ısı, iki, oku, otur, umut	an, ah ,af, ırmak, annem, on, örtü, altı, üç	üst, alt, ilk, erk	aşı, cadı, bab, çamur, fırça, koca, lamba, büro, baba, lamba, cadı, acı, koca, çamur, kaçık, demir, dere, efe, demir, sıfır, hasta, hayır, seher, ısı, iki, simit, yedi, jale, abajur, viraj, kirpi, sekiz, dokuz, sıcak, kalın, yorum, nasıl	çamur, fırça, lamba, ben, bir, çamur, kaçık, demir, da, erek, sen, ben, demir, fırsat, sıfır, güz, hasta, hayır, seher, ırmak, simit, abajur, viraj, kirpi, sekiz, dokuz, sıcak, kalın, dal, mal, umut, yorum, nasıl, annem, yüz, dokuz, sekiz, beş, bir, sekiz, ezmek, zor, tay, yüz, dev, evet, van, törpü, yüz, otur, yöntem, perde, kaplan, şarap, resim, fırça, sıfır, şarap, beş, tek, dokuz	genç, dört, kaçırt, tunç, dürt, utanç, kırt

7.4. Sesli İfadelerin Bilgisayar Ortamına Aktarılması ve İşlenmesi

Sesli ifadelerin bilgisayar ortamına aktarılmasından, tanınmaya hazır özellik vektörleri biçimine getirilene kadarki süreç bu kesimde açıklanacaktır.

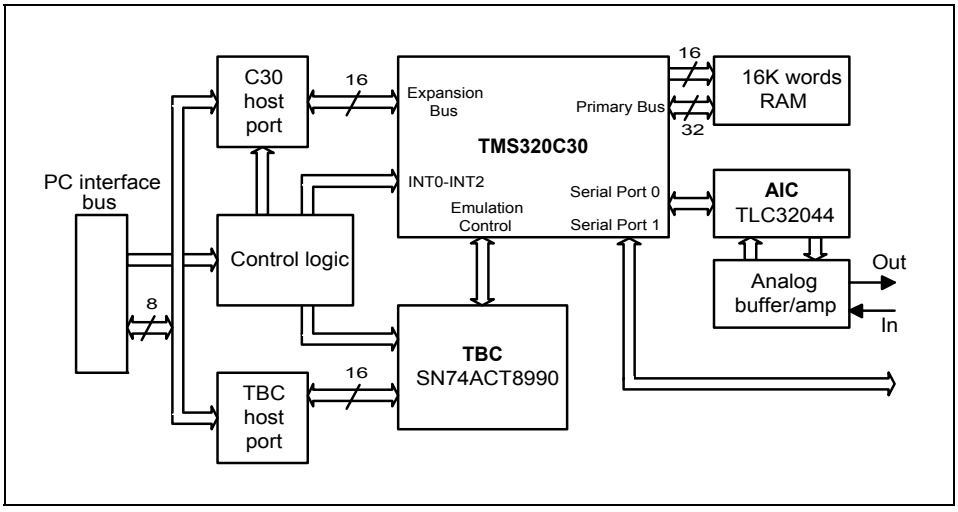


Çizim 7.5. Sesli ifade tanıma sisteminin genel görünümü.

Çizim 7.5'de sesli ifade tanımda sisteminin genel görümü verilmiştir. Bu çizimdeki ilk kesimler sesli ifade sinyallerinin sayısallaştırması ve tanınmaya hazır özellik vektörleri biçimine getirilmesi aşamalarını içermektedir.

7.4.1. Sesli ifadelerin Sayısallaştırılması

Sesli ifadelerin sayısallaştırılması amacıyla kişisel bilgisayarlar üzerine takılabilen üzerinde örnekselden sayısala dönüştürücü ve Sayısal İşleyici (*Digital Signal Processor - TMS320C30*) bulunan Texas Instrunment firmasının *EVM30* kartı kullanılmaktadır(Çizim 7.6). Sesli ifadelerin sayısallaştırılması *EVM30* kartı üzerindeki yine *Texas Instrument*'ın bir ürünü olan *TLC32044 AIC (analog interface controller)* adlı arabirimce sağlanmaktadır. *AIC*, programlanabilir özellikte bir arabirimdir. Üzerinde *anti-aliasing*, *high pass, (sin x)/x* filtreleri bulunmaktadır. Örneksel giriş düzeyinin +/-1,5 ya da +/-3,0volt arasında seçilebilmesi programla denetlenebilmektedir. Söz konusu giriş düzeyleri herhangi bir ön yükseltici devre kullanılmaksızın mikrofon bağlanabilmesini olanaklı kılmaktadır. Bu tez çalışmasında *Redson MD-54* stereo mikrofon doğrudan kartın girişine bağlanarak kullanılmıştır. *AIC* arabirimi aynı zamanda sayısal verileri örneksele dönüştürücü *(D/A)* çevrimi de içermektedir. *AIC*'in çıkış gerilimi ise 1,5*Vpp* dir. *EVM30* kartı üzerinde bu çıkış dış ortama aktarılmadan önce ayrıca yükseltilmektedir. Bu amaçla kart üzerinde *LM386* yükseltici birimi eklenmiştir. Eklenmiş olan bu yükseltici devre ile 6*Vpp* çıkış gerilimi sağlanabilmektedir.



Çizim 7.6. EVM30 kartının genel görünümü

AIC'in denetimi ve veri iletişimi *TMS320C30* üzerindeki iki ardıl bağlantıdan biri ile gerçekleştirilmektedir. *AIC*'in örneklem hızı ve giriş/çıkış filtre tanımları bu biçimde *TMS320C30* tarafından denetlenmektedir. Filtre ve örneklem hızına ilişkin parametre değerleri *TMS320C30*'un saat çıkış (*Master clock frequency-MCLK)* sıklığı ile birlikte hesaplanmaktadır. Bu saat sıklığı, 30MHz'lik ana saat sıklık değeri *TMS320C30* üzerinde dörde bölünerek 7,5MHz olarak üretilmekte ve doğrudan *AIC*

tarafından kullanılmaktadır. *AIC*'in *low-pass filtre* sıklık değeri ve örneklem hız tanımı için iki parametre kullanılmaktadır. *AIC'*in programlanmasındaki söz konusu *A* ve *B* adlı bu parametreler aşağıdaki bağıntılar kullanılarak oluşturulmaktadır: Burada *swithed-capacitive filter frequency (SCF)* değeri f_{SCF} ve örneklem hızı f_{MCKL} ile gösterilmektedir.

$$f_{SCF} = \frac{f_{MCLK}}{(2A)}$$
$$f_C = \frac{f_{MCLK}}{(2AB)} \qquad (7.1)$$

AIC'in örneksel dönüşüm hızı f_C 19,2 kHz ile sınırlıdır. Dönüşüm, *AIC*'in *sample and hold* girişine uygulanan saat sinyali ile sağlanmaktadır. *AIC*, örneksel sinyali kod uzunluğu 14 bit olan sayısal biçime dönüştürdükten sonra ardıl biçimde *TMS320C30*'ca alınması için kesilme istemi oluşturur. Bunun sonucunda üretilen kesilme *TMS320C30* üzerindeki kesilme yordamınca ele alınır. Bu yordamda 14 bitlik veriler, önceden belirlenmiş sayıya erişilene kadar bir dizide tutulur. Ancak bu sırada *TMS320C30*'un 32 bitlik bir işleyici olması nedeniyle gelen verilerin işaret ağırlıklı 32 bit'lik tamsayılara dönüştürülmesi de gerekmektedir.

EVM30 kartı bu tez çalışması içinde başlangıçta yalnız sesli ifadeleri sayısala çevirme ve ön filtreleme işlemlerini gerçekleştirmede kullanılmıştır. Daha sonraki aşamalarda kart, gerçek zamanlı özellik vektörü oluşturma uygulamalarında da kullanılmıştır.

EVM30 üzerinde bir diğer arabirim de *TBC (test bus controller)*dır. Bu arabirim *TMS320C30'*un mikrobilgisayar tarafından denetlenebilmesi amacıyla kullanılmıştır. Bu denetimler arasında işleyicinin *reset* edilmesi, yerel belleğine program yüklenip çalıştırılması ya da işleyicinin hata bulma *(debug)* amacıyla *single-step* işletilmesi yeralmaktadır. Mikrobilgisayarla iletişimi sağlayan *TBC* arabirimi, mikrobilgisayar karta veri aktardığında ve karttan veri okuduğunda iki ayrı kesilme üretmektedir. Bir diğer yordam, mikrobilgisayar veri okudu kesilmesini yönetir. Mikrobilgisayar üzerindeki yazılımın da, *EVM30* üzerindeki yazılımın kendisiyle zamanuyumlu çalışabilmesi için mikrobilgisayar karta veri aktardı kesilmesini yöneltmesi gerekmektedir. Bu kesilme üretildiğinde, *EVM30* üzerindeki modüller işlemlerini yeniden, bilinen bir noktadan sürdürür ve aralarındaki veri iletişimleri yeniden başlatılır.

TMS320C30 işleyicisinin kayan ayrımlı sayıların gösteriminde, standart dışı, kendine özgü bir tanımı bulunmaktadır. Bu nedenle işleyici üzerinden aktarılacak kayan ayrımlı sayılar, özellikle gerçek zamanlı uygulamalarda sorun çıkartmaktadır. Bu işleyici dışındaki bilgisayarlarda çoğu kez *IEEE 754* standartındaki kayan ayrımlı gösterim kullanılmaktadır. TMS320C30 için oluşturulmuş örnek bir dönüşüm algoritması Ek-2'de verilmiştir.

7.4.2. Spektral biçimlendirme

Bu kesimde sayısallaştırılan veriler üzerindeki ön işlemlerden söz edilecektir. Uygulamada sayısallaştırılan veriler iki türde saklanmaktadır. Bunlardan ilkinde EVM30 kartı yalnız 14 bitlik sayısallaştırcı olarak görev yapmaktadır. Bu kullanım biçiminde sesli ifadeler sayısallaştırıldıkları gibi 2 baytlık sözcüklerden oluşan biçimleri ile kartın bağlı olduğu bilgisayara aktarılmakta bu bilgisayarın diskine yazılmaktadır (*raw data*). Sayısallaştırma sırasında, sayısallaştırma sıklığı, *sin(x)/x* filtresi kullanımı, giriş genliği seçimi yapılabilmektedir.

Uygulamada karşılaşılan bir sorun da bu verilerin değerlendirilmek üzere okunmasında ortaya çıkmaktadır. Bu uygulama sorunu, verilerin 2 baytlık sözcüklerden oluşması ve bu baytların yerleşimleri ile ilgilidir. Sorun yanlış okuma ve değerlendirmeye neden olacağı için kullanılan bilgisayarın verileri alma özelliğinin bilinmesini gerektirir. İzleyen kesim bu ayrımın nasıl yapıldığını anlatmaktadır.

Sesli ifade verilerinin taşınabilirliğinin sağlanması *(LittleIndian-BigIndian)*

Uygulamada karşılaşılan basit bir sorun, SİTLab'daki bilgisayarların mimarilerinin birbirinden farklı olması nedeniyle, verilerin bir bilgisayardan diğerine aktarılmasında ortaya çıkan tanım uyuşmazlığıdır. Bu sorun verilerin sesli ifade kütüklerinde olduğu gibi, ikili olarak saklanıp aktarılması sırasında ortaya çıkmaktadır. Kısaca, bir tamsayının kütükten okunması bir bilgisayardan diğer bir tür bilgisayara değişebilmektedir. *Intel* türü işleyiciler 16 ikillik bir tamsayının küçük ağırlıklı baytını ilk bayt olarak okurken (*LittleIndian*), PA-RISC türü işleyicili bir İş istasyonunda işlem ters biçiminde gerçekleştirilmektedir (*BigIndian*). Değişik bilgisayarların aynı labaratuvar ortamında kullanılması durumunda bu tanım sorunun ortaya çıkmaması için yapılacak işlem, Çizim 7.7'deki gibi bir algoritmaya uyulmasıdır.

```
#include <stdio.h>
#include <stdlib.h>

int LittleIndian()
{
     char b[4];
     int *l = (int *)b;

     *l = 1;
     return( (int)b[0] );
}

main()
{
if(LittleIndian()==0)
     printf("\n BigIndian Machine \n");
else
     printf("\n LittleIndian Machine \n");
}
```

Çizim 7.7. *LittleIndian* ve *BigIndian* özelliği bulmada kullanılan algoritma örneği

***Preemhasis* filtre**

Sesli ifadeler düşük sıklık değerlerinde yüksek enerji değerlerine sahip olmaktadır. Düşük sıklıktaki enerji düzeylerinin yüksek sıklıktakilerini bastırmaması için düzeylerinin göreli azaltılması gerekmektedir. Bunu için *preemhasis* adı verilen filtre kullanılmaktadır. Bu filtre, $H(z)= 1- az^{-1}$ bağıntısı ile ifade edilebilir. Bağıntıdaki *a* katsayısı 0,0 ile 1,0 arasında bir değere sahiptir. Ancak uygulamada *a* katsayısı çoğu kez zaman kazanma amacıyla 0.0 olarak seçilmektedir. Çeşitli *a* katsayıları için sıklık ekseninin, *dB* genlik değerine oran grafiği Çizim 3.13'de verilmişti.

Center Clipping

Sesli ifade sinyalindeki düşük genlikli gürültülerin atılması işlemidir. Gürültü, genelde yüksek sıklıklı ve zayıf bir sinyaldir. Sesli ifade sinyalinin sıfır düzeyine

yakın kesimleri çıkarıldığında sesli ifadenin özelliği kaybolmamakta ve gürültülü kesim ortadan kaldırılmaktadır. Kesmeye taban alınacak düzeyin saptanmasında sayısal sonograf'tan yararlanılmıştır. Sonuç olarak orijinal sinyalin sıfır düzeyinden %3-5'lik kesiminin çıkarılması ile gürültünün istenmeyen etkisi azaltılmıştır. Bu değerler bilgisayarların bulunduğu bir oda için yeterli olmaktadır. *Center clipping* değerlerinin denetlenebildiği, uyum sağlayıcı yazılım kesimi düşünülmüş ancak işlem gücünü azaltacağı ve bellekte yer kaybına neden olacağı gerekçesi ile şimdilik uygulanmamıştır.

Pencereleme

Sesli ifadelerin her sayısal değeri *AIC* tarafından *TMS320C30*'da kesilme istemi oluşturmaktadır. İlk kesilme yordamında, alınan her veri belirli sayıya ulaşana kadar *TMS320C30*'un üzerindeki yerel belleğe alınmaktadır. Uygulamada pencere genişliği olarak sözü edilen bu sayı 256 olarak seçilmiştir. Bu biçimde 8.013kHz'lik örneklem hızı ile sesli ifadeler 31,9ms'lik kesimler içinde incelenebilmektedir. Pencere genişliğinin, ikinin katları biçiminde seçilmesi daha sonraki aşamada kullanılacak *FFT* işleminden önce ayrıca *zero padding* yapılmasını gerektirmediği için önemlidir. *FFT* işlemi ikinin katları olan pencere genişliğinde çalışabilmektedir. Pencere genişliği, 64, 128, 512 ya da 1024 olabilirdi. Ancak, 256'nın altındaki sayılar az veri ile işlem yapmayı ve bu nedenle tanıma aşamasındaki duyarlılığın azalmasına, üstündekiler ise duyarlılığı yükseltmeye karşın fazla bellek ve işlem gücünü gerektirdiğinden tercih edilmemiştir.

Pencereleme işlemi ile birlikte yapılan ikinci işlem bir sonraki pencerenin nasıl düzenleneceğidir. İkinci pencerenin ne kadar süre ya da kaç örnek sonra başlatılacağı diğer bir parametredir. Bir başka ifade ile pencerenin ne kadar kaydırılacağıdır. Saptanacak kaydırma miktarı ya da buna karşılık gelen sürenin bir pencerenin işlenme süresinden daha fazla olmamasını gerektirmektedir. Kaydırma miktarının az olması, işleyicinin, sesli ifadeleri işlemede işlem gücünün gereksiz yere artması ve sonuçta yetişememesine neden olurken, fazla tutulması sesli ifadelerin özelliklerinin belirlenmesinde bilgi eksikliğini ortaya çıkarmaktadır. Kaydırma miktarının uzun tutulmasında ortaya çıkan en büyük sorun kısa süreli ünsüzlerin belirlenmesinin zorlaşmasıdır. Bu bağlamda 256'lık bir pencerenin kaydırılma miktarı 8,013kHz'lik örneklem sıklığında 30 ile 120 arasında anlamlı olmaktadır. Çalışmada kaydırma

miktarı pencerelerin işlenme süreleri de gözönüne alınarak 120 olarak seçilmiştir. Bu sayı gerçek zamanlı uygulamada gerekli hızın elde edilebilmesine olanak sağlamaktadır. Bu biçimde, her 15ms'de bir, sesli ifadenin 31,9ms'lik kesimi pencere içine işlenmek üzere alınabilmektedir. Sayısal Sonograf üzerinde bu sayının değiştirilebilir özellikte olması sayesinde çalışmanın başlarında bu değerlerin özellik vektörlerine etkisi incelenebilmiştir.

***Hamming* filtresi**

Pencereleme ile birlikte düşünülmesi gerekli bir diğer işlem de *Hamming* filtreleme işlemidir. Pencere başı ve sonunda, örneklem değerlerinin sıfırmış gibi ele alınma durumunda kalınması *FFT* algoritmasında istenmeyen yüksek sıklık değerlerinin oluşmasına neden olmaktadır. Bu nedenle, sıklık ekseninde, *low-pass* filtre tanımının zaman eksenindeki biçimleri, pencere ile *convolution* işlemine tabi tutulurlar. Bu amaçla çeşitli *convolution* fonksiyonları kullanılmaktadır. İşlem zamanı açısından herhangi bir kısıtlama yoksa *Kaiser* fonksiyonu bu iş için parametrik özelliğinden ötürü en uygun olanıdır. Ancak uygulamalardaki yaygınlığı ve diğerlerinin etkilerinin büyük farklar oluşturmaması nedeni ile *Hamming* fonksiyonunun kullanımı tercih edilmiştir. Uygulamada 256 sayısal veriden oluşan pencere üzerine *Hamming* filtresi *convolution* çarpımı kullanılarak uygulanmaktadır. Sonuçta 15ms de bir sesli ifadenin 31,9ms'lik kesimleri, *Hamming* filtresinden geçirilmiş biçimde bir sonraki aşamaya iletilmektedir.

Zero padding

FFT dönüşümünde girişin, 128, 256, 512 gibi ikinin üssü sayıdaki sözcük dizisinden oluşması gerekmektedir. Bazı uygulamalarda sözcük sayısı bu değerin altında kalabilir ya da kalması istenir. Bu durumda *zero padding* olarak adlandırılan ve dizinin kalan kısmının sıfır değeri ile doldurulması işlemi gerçekleştirilir. Bu çalışmada *zero padding* işlemi işlemlere başlandığı anda ilk gelen pencere dışında sözkonusu edilmemiştir.

7.4.3. Spektral analiz

Spektral analiz bir sinyalin sıklık ekseninde incelenmesine verilen addır. Spektral analiz amacıyla üç teknik kullanılmaktadır. Bunlar Filtre dizileri*, Fast Fourier Transformation* ve *Linear Predictive Coding* teknikleridir. Bu tez kapsamında

bunlardan Filtre dizisi dışındaki tekniklerler kullanılmıştır. *Fast Fourier* dönüştürme tekniği ve uygulama içindeki kullanımı izleyen kesimde özetlenmiştir.

Filtre dizisi tekniği

Filtre dizisi tekniği, her biri belirgin bir sıklık aralığına atanan bir dizi filtrenin kullanımını gerektirmektedir. İncelenen sinyal dizide yer alan filtrelerin girişine paralel olarak uygulanır. Her filtre belirli bir sıklık aralığına kurulduğundan çıkışlarında o sıklık aralığına ilişkin spektral enerji elde edilir. Tez çalışması kapsamında filtre dizisi tekniği denenmiş ancak donanımsal alt yapının dış etkenlere fazlası ile duyarlı ve denetlenemez olması nedeni ile bu yol terk edilmiştir. *DSP* kartının temin edilmesinden sonra filtre dizisinin yazılımla oluşturulması olanaklı olmakla birlikte *Fast Fourier* dönüşümüne göre işlem gücü ve bellekteki yer kaybı açısından daha masraflı olacağı gerekçesi ile kullanılmamıştır.

Fast Fourier Transformation

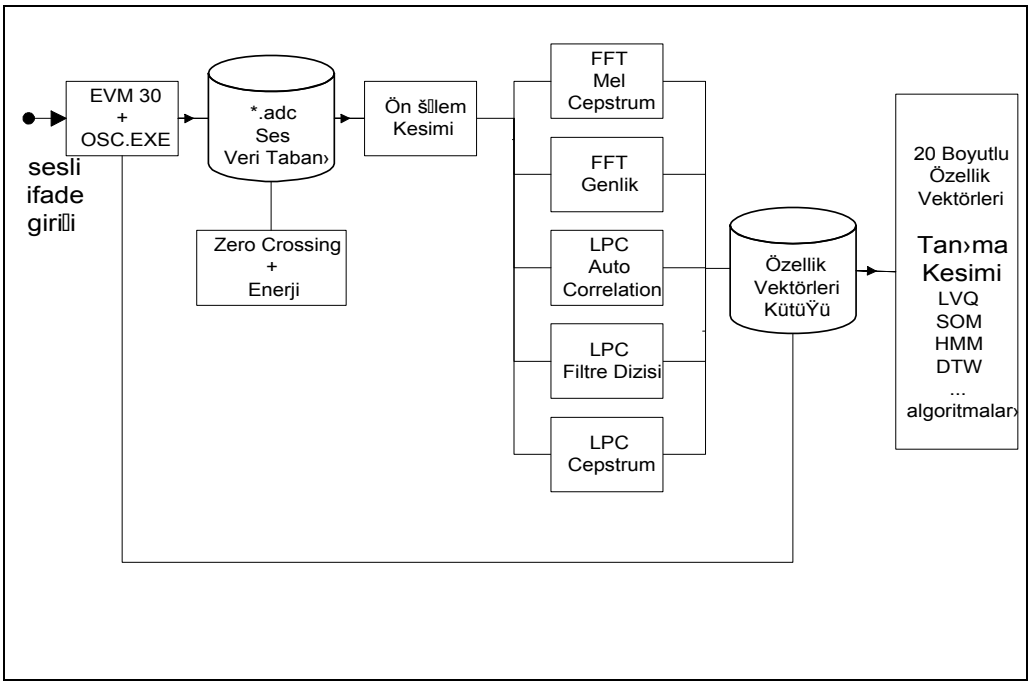
Fast Fourier dönüşümü, spectral analiz amacıyla sesli ifade sinyalinin sıklık bölgelerini ayırmada kullanılmaktadır. Uygulamada bu dönüşüm ile 31,9ms'lik (256 sözcüklük) sesli ifade sinyalininden eşit aralıklı (0- 4006kHz arasında) 128 sıklık bandındaki değerler hesaplanmaktadır. *Fast Fourier Transformation*, sinyal işlemede en fazla zaman kaybının olduğu bir kesimdir. Bu nedenle, dönüşüm algoritması simgesel dil ile yazılmıştır. Dönüşümde gerçel sayılarla çalışan, *radix-2 FFT* algoritması kullanılmıştır.

Linear Predictive Coding

LPC, spectral analiz amaçlı olarak, başlangıçta özellik vektörü çıkarmada kullanılmıştır. Ancak çeşitli denemeler sonucunda, gerçek zamanlı sistemde *DSP* üzerinde *FFT* dışında başka tekniklerin de çalıştırılması, ek işlem zamanına mal olacağı gerekçesi ile benimsenmemiştir. İleride ayrıntısı açıklanacak çalışmadan edinilen sonuca göre *FFT* kullanılarak oluşturulan *cepstrum* değerleri ile *LPC* parametrelerinden hesaplanan *cepstrum* değerleri arasında, özellik vektörleri olarak sesli ifadeleri temsil başarımları açısından aşırı farkların oluşmadığı gözlenmiştir. Bununla birlikte, mel skalasındaki *FFT*'den elde edilen *cepstrum* değerleri diğer özellik vektörü tanımlarından daha başarılı bulunmuştur.

7.4.4. Parametrik dönüşüm

Sesli ifade kesimlerinin temsil niteliği yüksek parametreler ile ifadesi kimi alt adımlarda gerçekleşmektedir. Bu adımlardan ilki, ifadeyi oluşturan seslerin anlam taşımayan kesimlerden ayrılması ya da başlangıç ve bitiş sınırlarının belirlenmesidir (*segmentation*). Bu biçimde ayıklanan sesli ifadenin özellik vektörleri, tanımada kullanılabilmektedir. Bunun için sınırları belirlenen sesli ifadelerin özellik vektörleri farklı yöntemlere dayalı olarak elde edilmiş ve karşılaştırma amacıyla toplu olarak bir çizelgede verilmiştir. Bu kesimde önce sesli ifade sınırlarının bulunması daha sonra çeşitli özellik vektörleri çıkarmada izlelen yöntemler tartışılmıştır.



Çizim 7.8. Sesli ifadelerden özellik çıkarma sürecinin genel görünümü.

Sesli ifade sınırlarının belirlenmesi

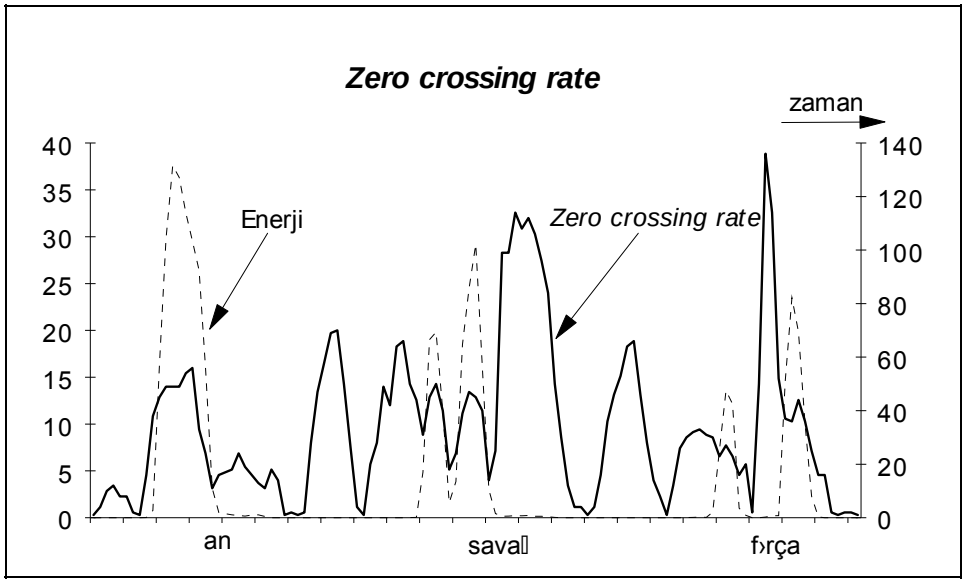
Sesli ifade tanımada zor olmakla birlikte yerine getirilmesi gereken bir işlem, kademeli olarak önce sözcük daha sonra fonem sınırlarının (başlangıç ve bitiş noktalarının) belirlenmesidir. Genelde her iki adımda da *zero crossing rate* ve enerji değerlerinden yararlanılmaktadır. Çalışmada, sözkonusu bu sınırların belirlenmesinde iki farklı yöntem kullanılmıştır. Bu yöntemlerin her ikisi de *zero crossing rate* ve enerji fonksiyonlarını kullanmaktadır. Yöntemlerden ilkinde, sözcük ve fonem sınırlarının sayısal sonograf üzerinde el ile, sesli ifade spektrumu

üzerinden belirlenmesi gerçekleştirilmektedir. Belirlenen sınırlar arasına sesli ifadeye karşı gelen sözcük, hece ya da fonem yazılarak sesli ifadeler etiketlenmektedir. Sayısal sonograf yazılımı üzerinde belirlenen bu kesimler arasındaki sinyale, mel skalasında *FFT*'ye dayalı *cepstrum* ya da *LPC* değerlerine dönüşüm sağlanır. Dönüşüm sonuçları daha sonra kullanılmak istenirse bir kütüğe yazdırılabilir. Diğer yöntemde ise bu belirlemenin gerçek zamanda yapılması gerçekleştirilir.

El ile yapılan sınır belirlemesinde, öncelikle sinyalin ünlü ya da ünsüz bir sese ilişkin olup olmadığı saptanır. Etiketi de oluşturulan fonem sinyalinin belirlenen sınırları arasında kalan kesimleri için parametre üretilir. Bilindiği gibi ünlüler, ses telleri titreştirilerek oluşturulan ve süreleri diğerlerine göre uzun seslerdir. Bu özelliklerinden dolayı diğer seslerden ayrılmaları daha kolay ve kesin olmaktadır. Ünsüzler için aynı şeyleri söylemek oldukça zordur. Sesli ifade tanımayı güçleştiren nedenlerden biri ünsüzlerin sözcük ya da hece içindeki konumunun belirlenmesindeki zorluktur.

Sayısal sonogram üzerinde sözcük sınırlarının belirlenmesi işlemine benzer bir yöntem gerçek zamanlı işlemlerde kullanılan algoritmaya taban oluşturmuştur. Algoritma, 3.2.6'ıncı kesimde, *başlangıç ve bitiş noktalarının belirlenmesi* olarak anlatılan biçimde ele alınmıştır. Kısaca, önce enerji değerinin değişimleri gözlenmektedir. Bu değerin yükselmesi ile bir ünlünün başlangıç noktası bulunur. Ancak ünlüden önce gelen ünsüz için *zero crossing rate* değerinin yükseldiği ve belirli bir süre yüksekliğini koruduğu kesim bulunur. Bitiş noktası da benzer bir yöntem izlenerek saptanır. Yöntemde ek olarak sesli ifadenin bittiği sanılan kesiminde kaybedilen veri olmaması için en az üç pencerelik veri kesimi de sesli ifade tanıma içine katılmaktadır.

Tez çalışmasında, sayısal sonografın yanı sıra, aynı işlevleri *Unix* üzerinde yerine getiren algoritmaların kullanımına gidilmiş ve sesli ifade sinyallerinden elde edilen *cepstrum* değerleri, enerji ve *ZCR* değerleri ile birlikte bir kütüğe yazdırılmıştır. Daha sonra bu kütükte yer alan değerlere, bir tablolama programı kullanılarak sayısal sonografla yapılana benzer bir biçimde sınır belirleme ve etiketleme işlemi uygulanmıştır. Sınır belirlemede yine enerji ve *ZCR* değerlerinin değişimleri kullanılmıştır. Ayrıca, bir başka kütükte salt etiketlenen alanlara ilişkin özellik vektörleri saklanmıştır. Bu kütük ileride anlatılacağı üzere sistemin eğitilmesi amacıyla oluşturulmaktadır.



Çizim 7.9. Bir sözcük sinyaline ilişkin örnek
enerji *(average magnitude)* ve *zero crossing rate* değerleri

Çizim 7.9 ve 7.10'da, *zero crossing rate* ve enerji değerlerinin grafiği verilmiştir. Grafikte üç sözcük için enerji ve *ZCR* değişimleri görülmektedir. İzleyen kesimde sınırların belirlenmesinde kullanılan *zero crossing rate* ve enerji fonksiyonun hesaplanmasındaki işlemler özetlenmiştir.

Zero Crossing Rate

Ses sinyalinin sıfırdan geçiş sayısı, *zero crossing rate* olarak bilinir. Kayıtlarda ses sinyalinin bulunmadığı kesimlerde bu sayı gürültünün yüksek sıklıkta bir sinyal olmasından dolayı artmaktadır. Sesli ifadelerin yer aldığı kesimlerde ise, bu değer düşük olmaktadır. Bu özellik, sesli ifadelerin başlangıç ve bitiş noktalarını belirlemede kullanılmaktadır.

Zero crossing rate hesaplama yöntemi bir sinyalin sıklığını belirlemede yararlanılan basit bir yöntemdir. Örneğin bir sinüssel sinyalde her period da iki sıfır geşişi bulunmaktadır. Bu gözlemden kalkarak *zero crossing rate* değerinden sinyal sıklığına doğrudan geçilebilmektedir. Periyodik bir sinyal üzerinde *zero crossing rate* değeri sıklık değerini kesin bir biçimde elde etmeye olanak verirken, periyodik olmayan ses sinyali gibi sinyaller için kestirimden söz edilmektedir.

Zero crossing rate değerinin bulunmasında izlenen yol, matematiksel olarak aşağıdaki gibidir:

$$Z(k) = \sum_{i=-\infty}^{\infty} \left| \text{sgn}\left[x(i)\right] - \text{sgn}\left[x(i-1)\right] \right| w(k-i) \tag{7.2}$$

Burada:

$$\begin{aligned} \text{sgn}\left[x(i)\right] &= +1 \quad \text{eğer} \quad x(i) \geq 0 \\ &= -1 \quad \text{eğer} \quad x(i) < 0 \end{aligned} \tag{7.3}$$

ve

$$\begin{aligned} w(k) &= \frac{1}{2N} \quad \text{eğer} \quad 0 \leq k \leq N-1 \\ &= 0 \qquad \text{diğer durumlarda} \end{aligned} \tag{7.4}$$

Enerji

Sesli ifadenin başlama ve bitiş noktasının belirlenmesinde çoğu kez *zero crossing rate* değeri ile birlikte kullanılan bu parametre sesin siddetine bağımlıdır. Sesin enerjisi, *sort time energy* olarak adlandırılan biçimiyle:

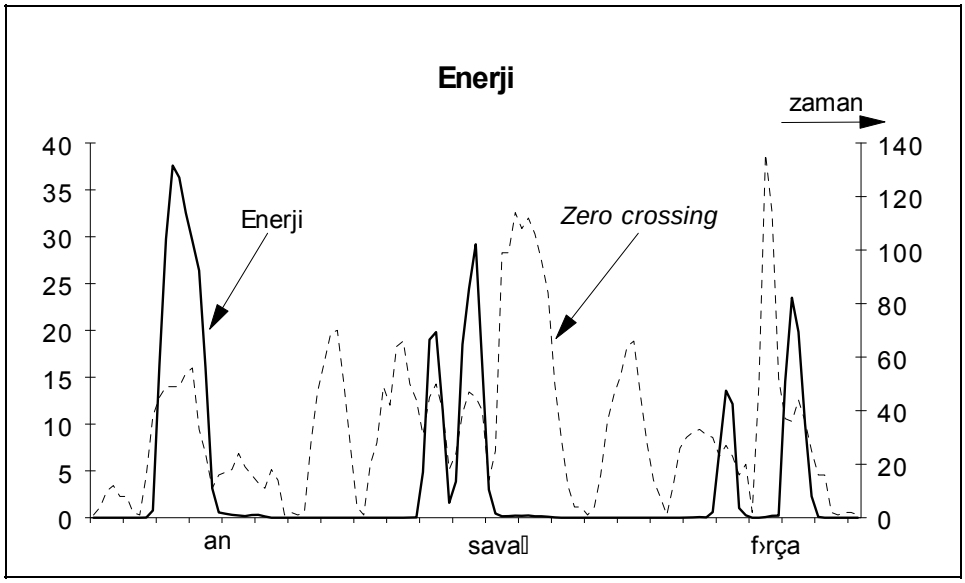
$$E(k) \ = \sum_{i=-\infty}^{\infty} \left[x(i) w(k-i) \right]^2 \tag{7.5}$$

olarak ya da

$$\begin{aligned} E(k) &= \sum_{i=-\infty}^{\infty} x^2(i) \cdot h(k-i) \\ h(k) &= w^2(k) \end{aligned} \tag{7.6}$$

formülleri ile hesaplanmaktadır. *w(k-i)* enerjinin hesaplandığı pencereyi temsil etmektedir. Enerjiyi, yukarıda verilen formülle hesap etmek yerine, *short time avarage magnitude* olarak bilinen bir diğer parametre ile temsil etmek de olanaklıdır. *Short time avarage magnitude* aşağıdaki gibi hesaplanmıştır:

$$M(k) = \sum_{i=-\infty}^{\infty} \left| x(i) \right| w(k-i) \tag{7.7}$$



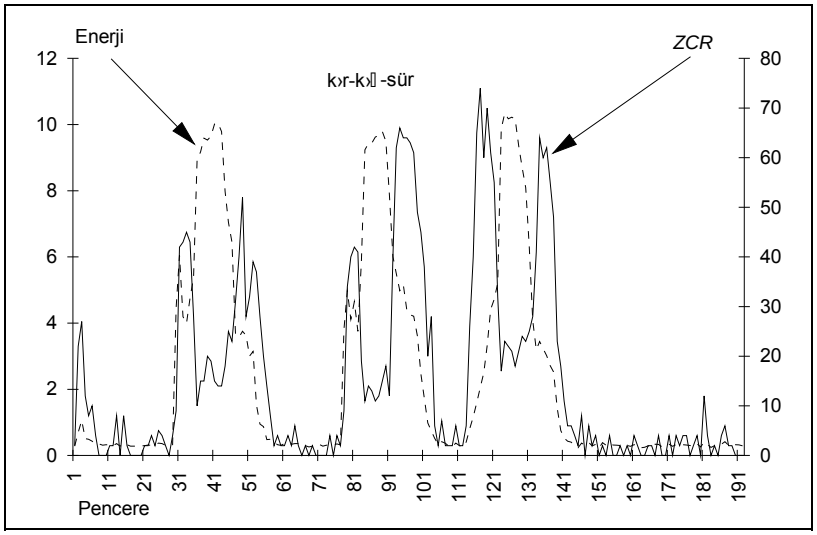
Çizim 7.10. Bir sözcük sinyaline ilişkin örnek enerji *(average magnitude)* ve *zero crossing rate* değerleri

Türkçe sesli ifadenin sınırlarının belirlenmesi

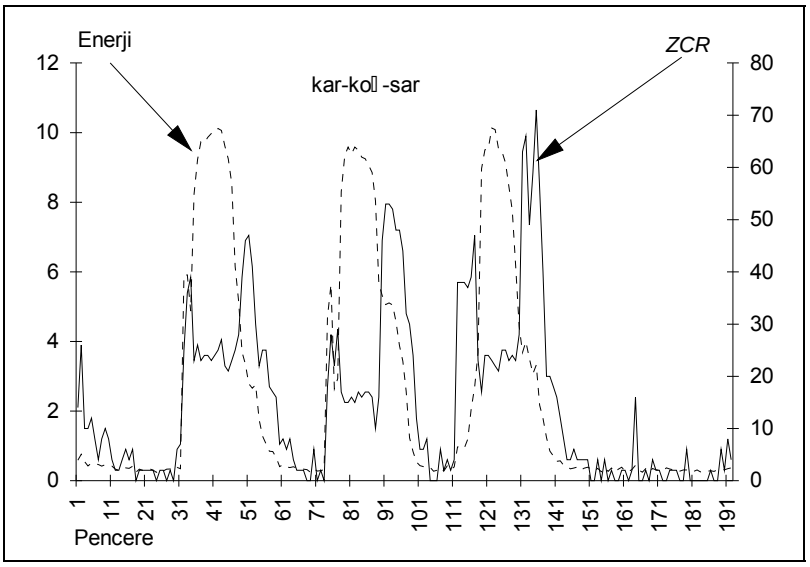
ZÜZ türünde hecelerden oluşan sözcüklerdeki ünlülerin sınır belirleme işlemleri için SİTLab'da geliştirilen Sayısal Sonograf yazılımından yararlanılmıştır. Daha önceki kesimde sözü edilen yalın çiftler kümesi üzerinde yapılan başlangıç ve bitiş noktalarının belirlenmesi işlemleri, Çizim 7.11 ve 7.12'de verilen enerji ve *ZCR* değerlerine ilişkin grafiklerle özetlenmiştir. Yalın çiftler ile sınır belirleme çalışması, Çizim 7.11'deki üç sözcüğün, Çizim 7.12'deki diğer üç sözcük ile karşılaştırması yoluyla örneklenmiştir. Her sözcük için, baştaki ve sondaki ünsüz fonemler ortada kalan ünlünün sınırlarının belirlenmesinde yardımcı olmaktadır. Grafiklerden de görüleceği üzere, sözcüklerin (tek heceli sözcükler) ünlü/ünsüz seslerinin sınırlarının belirlenmesinde enerji ve *ZCR* değerlerinin kullanımı elle yapılan belirleme için yeterli olabilmektedir. Bu belirleme şu ilkelere dayalı olarak yapılmıştır:

- Ünlülerin seslendirildiği kesimlerde, bulunan enerji değeri yüksek ancak *ZCR* değeri (pencerenin genel sıklık değeri) düşük olmaktadır.
- Ünsüzlerin bulunduğu kesimlerde, enerji değerleri, ünlülere göre düşük ancak *ZCR* değerleri yüksek olmaktadır.
- Sesli ifadeler dışında kalan kesimin enerji değeri sesli ifadenin sahip olduğundan çok daha düşük olmak durumundadır. Aksi halde tanıma işlemi işin başında zorlaşacak ya da olanaksızlaşacaktır.

Yalın çiftler kullanılarak ünlülerin ayrılması işlemi, sözcük sınırlarının bulunmasından sonra enerji ve *ZCR* değerlerinin üst üste çakıştırılarak mantıksal VE işleminin uygulanması biçiminde de düşünülebilir.



Çizim 7.11. ZÜZ türünde hecelerden oluşan yalın çift örnekleri. (Diğer çiftler Çizim 7.12'de verilmiştir)



Çizim 7.12. ZÜZ türünde hecelerden oluşan yalın çift örnekleri.

***Cepstrum* katsayılarının hesaplanması**

Cepstrum değerlerini hesaplamak için önce spektral değerlerin logaritması alınır. Daha sonra *Inverse Fourier* dönüşümü *log-spektrum* değerlerine uygulanır(Oppenheim 1989). Bu işlemler:

$$c(n) = \frac{1}{N}\sum_{k=0}^{N-1} \log_{10} \left| S_{avg}(k) \right| e^{(2\pi/N)kn} \qquad 0 \le n \le N-1 \tag{7.8}$$

eşitlikleri ile gerçekleştirilir. Buradaki *c(n), cepstrum* değerini temsil eder. Eğer bu işlem *Fourier* dönüşümü ya da filtre dizisinden sonra yapılırsa, sözkonusu *cepstrum* değerleri *Fourier* dönüşümünden elde edilen *cepstral* katsayıları olarak adlandırılır. *c(0)* değeri sinyalin, spektrum ortalamasını ya da kare kök ortalamasını *(rms)* ifade eder. 7.8'de verilen eşitlik, *log-spectrum*'un *inverse DFT* değeri için tanımlandığında yalınlaşacaktır.

$$c(n) = \frac{2}{N}\sum_{k=0}^{N} S_{avg}(I(k)) \cos\left(\frac{2\pi}{N} kn\right) \tag{7.9}$$

n'nin küçük değerleri için hesaplanan *cepstrum* değerleri, sesli ifade sinyalinin kısa süreli *(short term)* korelasyon bilgisini taşımaktadır.

FFT üzerinden *cepstrum* elde etmede 7.9 eşitliğinin *mel* aralıklı düzenlenmiş filtre çıkışları kullanılmaktadır (Davis 1980). MFCC *(Mel-frequency cepstrum coefficients)* olarak adlandırılan *cepstrum* değerleri:

$$MFCC_i = \sum_{k=1}^{20} X_k \cos\left[i \left(k - \frac{1}{2} \right) \frac{\pi}{20} \right] \quad i = 1, 2, \ldots, M \tag{7.10}$$

ile bulunur. *M cepstrum* sayısı, X_k {*k*=1,2,...,20} *k*'ıncı sıklık bölgesi *cepstrum* değeridir. *DFT* çıkışları kullanıldığında elde edilen katsayılar doğrusal sıklık *cepstrum* katsayılarıdır *(Linear-frequency cepstrum coefficients)*:

$$LFCC_i = \sum_{k=1}^{K-1} Y_k \cos\left[\frac{i\pi}{K} \right] \quad i = 1, 2, \ldots, M \tag{7.11}$$

Burada *K*, logaritması alınmış *DFT*'nin çıkış (Y_k) sayısıdır.

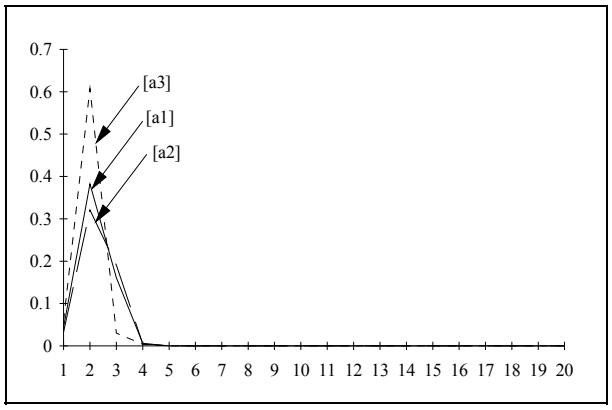
Özellik vektörlerinin hazırlanması

Parametrik dönüşüm işlemleri ile, sesli ifade tanımada kullanılacak özellik vektörleri oluşturulur. Özellik vektörleri oluşturmada yaygın biçimde kullanılan birden fazla teknik bulunmaktadır. Bunlardan en yaygın olarak kullanılanları şunlardır:

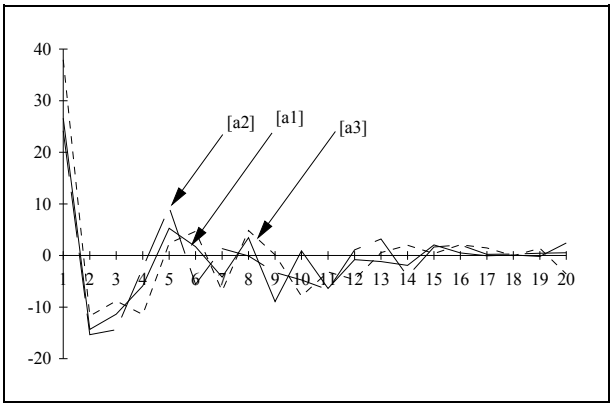
- doğrusal *FFT*'ye dayalı *cepstrum*,
- *mel* skalasında *FFT*'ye dayalı *cepstrum* (*Bark* ve *Erg* skalaları *mel*'e karşı kullanılan seçenekleri oluşturmaktadır),
- *Linear Predictive Coefficients*,
- *LPC*'ye dayalı *cepstrum*,
- *LPC-autocorrelation*,
- *LPC-reflection coefficients* ve
- *LPC Smoothed Group Delay Spectrum*

katsayıları hesaplama teknikleridir.

Türkçe sesli ifade tanıma korpüsünden alınan /a/ fonemine ilişkin üç örnek için Doğrusal *FFT*'ye dayalı olarak elde edilen 20 eşit aralıklı sıklık bandı katsayı değerlerin grafiği Çizim 7.13'de verilmiştir. Mel skalasındaki *FFT*'den elde edilen *cepstrum* katsayı değerlerinin grafiği ise Çizim 7.14'de verilmiştir.

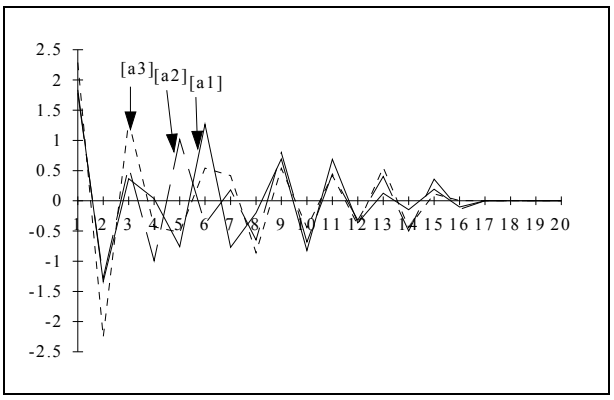


Çizim 7.13. [a1], [a2], [a3] *phon*'ları için
FFT 'ye dayalı sıklık bandı katsayı değerleri

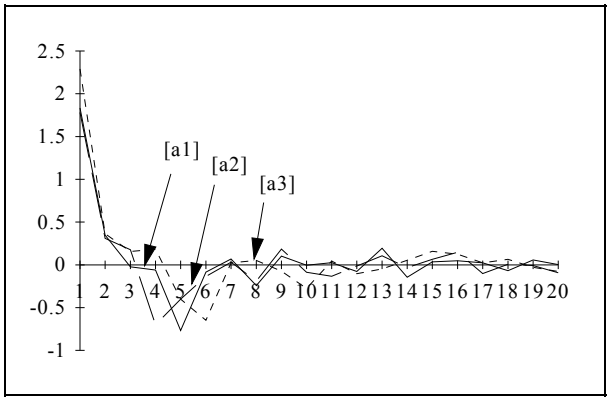


Çizim 7.14. [a1], [a2], [a3] *phon*'ları için *Mel* Skalasında *FFT* 'ye dayalı *Cepstrum* katsayı değerleri

Çizim 7.15'de, aynı *phon*'lar için *LPC* katsayı değerlerinin çizimi verilmiştir. Çizim 7.16'da *LPC*'ye dayalı olarak hesaplanan *cepstrum* değerleri örneklenmiştir.

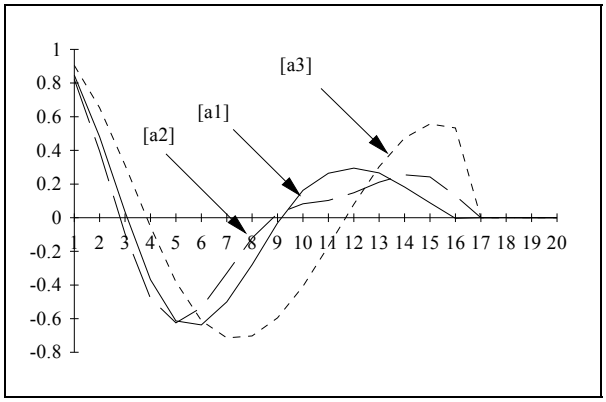


Çizim 7.15. [a1], [a2], [a3] *phon*'ları için oluşturulmuş *LPC* değerleri

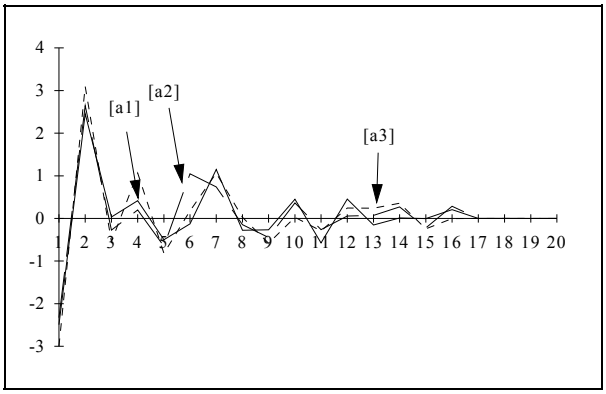


Çizim 7.16. [a1], [a2], [a3] *phon*'ları için oluşturulmuş *LPC Cepstrum* değerleri

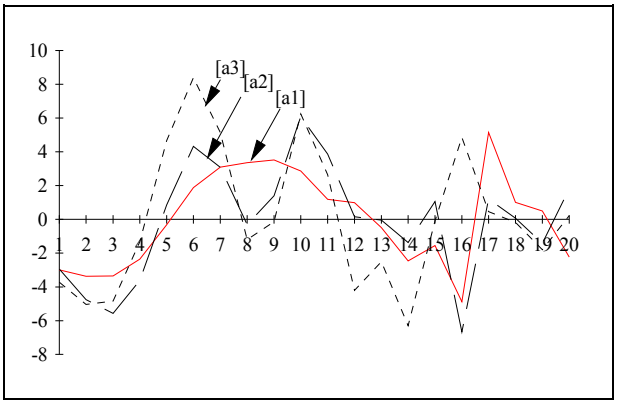
Çizim 7.17'de *LPC-autocorrelation* değerleri çizdirilmiştir. Bu değerler, sesli ifade sinyaline uygulanan *autocorrelation* fonksiyonundan elde edilmiştir. Çizim 7.18'de ise *LPC-reflection coefficents* değerlerinin logaritması verilmiştir.



Çizim 7.17. [a1], [a2], [a3] *phon*'ları için oluşturulmuş *autocorrelation* değerleri



Çizim 7.18. [a1], [a2], [a3] *phon*'ları için hesaplanmış *reflection coefficient* değerlerinin logaritması.



Çizim 7.19. [a1], [a2], [a3] *phon*'ları için hesaplanmış *LPC-SGDS* katsayı değerleri

Çizim 7.19'da *LPC*'ye dayalı olarak hesaplanan katsayılara *Herald SINGER*'in geliştirdiği algoritma uygulanmıştır. Bu yöntem, daha sonraki karşılaştırmalarda görüleceği üzere oldukça başarılı özellik vektörlerinin elde edilmesine olanak sağlamaktadır. Yöntem *Low Bit Rate Quantization of the Smoothed Group Delay Spectrum* olarak anılır (*Singer* 1990). Algoritmada herbir vektör birleşeni 7.12'deki formül kullanılarak bulunmaktadır.

$$sgds[i] = \frac{\sum_{k=0}^{nceps-1} \sin(k \cdot band\_width[i]) \cdot lpcceps[k] \cdot \cos(k \cdot center\_freq[i])}{band\_width[i]} \qquad (7.12)$$

burada *band_width* ve *center_freq* 7.13'e göre bulunur:

$$center\_freq[i] = \frac{2.0 \cdot \pi \cdot Mel\_scaled\_freq[i + 0.5]}{sample\_rate}$$
$$band\_width[i] = \frac{\pi (Mel\_scaled\_freq[i] - Mel\_scaled\_freq[i+1])}{sample\_rate} \qquad (7.13)$$

7.5. Türkçe Sesli İfade Tanıma için Özellik Vektörlerinin Seçilmesi

Buraya kadar sözü edilen çeşitli özellik vektörü hesaplama yöntemleri Türkçe sesli ifade korpüsüne uygulanmış, elde edilen vektörlerin, fonemleri temsil etme başarımları *Learning Vector Quantization (LVQ)* algoritması kullanılarak ölçülmüştür. Ölçüm sonuçları Çizelge 7.5 ve 7.6'da her fonem için ayrı ayrı verilmiştir. Tanımada da doğrudan kullanılabilecek *Learning Vector Quantization*

algoritması, özellik vektörlerinin başarımını incelemede kullanılmıştır. İzleyen kesimde bu algoritmaya ilişkin ayrıntılı bilgi ve bunun yanısıra sözkonusu çizelgelerdeki başarı oranlarının yorumları verilecektir.

7.5.1. Özellik vektörü Hesaplama Yöntemlerinin Türkçe Sesli İfade Tanıma için Karşılaştırılması

LVQ algoritması *Kohonen* tarafından geliştirilen gözetimli öğrenme yöntemini kullanan bir algoritmadır (Kohonen 1990). Bu algoritma ile sesli ifade özellik vektörlerinden *codebook* adı ile anılan vektörler çizelgesi oluşturulmaktadır. *Codebook*, sesli ifade özellik vektörlerinden birbirlerine benzeyenlerin kümelendiği bir çizelgedir. Bu tabloda Türkçe her fonem için bu fonemi temsil eden özellik vektörü, yine fonemi temsil eden harfle birlikte tutulmaktadır.

Sesli ifadenin özellik vektörleri kullanılarak tanınması iki adımda gerçekleşmektedir. İlk adım sesli ifadeleri tanımaya yarayan yapının kurulması ya da öğrenme aşamasıdır. Bu aşama genelde gözetimli öğrenme tekniğini kullanır. Sesli ifadelerin özellik vektörleri arasındaki sınıflandırma işlemi bu aşamada gerçekleştirilir. Sınıflandırma işleminde çeşitli algoritmalar kullanılabilmektedir. *K-Nearest Neigbour* algoritması bunlardan en yaygın kullanılan kümeleme algoritmasıdır. Sınırları bir biçimde belirlenmiş tüm fonemlerin çeşitli ses örneklerine ilişkin özellik vektörleri sınıflandırılarak ileride karşılaştırma amacıyla kullanılacak ölçüt fonem sınıfları öğrenme aşamasında oluşturulur.

Kesimlenen (sınırları belirlenen) fonemlerin, yukarıda anılan değişik yöntemlerle oluşturulan özellik vektörleri *SPSS* istatistik yazılım paketi ve *LVQ* algoritmasına dayalı olarak sınıflandırmaları yapılmıştır. Bu yolla hangi özellik vektörü oluşturma yönteminin, kümeleme açısından daha başarılı sonuç verdiği araştırılmıştır (Artuner 1994). İleride de açıklanacağı üzere, bu çalışmanın sonucunda, *mel* skalasında *FFT*'ye dayalı *cepstrum* değerlerinden oluşan özellik vektörlerinin en uygun özellik vektörü olduğu görülmüştür. Bu çalışmaya ilişkin sonuçlar Çizelge 7.5 ve 7.6'da verilmiştir. *Mel* skalasında *FFT*'ye dayalı *cepstrum* değerlerinden oluşan özellik vektörleri ile yapılan kümeleme sonunda tanıma amacıyla kullanılacak *codebook* elde edilmiştir.

İzleyen kesimde *LVQ* algoritmasına kısaca değinilecek daha sonra yapılan özellik vektörleri başarım ölçüm çalışmasının sonuçları aktarılacaktır.

***Learning Vector Quantization LVQ* algoritması**

Learning vector quantization yöntemi Kohonen ve grubunca istatistiksel sınıflama ve örüntü tanıma amacıyla ortaya atılmıştır. Bu yöntemle, *codebook* olarak anılan *ölçüt* (referans) sınıf vektörlerinin oluşturulması amaçlanmaktadır. *Learning vector quantization* amacıyla üç farklı algoritma geliştirilmiştir. Bunlar *LVQ1, LVQ2.1* ve *LVQ3* olarak bilinmektedirler (Kohonen 1990).

LVQ1 algoritmasında m_i ile temsil edilen *codebook* vektörü ile giriş vektörleri x arasındaki uzaklık minimize edilmeye çalışılır. Bunun için bir foneme ilişkin olduğu varsayılan bir özellik vektörünün, hangi *codebook* vektörüne (m_i) en yakın olduğu 7.14'e göre belirlenir. Bu vektör (m_c), küme merkezi olarak tanımlanır. Elde edilen bu merkezin, incelenen özellik vektörünün ait olduğu varsayılan foneme ilişkin olup olmamasına göre, 7.15'deki kurallardan biri uygulanarak değişik günlemeler yapılır.

$$c = \arg\min_i \|x - m_i\| \tag{7.14}$$

Eğer x ve m_c aynı fonemle ilgili ise, $m_c(t+1) = m_c(t) + \alpha(t)\left[x(t) - m_c(t)\right]$

Eğer x ve m_c farklı fonemlerle ilgili ise, $m_c(t+1) = m_c(t) - \alpha(t)\left[x(t) - m_c(t)\right]$ (7.15)

$m_c(t+1) = m_c(t); \quad i \neq c$ için.

Burada, α değeri değişmez olup [0,0-1,0] aralığındadır.

Bu algoritmanın iyileştirilmiş biçimi *OLVQ1* (*Optimized Learning Vector Quantizing*) olarak sunulmuştur. x'in aynı ya da farklı fonemlere ilişkin olup olmaması 7.16'da verilen tek bir formülle ele alınmaktadır.

$$m_c(t+1) = \left[1 - s(t)\alpha_c(t)\right]m_c(t) + s(t)\alpha_c(t)x(t) \tag{7.16}$$

Bu formülde, aynı fonem için $s(t)$=+1, değilse $s(t)$=-1 alınır. a değeri de değişmez olmak yerine 7.17 ve 7.18'de verilen fomüllerle hesaplanmaktadır:

$$\alpha(t) = \left[1 - s(t)\alpha_c(t)\right]\alpha_c(t-1) \tag{7.17}$$

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \tag{7.18}$$

LVQ2.1'de, *LVQ1*'den farklı olarak, bulunan merkez *(m)* ile incelenen özellik vektörünün *(x)* aynı ya da farklı fonemlere ilişkin olmasına göre, m_i ve m_j ile temsil edilen iki ayrı *codebook* tutulmaktadır. x vektörünün m_i ve m_j ye Euclid uzaklığı d_i ve d_j ise fonem merkezinin belirlenmesi 7.19 ve 7.20'ye göre yapılır:

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s; \quad s = \frac{1-w}{1+w}. \tag{7.19}$$

Burada w'nin değeri, 0.2 ile 0.3 arasında seçilir.

$$\begin{aligned} m_i(t+1) &= m_i(t) - \alpha(t)\left[x(t) - m_i(t)\right], \\ m_j(t+1) &= m_j(t) + \alpha(t)\left[x(t) - m_j(t)\right], \end{aligned} \tag{7.20}$$

Burada m_i x'in ait olduğu foneme, m_j ise farklı foneme ilişkin *codebook* vektörleridir.

LVQ3 algoritmasında ise

$$\begin{aligned} m_i(t+1) &= m_i(t) - \alpha(t)\left[x(t) - m_i(t)\right], \\ m_j(t+1) &= m_j(t) + \alpha(t)\left[x(t) - m_j(t)\right], \end{aligned} \tag{7.21}$$

ifadeleri, 7.22'deki tek bir ifade ile ele alınır:

$$m_k(t+1) = m_k(t) + \varepsilon\alpha(t)\left[x(t) - m_k(t)\right], \tag{7.22}$$

Burada, $k \in$ olup ε değeri 0.1 ile 0.5 arasında bir değerdir.

Bu tez çalışması kapsamında, yukarıda açıklanan algoritmalardan *OLVQ* algoritması kullanılmıştır. Bu yolla, Türkçe tüm ünlü ve ünsüz fonemler için *codebook* oluşturulmuştur. Bu yapılırken özellik vektörü olarak kullanılan vektörler, değişik özellik vektörü oluşturma yöntemlerine dayalı olarak elde edilmiştir. Bu bağlamda her özellik vektörü oluşturma yöntemi için ayrı bir *codebook* oluşmuştur. Bu *codebook*'lar oluşturulurken her fonemle ilgili çok sayıda özellik vektörünün *codebook* içinde bu fonemle ilgili ölçüt (referans) vektörle uyuşup uyuşmaması

sınanmış ve bu sınamalara göre özellik vektörü hesaplama yöntemlerinin başarım çizelgeleri çıkarılmıştır. Bu bağlamda iki deney yapılmıştır. İlk deneyde *codebook*'un 28 fonemlik, ikinci deneyde ise, kimi fonem altı ses birimleri ayrıştırılarak 65 ögelik *codebook*'lar söz konusu edilmiştir. Her iki deney kapsamında elde edilen başarım çizelgeleri Çizelge 7.5 ve 7.6'da verilmiştir. Her iki deney kapsamında da 2183 ünlü ve ünsüz etiketli *phon* takımı kullanılmıştır. Daha önce de belirtildiği üzere, *mel* skalasında *FFT*'ye dayalı *cepstrum* değerlerinden oluşan özellik vektörlerinin en uygun özellik vektörü olduğu sonucuna varılmıştır.

Tez çalışmasında, *Fourier* dönüşümünden elde edilen 64, 128 ya da 256 adet sıklık değerlerinin logaritması alınmakta ve bu değerlerden, 7.9 eşitliği kullanılarak *mel* aralığında 20 adet *cepstrum* değeri elde edilmiştir.

Yukarıda da belirtildiği üzere iki deney yapılmış ve bu deneylerden birinde 28 ögelik ikincisinde ise 65 ögelik *codebook*'lar sözkonusu edilmiştir. 28 ögelik *codebook*'ta Türkçe alfabeye de taban oluşturan 28 fonem bulunmaktadır. Fonemlerin hecelerde başta, ortada ya da sonda olmasına göre ayrıştırılmasının, kümeleme başarımını önemli ölçüde artırdığı görüldüğünden, ikinci bir deney kapsamında *codebook* öge sayısı, kimi fonemler hecelerdeki konumlarına göre fonem altı ses birimlerine ayrıştırılarak 65 ögelik ikinci *codebook* elde edilmiştir. İkinci *codebook* kapsamında, örneğin /a/ fonemi yerine [a1], [a2], [a3] olarak üç değişik fonem altı ses birimi düşünülmüştür. Bunlardan [a1] /a/'nın hece başındaki, [a2] hece ortasındaki, [a3] ise hece sonundaki biçimini temsil etmektedir.

Çizelge 7.5. Türkçe fonem altı ses birimler için çeşitli özellik vektörü çıkarma tekniklerinin kümeleme başarımları yönünden karşılaştırılması.

Fonem		fas	fcs	lacs	lars	lcs	lfs	lsgs
/a/	434	11.06%	78.57%	73.96%	79.95%	74.88%	89.63%	71.20%
/e/	251	21.51%	78.09%	67.34%	72.91%	80.48%	29.88%	78.09%
/i/	118	7.93%	79.66%	50.00%	67.80%	77.12%	16.95%	66.95%
/ı/	108	27.78%	69.44%	17.59%	68.52%	64.81%	6.48%	71.30%
/o/	93	16.13%	80.65%	46.24%	76.34%	88.17%	23.66%	78.49%
/ö/	48	0.00%	93.75%	37.42%	77.08%	79.17%	8.33%	83.33%
/u/	61	24.59%	86.89%	16.39%	72.13%	88.52%	6.56%	67.57%
/ü/	69	11.59%	97.10%	57.07%	97.65%	87.51%	10.14%	89.86%
Toplam	*1182*	*15.17%*	*80.03%*	*57.39%*	*76.43%*	*78.03%*	*44.67%*	*74.22%*
/b/	56	0.00%	69.64%	7.14%	53.57%	50.00%	32.14%	51.79%
/c/	25	0.00%	76.00%	52.00%	80.00%	68.00%	8.00%	52.00%
/ç/	26	0.00%	92.31%	61.54%	96.15%	80.77%	50.00%	76.92%
/d/	64	0.00%	73.44%	9.38%	77.00%	68.75%	59.38%	51.56%
/f/	42	7.14%	73.81%	42.86%	66.67%	73.81%	37.71%	54.76%
/g/	21	0.00%	57.14%	9.52%	66.67%	76.19%	0.00%	19.05%
/h/	42	2.38%	76.19%	52.38%	66.67%	73.81%	23.81%	76.19%
/j/	51	0.00%	60.78%	43.14%	80.39%	76.47%	60.78%	66.67%
/k/	68	16.18%	57.88%	52.94%	58.82%	64.71%	22.06%	47.59%
/l/	30	0.00%	70.00%	0.00%	60.00%	50.00%	0.00%	53.33%
/m/	58	62.07%	87.93%	24.14%	68.97%	82.76%	53.45%	58.62%
/n/	59	6.78%	88.14%	67.80%	79.66%	76.27%	16.95%	79.66%
/p/	51	0.00%	74.51%	27.45%	60.78%	39.22%	7.88%	39.22%
/r/	118	37.59%	68.64%	44.92%	68.64%	59.32%	63.56%	61.86%
/s/	54	0.00%	92.59%	77.93%	87.04%	90.74%	96.30%	79.63%
/ş/	53	32.08%	92.45%	88.68%	86.79%	88.68%	84.91%	90.57%
/t/	38	0.00%	76.32%	57.26%	71.05%	84.21%	7.26%	50.00%
/v/	37	0.00%	67.57%	13.51%	43.24%	77.68%	21.62%	51.35%
/y/	56	0.00%	66.07%	51.79%	58.93%	80.36%	37.71%	57.14%
/z/	52	27.00%	61.54%	53.85%	59.62%	59.62%	63.46%	63.46%
Toplam	1001	13.03%	73.86%	43.24%	69.16%	70.10%	42.43%	60.38%

Toplam başarı	2183	13.93%	77.14%	50.48%	72.97%	74.30%	43.47%	67.75%

Çizelge 7.6. Türkçe fonem altı ses birimler için çeşitli özellik vektörü çıkarma tekniklerinin kümeleme başarımları yönünden karşılaştırılması. (1. Kesim-Ünlüler)

phon	Sayı	fah	fch	lach	larh	lch	lfh	lgh
a1	92	0.00%	84.78%	46.74%	77.17%	77.17%	77.00%	79.35%
a2	154	0.00%	72.73%	33.77%	87.66%	77.27%	82.47%	74.03%
a3	188	0.00%	87.23%	28.72%	89.36%	81.91%	89.36%	68.62%
e1	37	0.00%	100.00%	0.00%	91.89%	97.30%	70.27%	97.30%
e2	113	0.00%	87.61%	43.36%	88.50%	79.65%	90.27%	91.15%
e3	101	0.00%	87.15%	14.85%	94.06%	74.26%	87.15%	87.15%
i1	13	0.00%	76.92%	0.00%	76.92%	76.92%	76.92%	76.92%
i2	53	0.00%	86.79%	0.00%	92.45%	88.68%	67.92%	86.79%
i3	52	0.00%	92.31%	0.00%	92.31%	96.15%	73.08%	92.31%
ii1	7	0.00%	87.71%	0.00%	71.43%	71.43%	57.14%	100.00%
ii2	46	0.00%	91.30%	0.00%	84.78%	89.13%	54.35%	84.78%
ii3	55	0.00%	100.00%	3.64%	96.36%	100.00%	69.09%	90.91%
o1	27	0.00%	100.00%	11.11%	100.00%	100.00%	92.59%	92.59%
o2	13	0.00%	100.00%	0.00%	100.00%	100.00%	53.85%	92.31%
o3	53	0.00%	96.23%	16.98%	96.23%	90.57%	77.47%	92.45%
ö1	8	0.00%	100.00%	77.00%	100.00%	100.00%	0.00%	100.00%
ö2	40	100.00%	100.00%	2.50%	100.00%	97.50%	77.50%	92.50%
u1	6	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	83.33%
u2	38	0.00%	100.00%	100.00%	94.74%	100.00%	81.58%	89.47%
u3	17	0.00%	100.00%	0.00%	94.12%	100.00%	58.82%	94.12%
ü1	10	0.00%	100.00%	0.00%	90.00%	80.00%	80.00%	80.00%
ü2	34	0.00%	100.00%	0.00%	100.00%	100.00%	82.35%	100.00%
ü3	25	0.00%	96.00%	0.00%	96.00%	96.00%	56.00%	88.00%
Toplam	1182	3.38%	89.10%	23.03%	90.61%	85.87%	79.01%	84.01%

Çizelge 7.6. Türkçe fonemler için çeşitli özellik vektörü çıkarma tekniklerinin kümeleme başarımları yönünden karşılaştırılması. (2. Kesim- Ünsüzler)

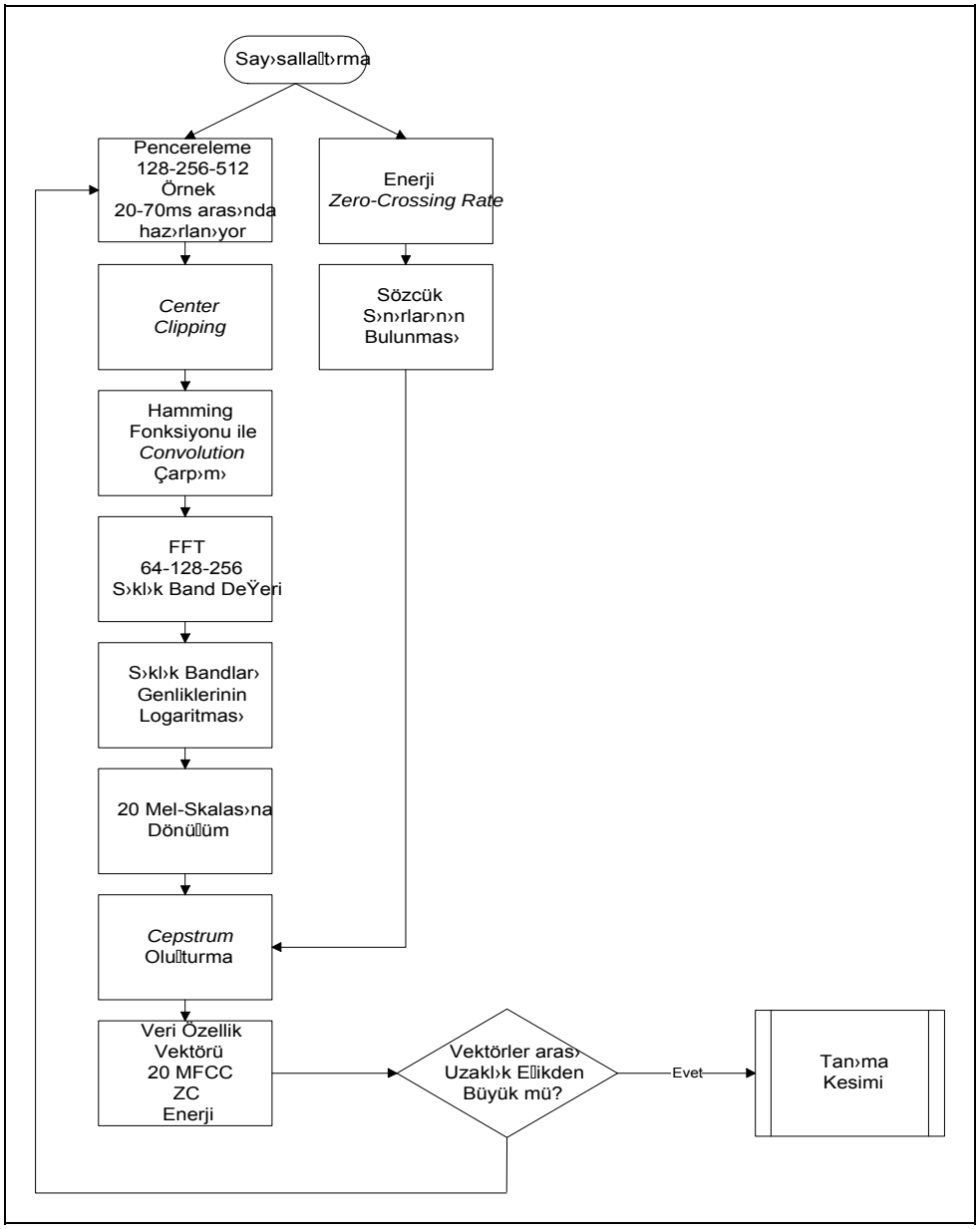
phon	*Sayı*	*fa*	*fc*	*lac*	*lar*	*lc*	*lf*	*lg*
b1	45	0.00%	82.22%	0.00%	80.00%	80.00%	73.33%	82.22%
b3	11	0.00%	72.73%	0.00%	54.55%	63.64%	36.36%	47.45%
c1	25	0.00%	88.00%	24.00%	80.00%	88.00%	52.00%	72.00%
ç1	8	0.00%	87.50%	27.00%	77.00%	77.00%	50.00%	87.50%
ç3	18	0.00%	100.00%	16.67%	100.00%	88.89%	94.44%	72.22%
d1	64	0.00%	92.19%	0.00%	84.38%	87.94%	89.06%	89.06%
f1	25	0.00%	96.00%	72.00%	92.00%	96.00%	88.00%	64.00%
f3	17	0.00%	94.12%	37.29%	94.12%	88.24%	76.47%	94.12%
g1	15	0.00%	46.67%	0.00%	73.33%	66.67%	33.33%	26.67%
g3	6	0.00%	100.00%	66.67%	100.00%	100.00%	100.00%	0.00%
h1	20	0.00%	90.00%	17.00%	80.00%	97.00%	80.00%	97.00%
h3	22	0.00%	86.36%	18.18%	90.91%	81.82%	54.55%	72.73%
j1	22	0.00%	81.82%	36.36%	90.91%	100.00%	63.64%	86.36%
j3	29	0.00%	100.00%	79.31%	100.00%	100.00%	96.55%	93.10%
k1	43	0.00%	69.77%	58.14%	83.72%	69.77%	62.79%	86.05%
k2	3	0.00%	66.67%	0.00%	0.00%	100.00%	0.00%	100.00%
k3	22	0.00%	90.91%	0.00%	77.27%	54.55%	59.09%	63.64%
l1	17	0.00%	82.35%	0.00%	76.47%	82.35%	29.41%	70.59%
l2	2	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	0.00%
l3	11	0.00%	72.73%	0.00%	18.18%	0.00%	0.00%	47.45%
m1	37	0.00%	100.00%	0.00%	94.59%	97.30%	72.97%	94.59%
m3	21	0.00%	80.95%	0.00%	76.19%	80.95%	61.90%	42.86%
n1	9	0.00%	88.89%	0.00%	88.89%	88.89%	66.67%	77.78%
n2	5	0.00%	100.00%	0.00%	100.00%	100.00%	0.00%	100.00%
n3	45	0.00%	97.56%	0.00%	84.44%	77.78%	60.00%	86.67%
p1	22	0.00%	47.45%	0.00%	31.82%	50.00%	13.64%	27.27%
p3	29	0.00%	68.97%	0.00%	77.86%	89.66%	62.07%	68.97%
r1	28	0.00%	89.29%	0.00%	82.14%	82.14%	50.00%	78.57%
r2	6	0.00%	33.33%	0.00%	33.33%	50.00%	33.33%	0.00%
r3	84	0.00%	84.52%	61.90%	91.67%	80.95%	80.95%	87.71%
s1	44	0.00%	97.73%	68.18%	97.73%	97.73%	97.73%	90.91%
s3	10	0.00%	90.00%	20.00%	80.00%	90.00%	50.00%	90.00%
ş1	36	0.00%	100.00%	88.89%	97.22%	97.22%	97.22%	100.00%
ş3	17	0.00%	88.24%	23.53%	82.35%	88.24%	76.47%	82.35%
t1	15	0.00%	66.67%	0.00%	73.33%	93.33%	46.67%	40.00%

t3	23	0.00%	82.61%	13.04%	52.17%	69.57%	52.17%	47.83%
v1	16	0.00%	87.50%	0.00%	37.50%	81.25%	18.75%	62.50%
v3	21	0.00%	76.19%	0.00%	47.62%	90.48%	57.14%	61.90%
y1	33	0.00%	84.85%	6.06%	77.76%	78.79%	54.55%	78.79%
y3	23	0.00%	100.00%	4.35%	86.96%	100.00%	86.96%	86.96%
z1	15	0.00%	73.33%	0.00%	53.33%	60.00%	33.33%	60.00%
z3	37	0.00%	89.19%	86.49%	83.78%	91.89%	86.49%	89.19%
Toplam	1001	0.00%	85.75%	26.06%	80.76%	83.30%	67.13%	76.87%
Genel	2183	1.83%	87.40%	24.37%	86.03%	84.61%	73.34%	80.53%

x1-hece başında; x2 -hece ortada; x3-Phone, hece sonunda

fa (FFT amplitude); fc (FFT Cepstrum); lac(LPC Autocorrelation); lar (LPC Area); lc (LPC Cepstrum); lf (LPC Frequancy); lg (LPC-SGDS)

Özellik vektörleri üzerinde yapılan bu çalışmalar gerçek zamanlı uygulamaya temel oluşturmada kullanılmaktadır. Bu nedenle, *Gerçek Zamanlı Sesli İfade Çözümleme* yazılımı elde edilen bu veriler ışığında TMS320C30 üzerinde çalışacak biçimde Çizim 7.23'deki önerilen model çerçevesinde oluşturulmaya çalışılmıştır. Bu yazılım aracılığıyla sesli ifadeler, gerçek zamanda özellik vektörlerine dönüşmekte ve tanıma kesimine aktarılmaktadır.



Çizim 7.20. Gerçek zamalı sesli ifade spektral analizi ve özellik çıkarım modeli

7.7. Türkçe Sesli İfade Tanıma

Sesli ifade tanıma dendiğinde özellik vektörleri ile kodlanmış sesli ifade birimlerinin tanınması anlaşılır. Daha önce de belirtildiği üzere, sesli ifadelerin birimlenmesi, genelde ya sözcük ya da fonem tabanında yapılmaktadır. Türkçe gibi sonek alan dillerde bu birimlemeyi sözcük tabanında yapma olanağı bulunmamaktadır. Bu tez

kapsamında birimleme fonem ve kimi fonem altı ses birimleri düzeyinde yapılmıştır. Bu nedenle sesli ifadeler özellik vektörleri ile kodlanırken önce bunlar içindeki fonem sınırları belirlenmekte, sonra bu sınırlar arasında özellik vektörü hesaplanarak fonemler kodlanmaktadır. Tanıma katmanı fonem özellik vektörlerini sınıflandıran ve bu yolla ilgili özellik vektörünün hangi foneme ilişkin olduğunu belirleyen bir katmandır. İzleyen kesimde sesli ifade tanıma kapsamında kullanılan teknikler açıklanmıştır:

7.7.1. Tanıma teknikleri

5'inci bölümde ayrıntısı ile verilen belli başlı sesli ifade tanıma teknikleri:

- *Dynamic Time Warping,*
- *Hidden Markov Model,*
- Nöron Ağı

teknikleri olarak sıralanabilir. Bunlardan Nöron ağı tekniğine ilişkin olanlar içinde ayrıca bir alt sınıflandırma yapmak da mümkündür. Bu sınıflandırma aşağıda verilmiştir:

- *Time Delay Neural Network,*
- *Multi Layer Neural Network,*
- *Self Organizing Neural Network*.

Tanıma çalışmalarda yukarıda sözü edilen yöntemlerden biri ya da birkaçı birlikte kullanılabilmektedir. Örnek olarak, *Dr. Toni Robinson* İngilizce sözcük tanıma kapsamında *Multi Layer Perceptron* ve *Hidden Markov Model* (*Hybrid Model*) yöntemlerini birlikte kullanmıştır (Robinson 1989-1993). Bu tez çalışmasında yukarıda sıralanan tüm modellere ilişkin yazılım kitaplığı oluşturulmuş ve örnekler üzerinde denemeler yapılmıştır. Bunlardan nöron ağlarına ilişkin olanlar *HP-720* İş İstasyonu üzerinde çalışan Nöron Ağ simulatörlerinde kurulup çalışma ilkelerine ilişkin bilgi birikimi oluşturulmuştur (*Multi Layer Perceptron*, *Self Organizing Feature Map* ve *Time Delay Neural Network*). Bu tür ağ benzetimlerinin yapılmasının, boyut ve hız açısından gerçek uygulamalara bir fikir vermesinden öteye bir işlevi yoktur. *Self Organizing Feature Map* algoritmasının tanıma sistemi için başlangıçta uygun olacağı düşünülmüş ve kullanılmıştır. Salt bu yöntemin kullanılması sistemin başarımında kimi olumsuzlukları beraberinde getirmektedir.

Tanımaya ilişkin kesime geçmeden önce izleyen kesimde kullanılan *Self Organizing Feature Map* algoritmasına ilişkin özet bilgi verilmiştir.

Self Organizing Feature Map (SOM), Kohonen'in gözetimsiz öğrenme (*unsupervised learning*) özelliğinin kullanıldığı bir algoritmadır. *Kohonen*, *SOM* yaklaşımını Fince ve Japonca üzerinde başarılı bir tanıma yöntemi olarak kullanmıştır. *Kohonen'*in çalışmasında hedef, sesle çalışan bir daktilo gerçekleştirmek olmuştur. Bu bağlamda kullanılan yöntem, sesli ifade tanımanın doğadaki biçimi ya da sesin duyulmasından anlamsal kavram oluşturmaya kadar geçen sürecin biyolojik bir modelidir. Beyindeki duyma merkezi, fonetik harita ya da *phonotopic map* olarak adlandırılan bir yapı ile modellenmiştir. Duyma süreci ise bu fonetik haritanın çalışma ilkesini oluşturmaktadır.

Kohonen'in yapay nöron ağı olarak kullandığı modelde, iki boyutlu, tek katmanlı 9x12 düğümden oluşan, altıgenler biçiminde dizilmiş düğümler kullanılmıştır. Düğümlerin altıgenler biçiminde dizilmesi her düğüm için komşulukların eşit uzaklıkta olmasını ve işlem kolaylığını sağlamaktadır. Çalışma ilkesi, özet olarak, *n* boyutlu giriş vektörünün iki boyutlu düzlemdeki izdüşümünün bulunması ya da *vector quantization* işlemi olarak açıklanabilir. Algoritmada, giriş vektörü ile iki boyutlu düzlemdeki *codebook* biçiminde tutulan vektörler arasındaki en küçük uzaklıklar bulunmaya çalışılır. Bu işlem:

$$\|x - m_c\| = \min_i \{\|x - m_i\|\} \text{ ya da}$$
$$c = \arg\min_i \{\|x - m_i\|\} \tag{7.23}$$

ifadesi ile gösterilebilinir.

Kohonen söz konusu bu algoritma ile tek katmanlı, iki boyutlu bu nöron ağında *phon* türleri arasında karar yüzeyleri oluşturmaktadır. Bunu gerçekleşirken her düğüm, giriş vektör bileşenleri ile arasında $m_i(t)$ ağırlık vektörüne sahiptir. Bu biçimde örüntü vektörlerinin birbirleriyle karşılaştırılabilecekleri ortam hazırlanmış olur. Öğrenme aşamasından önce tüm ağırlık vektörlerine rasgele sayılar atanır. Öğrenme işlemi, belirli aralıklarla gelen sinyal ile birlikte örüntü vektör kümeleri ile yürütülür. Ağırlık vektörleri herbir düğümü ayrı ayrı etkiler. Her örnek vektör $x(t)$ için aşağıdaki algoritma, ağırlık vektörlerini oluşturmada kullanılmaktadır:

- c düğümü, $x(t)$ giriş vektörü ile arasında en küçük *Euclidian* uzaklığına sahip olandır:

$$\|x(t) - m_c(t)\| = \min_i \|x(t) - m_i(t)\| \tag{7.24}$$

- Topolojide *c*'nin *Nc(t)* olarak anılan komşu düğümleri aşağıdaki algoritma ile günlenir:

$$\begin{aligned} &\forall i \in N_c(t).\ \ m_i(t+1) = m_i(t) + \alpha(t)\left[x(t) - m_i(t)\right] \\ &\forall i \notin N_c(t).\ \ m_i(t+1) = m_i(t) \end{aligned} \tag{7.25}$$

$N_c(t)$ ya da *c*'nin komşuları olan tüm düğümler başlangıçta tüm düğümleri kapsar. Zaman içinde komşu düğümlerin sayısı ve aranılan düğüme uzaklığı azaltılır. Bu azaltma *c*'nin kendisi elde edilinceye kadar sürdürülür. Bu durumda *c,* öğrenme aşamasında istenilen düğüm, test aşamasında ise aranılan sonuç düğüm olmaktadır. 7.24 ve 7.25 eşitliklerindeki *a(t)* doğrusal azalma fonksiyonudur.

Yöntem, her *phon* ya da fonem altı birimin (*quasi-phoneme*) benzerleri ile kümelenmesi ilkesini kullanır. Sözkonusu bu ses birimleri sesli ifadelerden çıkarılan özellik vektörleri ile temsil edilirler. Kümeleme aşamasında yakınlıkları, belirlenen bir değerin altında olanlar ilgili kümeye katılırlar. Bir kümeye katılan özellik vektörü o kümenin ağırlık merkezini etkiler. Dildeki tüm fonemlere ilişkin sesli ifade örnekleri kullanılarak, her fonem (kimi durumlarda fonemaltı ses birimleri) için bir küme merkezi oluşturulur. Bu küme merkezleri topluluğuna *codebook* adı verilir. Burada her kümenin bir *phon'*a karşı geldiği söylenebilir. Böylece ağ üzerinde her fonemin birden fazla *phon* tanımı bulunur. Bunlardan birbirlerine yakın olanlar aynı foneme ilişkin olurken, göreli ayrık olanlar için aynı fonemin ayrı *phon*'larından sözedilebilir.

7.7.2. Tanıma

Türkçe veri korpüsü üzerinde yapılan çalışmalar sonucunda, Türkçe sesli ifadelere ilişkin en uygun özellik vektörlerinin, *Hamming* pencereleme filtresinden elde edilen sinyaller üzerine uygulanan *mel* skalasında *FFT'*ye dayalı *cepstrum*'lar olduğu gözlenmiştir (Çizelge 7.5 ve 7.6). Korpüste saklanan sinyallerden hesaplanan özellik vektörlerinden sesli ifadelere ilişkin olanları elle etiketlenerek, salt sesli ifadelere ilişkin özellik vektörlerinden oluşan kütükler elde edilmiştir. Söz konusu kütükler hece ve fonem kütükleri olarak ayrı ayrı ele alınmıştır. Hecenin taban alındığı kütükte, fonemin hecenin içindeki yeri fonem simgesi ile etiketlenerek kullanılmıştır.

Tanımaya ilişkin önerilen model Çizim 7.21'de özetlenmiştir. Çizimden de görüleceği gibi, özellik vektörleri, hesaplandıkları mikrobilgisayar sisteminden, yerel ağ bağlantısı aracılığıyla, tanınmak üzere iş istasyonunun diskine yazılır. Özellik vektörü çıkarımı kesimi, hem süreklilik hem de hız isteyen bir kesim olması nedeniyle *TMS320C30* sinyal işleyicisi üzerinde gerçekleşir. Bu biçimi ile EVM-30 kartı sesli ifadelerin sürekli bir biçimde sayısallaştırılıp özellik vektörlerine dönüştürüldüğü bir donanım olarak düşünülmüştür. Bu kartı taşıyan mikrobilgisayar sistemi ise hesaplanan özellik vektörlerinden sesli ifade birimlerine ilişkin olanlarını iş istasyonuna aktaran bir sistemdir.

Çizim 7.21. Sesli ifade tanıma sistemi için bir Model

Sesli ifadelere ilişkin özellik vektörleri iki amaçla kullanılmaktadır. Bunlardan ilki nöron ağının alıştırılması diğeri ise tanımadır. Alıştırma, sistemin konuşmacıya uyum sağlanması, sistemin tanımaya hazırlanmasıdır. Alıştırma aşaması sesli ifadelerden elde edilen özellik vektörleri ile *codebook* oluşturma aşamasıdır. Türkçe korpüsten alınan *phon*'lardan yararlanılarak Türkçe iki boyutlu *codebook, Kohonen Self Organizing Feature Map* algoritması kullanılarak oluşturulmuştur. Sistemde işlem gücü gereğini azaltmak için iki aşamalı *codebook* oluşturma yoluna gidilmiştir. *Codebook* oluşturulurken ilk aşamada her düğüm için komşulukların içine alındığı dairesel alanın yarıçapı, tüm düğümleri içine alacak kadar büyük tutulur. Ancak bu durumda algoritmanın hızlı bir biçimde yakınsaması için iterasyon sayısı düşük, hata oranı büyük bir değerde tutulur. İkinci aşama ilk aşamada elde edilen *codebook* üzerine uygulanır. Bu aşamada hata oranı düşürülürken iterasyon sayısı ilk aşamaya göre büyütülür. Bu biçimde elde edilen *codebook* üzerinde hangi *phon*'ların hangi fonemlere karşılık geldiği bilinmemektedir. Bu nedenle yürütülen paralel bir çalışma ile fonemlerin ya da fonem altı birimlerin simgeleri elde edilen kümelerle ilişkilendirilir. Etiketlenmiş özellik vektör kütüğü *codebook* ile birleştirilir. Sonuçta ortaya etiketlenmiş *codebook* kütüğü çıkar. *Codebook*'un etiketlenmesi aşamasından sonra Türkçe *fonotopic map* (fonem dağılımlı *codebook*) elde edilmiş olur. Daha sonra etiketlenmiş *codebook,* tanıma aşamasında referans vektör tablosu olarak kullanılır.

Sesli ifade özellik vektörleri, Türkçe sesli ifade tanıma laboratuvarı yazılım kütüphanesinde yeralan grafik tabanlı bir yazılım ile tanıma işlemlerine tabi tutulmuştur. *Codebook*'ta yer alan vektörlerin, kimi fonemler için fonem altı birimlerden oluşması durumunda tanıma başarımının artmasını sağladığı gözlenmiştir. Bu gerekçeye dayalı olarak *codebook*'ta 300 vektöre yer verilmiştir. Bu sayı, iki boyutlu, 20 x 15 hexagonal biçimde dizilmiş nöron ağı yapısındaki nöron sayısına karşılık gelir. Her bir düğüm *codebook*'un bir vektörünü kullanır. Bu biçimde her düğüm, bir fonem altı ses birimini temsil etmede kullanılmaktadır.

Ancak bu fonem altı sesbirimlerin içinde doğrudan bir harfe karşılık gelmeyen birimler de bulunmaktadır. Bu birimler genelde sesli ifadenin anlam taşıyan kesimlerinin dışındaki kesimlere ilişkin vektörlerdir.

Tanıma kesimi, alıştırma kesimine göre daha yalındır. Ardışık biçimde gelen sesli ifade özellik vektörleri alıştırma aşamasında kurulan *codebook'*taki tüm düğümler ile karşılaştırılır. En küçük uzaklık değerini veren düğüm, sonuç düğümü olarak çıkışa yansıtılır. Bu düğümün etiketi de ilgili fonemin simgesini taşıdığından çıkış simgesi Türkçe için harf olmaktadır. Tanıma işleminin sonucu, X-Windows altında çalışan bir programca üretilen, tüm düğüm çıkışlarını gösteren grafik üzerinde yer almaktadır. Bu tür grafikler, Çizim 7.22-49 arasında etiketlenmiş *codebook*'u ve Türkçe fonemlerin düğümler üzerindeki dağılımını vermek için kullanılmıştır.

8. SONUÇ, TARTIŞMA VE ÖNERİLER

Türkçe fonem kümeleme sistemi tasarım ve gerçekleştirimi adlı bu tez kapsamında sesli ifade tanımada kullanılan temel yaklaşımlar incelenmiş, ayrışık sözcük tanıma ve sözcük altı ses birimleri tanıma yaklaşımlarından, Türkçe için fonem tanıma tercihi yapılmıştır. Bu tez kapsamında değişik kesimlerde de dile getirildiği üzere, ismin i, e, de ,den halleri, fiil çekimi gibi nedenlerle çok yoğun biçimde sonek alan bir dil olması bu tercihin temel gerekçesini oluşturmaktadır. Zira Türkçe'de kökler sonekler alarak, diğer dillerde ancak birkaç tümce ile ifade edilebilen anlam

yoğunluğunda sözcüklere dönüşebilmektedir. Bilindiği üzere ayrışık sözcük tanıma yaklaşımının kullanıldığı sistemlerde sözlük kullanımı temel alınmakta ve tanımaya çalışılan sesli ifadeler (sözcükler) bu sözlük içinde aranmaktadır. Bu yaklaşımla Türkçe'de, sonek alanlar da dahil olmak üzere olası tüm sözcükler için bir sözlük oluşturma olanağı bulunmadığından bu yaklaşımın genel amaçlı bir Türkçe sesli ifade tanıma için kullanılması olanaksızdır. Bu zorunluluktan dolayı en küçük sesli ifade birimi olarak fonem alınmıştır. Bu yaklaşım Türkçe'nin fonemik bir dil olması itibarıyla sesli ifadelerden yazılı metinlere geçişi de kolaylaştıracak bir yaklaşım olarak düşünülmüştür. Bu düşünce de yapılan tercihi etkileyen bir etmen olmuştur.

Bu bağlamda, önce, ZÜZ türünde tek heceli 47 ve Ü, ÜZ, ÜZZ, ZÜ, ZÜZ, ZÜZZ türünde iki ve daha çok heceden oluşan 139 sözcüklük deneysel Türkçe korpüs hazırlanmıştır. Korpüs içerisinde tek heceli sözcükler yalın çiftler(*minimal pair*) olarak ünlü seslerin, diğerleri ise daha çok ünsüz seslerin özellik vektörlerinin hesaplanması için kullanılmıştır.

Özellik vektörlerinin belirlenmesi için aşağıdaki özellik vektörleri çıkarma yöntemleri incelenmiştir. Bunlar sırasıyla şunlardır:

- *FFT*'ye dayalı sıklık değerlerine dayalı özellik çıkarma yöntemi
- *Mel* skalasında *FFT*'ye dayalı cepstrum katsayılarına dayalı özellik çıkarma yöntemi
- *LPC* parametrelerine dayalı özellik çıkarma yöntemi
- *LPC* -autocorrelation parametrelerine dayalı özellik çıkarma yöntemi
- *LPC* -reflection coefficient parametrelerine dayalı özellik çıkarma yöntemi,
- *LPC* parametrelerine dayalı filtre dizisi yöntemi
- *LPC*'parametrelerine dayalı *Smoothed Group Delay Spectrum* yöntemi

Bu yöntemlerden Kohonen *Learning Vector Quantization* algoritması ile yürütülen fonem kümeleme işlemlerinde en yüksek başarımı *Mel* skalasında *FFT*'ye dayalı *cepstrum* katsayıları yöntemi vermiştir. Bu nedenle bu tez çalışması kapsamında özellik vektörleri bu yöntem kullanılarak hesaplanmıştır. Türkçe *codebook*'un oluşturulmasında bu yöntem kullanılagelmiştir.

Alıştırma aşamasında, Türkçe korpüsten alınan *phon*'lardan yararlanılarak Türkçe iki boyutlu *codebook* Kohonen *Self Organizing Feature Map* algoritması kullanılarak oluşturulmuştur. *Codebook*'un etiketlenmesi aşamasından sonra Türkçe fonemlerin *fonotopic map* (fonem dağılımlı *codebook* çizimleri) elde edilmiştir. Etiketlenmiş *codebook*'un tanıma aşamasında referans vektör tablosu olarak kullanılması amaçlanmıştır. Bir sesli ifadeye ilişkin özellik vektörleri, Türkçe sesli ifade tanıma labaratuvarı yazılım kütüphanesinde yeralan *gavisual* adlı grafik tabanlı bir yazılım ile kümeleme işlemlerine tabi tutulmuştur. *Codebook*'ta yer alan vektörlerin, kimi fonemler için fonem altı birimlerden oluşması durumunda kümeleme başarımının artmasını sağladığı gözlenmiştir. Bu gerekçeye dayalı olarak *codebook*'ta yaklaşık 300 vektöre yer verilmiştir.

Fonem olup olmadığı tartışma konusu olan *ğ*'nin yer aldığı örnekler özellikle korpüsün dışında tutulmuştur. *ž*'nin başlı başına *phon* olup olmadığı, ya da diğer *phon*'lardan mı oluştuğu ileriki çalışmalarda konu edilecektir. Bu çalışma sonrasında yapılması gereken ilk araştırma elde edilen *phonotopik map*'ları kullanarak yürütülecek sistemli tanımanın aşaması ölçülerine ilişkin olmalıdır.

Bu çalışmada, *Transputer* donanımı ve bunun üzerinde paralel kod üreten derleyicilerin kullanımında deneyim kazanma yoluna gidilmiş ancak uygulama aşamasına tam geçilememiştir. Sözkonusu bu donanım üzerindeki uygulama, nöron ağı yaklaşımını içine alan tanıma aşamasının *Transputer*'lü sistem üzerinde çalıştırılması biçiminde olmalıdır. Bu bağlamda tanıma kesimindeki programların bağımsız çalışabilen kesimleri ayrı işleyiciler üzerinde çalıştırılarak işlem hızının göreli artması sağlanabilecektir.

Bu çalışmanın devamı niteliğinde yapılabilecek bir diğer çalışmada Türkçe korpüs'ün genişletilmesi sözkonusu edilmelidir. Bunun yanı sıra korpüsün birden çok konuşmacıya ilişkin verileri taşıması ve bu yolla konuşmacıdan bağımsız sesli ifade tanıma sistemleri üzerinde araştırma yapılması gerekmektedir.

Yapılması gerekli diğer bir çalışma da Türkçe sesli ifade tanımada elde edilen birikimin, dilden bağımsız sesli ifade tanıma çalışma gruplarınca yürütülen projelerle birleştirilmesidir. Bu bağlamda değişik gruplarca üzerinde çalışılan korpüslerin karşılaştırılarak incelenmesi gerekmekte ve bu gruplarla paralel ve ortak çalışma yolları aranmalıdır. Özellikle çalışma gruplarınca *CD-ROM*'lar üzerinde dağıtılan ses veri tabanlarında Türkçenin de yer alması sağlanmalıdır. Bunun için kayıt

koşullarının iyileştirilmesi (sessiz oda ve kayıt aletleri gibi) ve belirli bir standarta ulaştırılması gerekmektedir.

Sesli ifade tanıma çalışmalarından elde edilen birikimden, Türkçe yapay sesli ifade oluşturma konusunda yararlanılması sağlanmalıdır. Bu bağlamda elde edilen *codebook*'un sesli ifade oluşturma için denenmesi, hem *codebook*'un sınanması hem de yapay sesli ifade oluşturmaya katkı bakımından yararlı olabilecektir.

Sesli iletişim ortamının, örneğin telefon hattı, kalabalık bir ortam gibi, niteliğine göre araştırma genişletilmelidir.

Bu çalışmanın devamında yapılması gerekli bir diğer araştırma, tanınan sesli ifadelerden yazılı metinlere geçişi sağlayacak anlayışlı ya da uzman sistem kesimine ilişkin olmalıdır. Bu bağlamda bu çalışmanın Türkçe üzerinde yürütülen anlamsal ve biçimsel *(semantic and morphologic)* çalışmalarla birlikte düşünülmesi gerekecektir.

KAYNAKLAR

Aksan, D., 1980, Her Yönüyle Dil -Ana Çizgileriyle Dilbilim, Türk Dil Kurumu Yayınları.

Aleksander, I., 1989, Neural Computing Architecture, North Oxford Academic.

Alspector, J., Allen, R.B., 1987, A Neuromorphic VLSI Learning System, Proc. of the 1987 Stanford Conf.: Adv.Res. in VLSI.

Alspector, J., 1988, Research Result in VLSI Implementations of Neural Networks, Conferance Acoustical Soc. of Japan.

Amari, S.I., 1972, Characteristics of Random Nets of Analog Neuron-Like Elements, IEEE Transactions on Systems Man and Cybernetics, 11/2.

Anderson, J.A., 1983, Cognitive and Psychological Computation with Neural Models, IEEE Transactions on Systems Man and Cybernetics, 1/13.

Artuner, H., Savcı, F., Saatçi, A., 1994, Performance of Speech Feature Extraction Techniques for Turkish Speech Recognition, ISCIS IX, Antalya.

Auger, J.M., 1991, Parallel Implementation on Transputers of Kohonen's Algorithm, NATO ASI Series, Neurocomputing, 215-226.

Barto, A., 1981, Assosiative Search Network: A Reinforcement Learning Associtive Memory, Biological Cybernetics, 4, 201.

Banks, S., 1990, Signal Processing, Image Processing and, Pattern Rcognitiıon, Prentice Hall.

Baum, E.B., 1986, Internal Representations for Associative Memory, University of California, 86-.

Bengio, Y., Cardin, R., 1989, Programmable Execution of Multi Layered Networks For automatic Speech Recognition., Communications Of The ACM, 2/32, 195.

Blelloch, G., Rosenberg, C.R., 1987, Network Learning on the Connection Machine, Ablex Publishing Corp.

Bose, N.K., 1993, Neural Network Design Using Voronoi Diagrams, IEEE Trans. Neural Networks, 4/5, 778-787.

Brause, R., 1989, Using Neural Networks, Euromicro-Microprocessing and Microprogramming Journal, 8/27, 179.

Buda, O.R., Hart, P.E., 1973, Pattern Classification and Scene Analysis, John Wiley & Sons.

Caianiello, E.R., 1989, Parallel Architectures and Neural Networks, Word Scientific.

Camp, D., 1993, A Users Guide for the Xerion Neural Network Simulator. Version 3.1, University of Toronto.

Carpenter, G.A., Grossberg, S., 1986, Neural Dynamics of Category Learning and Recognition:Attention Memory Consolidation and Amnesia, AAAS Symposium Series, and Amnesia.

Carpenter, G.A., Grossberg, S., 1986, Assosiative Learning adaptive Pat. Recog. and Cooperative-competitive desicion making by NN, SPIE, 634-.

Carpenter, G.A., Grossberg, S., 1988, The ART of Adaptive Pattern Recognition by a Self_Organizing Neural Network, IEEE Computer, 3/21, 77.

Chen, C.H., 1973, Statistical Pattern Recognition, Hayden Book Company,

Chua, L.O., Yang, L., 1988, Cellular Neural Networks:Theory, IEEE Transactions on Circuits and Systems, 1/35, 1257.

Chua, L.O., Yang, L., 1988, Cellular Neural Networks:Applications, IEEE Transactions on Circuits and Systems, 1/35, 1273.

Chomsky, N., Halle, M., 1968, The Sound Pattern of English, Harper and Row, NewYork.

Cohen, M.A., Grossberg, S., 1983, Absolute Stability of Global Pattern Formation & Parallel Memory Storage by Competitive NN, IEEE Transactions on Systems Man and Cybernetics, 1/13.

Cooke, M., Crawford, M., 1993, Vısual Reprasentation of Speech Signals, John Wiley and Sons Ltd.

d'Alessandro, C., Sylvain, J., 1988, Decomposition of The Speech Signal into Short-Time Waveforms Using Spectral Segmentation, IEEE CH 2651-9, 351-354.

Davis, S.B., Mermelstein, P., 1980, Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, IEEE transaction on ASSP, 28/4, 357-367.

Delgutte, B., 1984, Speech Coding in the Auditory Nerve:II.Processing Schemers for Vowel-Like Sounds, Journal of Acoust. Soc. Am. 75, 879-.

Demircan, Ö., 1979, Türkiye Türkçesinin Ses Düzeni Türkiye Türkçesinde Sesler, Türk Dil Kurumu Yayınevi.

Demirezen, M., 1986, Phonemics and Phonology: Theory through Analysis, Bizim Büro Yayınevi.

Demirezen, M., 1987, Articulatory Phonetics and the Principles of Sound Production, Yargı Yayınevi.

Devijer, P.A., Kittler, J., 1977, Pattern Recognition: A Statical Approach, Prentice Hall.

Devijver, P.A., Kittler, J., 1977, PAttern Recognition: A Statical Approach, Prentice Hall.

Doddington, G.R., Schalk, T.B., 1981, Speech Recognition: Turing Theory to Practice, IEEE SPECTRUM, 9/18, 26.

Ellis, E.M., Robinson, A.J., 1993, A Phonetic Tactile Speech Listening System, Cambridge Univ./F-INENG, TR122.

Elman, J.L., Zipser, D., 1987, Learning the Hidden Structure of Speech, Univ. of California at San Diego ICS Report 8701, 2/2.

Ergenç, İ., 1989, Türkiye Türkçesinin Görevsel Sesbilimi, Engin yayınları.

Erman, L.D., Hayes, R.F., 1987, THE HEARSAY-II Speech-Understanding System:Integrating Knowledge to Resolve Uncertainty, Computing Surveys, 12/2, 214.

Fahlman, S.E., Hinton, G.E., 1987, Connectionist Architectures for Artifical Intelligence, Computer, 1, 100.

Fahlman, S.E., 1988, An Empirical Study of Learning Speed in Back-Propagation Networks, CMU Techincal Report, 5, 88-.

Fant, G., 1959, The Acoustics of Speech, Proceedings of the Third International Congress on Acoustics.

Feldman, J.A., 1982, Dynamic Connections in Neural Networks, Biological Cybernatics, 27-.

Feldman, J.A., Ballard, D.H., 1982, Connectionist Models and Their Properties, Cognitive Science, 6, 205-.

Feldman, J.A., Goddard, N.H., 1988, Computing with Structured Connectionist Networks, Communications of ACM, 2/31, 170-.

Feldman, J.A., Fanty, M.A., 1988, Computing with Structured Neural Networks, IEEE, 3/21, 91-.

Flanagan, J.L., 1972, Voices of Men and Machines, Journal of Acoustical Society of America.

Fukushima, K., Ito, T., 1983, Neocognitron:A Neural Network Model Model for a Mechanism of Visual Pattern Recognition, IEEE Transactions on Systems Man and Cybernetics, 1, 13-.

Fukushima, K., 1988, A Neural Network for Visual Pattern Recognition, IEEE Computer, 3/21.

Gallager, R.G., 1968, Information Theory and Reliable Communication, John Wiley & Sons.

Gallant, S.I., 1988, Connectionist Expert Systems, Communication of ACM, 2/31, 152-.

Genvis, A.S., Morgan, N.H., 1988, Applications of Neural-Network (NN) Signal Processing in Brain Research, IEEE Transactions on Acoustics Speech and Sig.PR, 7, 36-.

Glass, J.R., Zue, V.W., 1988, Multi-Level Acoustic Segmentation of Continuous Speech, IEEE CH2561-9, 429-432.

Goddard, N. H., 1989, The Rochester Connectionist Simulator: User's Manual, University of Rochester Technical Report.

Gold, B., 1987, Hopfield Model Applied to Vowel and Consonant Discrimination, MIT Lincoln Laboratory Tecnical Report, 4.

Gold, B., 1988, A Neural Network for Isolated-Word Recognation, IEEE CH2561-9,44-47.

Graf, H.P., Jakel, L.D., 1986, VLSI Implementation of a Neural Network Memory with Several Hunderts of Neurons, Am. Inst. of Pysics Conferance Proceeding 151, 182-.

Graf, H.P., deVegvar, P., 1987, A CMOS Implementation of a Neural Network Model, Proc. Stanford Conf. Advanced Res. in VLSI, 351-.

Graf, H.P., Jakel, L.D., 1988, VLSI Implementation of a Neural Network Model, IEEE Computer, 3/21, 41-.

Grant, P.M., Sage, J.Q., 1986, A Comparison of Neural Network and Matched Filter Processing for Detecting Lines in Ymages, Neural Networks for Computing, AIP.

Gürgen, F., 1992, Phoneme Recognition Neural Networks, ISCIS 7,569-573.

Habib, M.K., Akel, H., 1988, Logic Gate Formed Neuron Type Processing Element, IEEE ISCAS'88, 491-.

Haffner, P.A., Waibel, A., 1988, Fast Back Propagation Learning Methods for Neural Networks in Speech, ATR Telephone Lab. Tech. Report.

Hammerstrom Dan, D., 1988, A Connectivity Analysis of a Class of Simple Associative Neural Networks, Technical Report No. CS/E-86-009 Oregon Gra.Cent.

Hinton, G.E., Rumelhart, D.E., 1988, Neural Network Architecture for Artificial Intelligence, Tutorial AAAI.

Hirai, Y., 1983, A Model of Human Associative Processor (HASP), IEEE Transactions on Systems Man and Cybernatics, 1, 13.

Hoggle, G., 1992, An FFT-Based Speech Recognition System, Journal of the Franklin Institute, 329/3, 555-562.

Hopfield, J.J., 1982, Neural Networks and Physical Systems with Emergent Collevtive Computational Abilities, Proc. Natl. Acad. Sci. USA, 4/79, 2554-.

Hopfield, J.J., 1984, Neurons with Graded response have Collective Computational Properties Like Those of 2-State Neurons, Procedings National Academy of Science, 5/81, 3088.

Hopfield, J.J., 1984, Neuronswith Graded Response Have Collective Computational PropertiesLike Those of Two State Neurons, Proc. Natl. Acad. Sci. USA, 5, 81-.

Hopfield, J.J., Tank, D.W., 1985, Neural Computation of Decisions in Optimization Problems, Biological Cybernetics, 141-.

Hopfield, J.J., Tank, D.W., 1986, Computing with Neural Circuits: A Model, SCIENCE, 8, 625-.

Hopfield, J.J., Tank, D.W., 1987, Collective computation in neuronlike circuits, Scientific American, 6, 104-.

Horio, Y., Nakamura, S., 1988, Speech Recognition Network with SC Neuron-Like Components, IEEE ISCAS'88, 495-.

Huang, W., Lippmann, R., 1988, A Neural Net Approach to Speech Recognition, IEEE 2561-9, 99-102.

Hubel, D.H., Wiesel, T.N., 1979, Brain Mechanisms of Vision, Scientific American, 8/24, 150.

Hutchinson, J., Koch, C., 1988, Computing Motion Using Analog and Binary Resistive Networks, IEEE Computer, 3/21, 52.

İnce, N., 1992, Digital Speech Processing: Speech Coding, Synthesis and Recognition, Kluwer Academic Publishers.

James, D.A., Young, S.J., 1994, A Fast Lattice-Based Approach to Vocabulary Independent Wordspotting, ICASSP.

James, M.R., 1992, Design of Low-cost, Real-time Simulation Systems for Large Neural Networks, MS thesis at the University of Sydney.

Jones, M., Woodland, P.C., 1993, Using Relative Duration ın Large Vocabulary Speech Recognition, Proc. EuroSpeech'93.

Joseph, W.P., 1993, Signal Modelling Techniques in Speech Recognition, Proceedings of the IEEE, 81/9, 1215-1247.

Judd, J.S., 1987, Complexity of Connectionist Learning with Various Node Functions, Proceeding IEEE Int'l. Conf. on Neural Networks, 6.

Kadirkamanathan, K., Niranjan, M., 1992, Application of an Archıtectureally Dynamic Network for Speech Pattern Classification, Proceedings of the Instutude of Acoustics, 14/6, 343-350.

Kangas, J., 1994, On the Analysis of Pattern Sequences by Self-Organizing Maps, Doktora Tezi, Helsinki Üniversitesi, Finlandiya.

Kitano, H., 1991, An Experimental Speech to Speech Dialog Translation System, IEEE Computer, 24/6.

Klatt, D.H., 1977, Reviev of the ARPA Speech Understanding Project, Journal of Acoustic Soc. American, 12, 1345,

Kohonen, T. , 1982, Self-Organized Formation of Topologically Correct Feature Maps, Biological Cybernetics, 43, 59-69,

Kohonen, T., 1984, Self-Organization and Associative Memory, Springer-Verlag.

Kohonen, T., Mas, I.K., 1984, Phonotopic Maps-Insightful Representation of Phonological Peatures for Speech Representation, Proceedings IEEE Inter. Conf. on Pattern Recognition.

Kohonen, T., 1986, Dynamically Expanding Context with Application to the Correction of Symbol Strings in the Rec. CS., Proceeding 9. Int. Conf. Pattern Recognition IEEE.

Kohonen, T., 1988, The Neural Phonetic Typewriter, IEEE Computer, 3/21, 11.

Kohonen, T., Torkkola, K., 1988, Phonetic Typewriter for Finnish and Japanese, IEEE CH2561-9, 607-610.

Kohonen, T., 1990, The Self-Organizing Map, IEEE, 78/9, 1464-1480.

Korb, T., Zell, A., 1989, A Declarative Neural Network Description Language, Microprocessing and Microprogramming, 8/27, 181-.

Kosko, B., 1987, Constraction an Associative Memory, BYTE, 137-.

Kosko, B., 1992, Neural Networks for Signal Processing, Prentice Hall.

Kuhn, R., 1992, A Cache-Based Language Model for Speech Recognition, IEEE Transactions on Pattern Recognition and Machine Intelligence, 14/6, 691-692.

Lang, K.J., Withbrock, M.J., 1988, Learning to Tell Two Spirals Apart, Proceedings of the Connectionist Models Summer School.

Lang, K.J., Waibel, A.H., 1990, A Time-Delay Neural Network Architecture for Isalated Word Recognition, Neural Networks, 3, 33-43.

Lea, W.A., 1980, Trends in Speech Recognition, Prentice Hall,.

Lee, K.F., 1989, Automatic Speech Recognition. The Development of the SPHINX System, Kluwer Academic Publishers.

Leighton, R. R., 1992, The Aspirin/MIGRAINES Neural Network Software User's Manual, Mitre Corparation.

Leung, H., Zue, V.W., 1988, Some Phonetic Recognation Experiments Using Artifical Neural Nets, IEEE CH2561-9, 422-425.

Levinson, S.A., Rabiner, L.R., 1983, An Introduction to the Application of the theory of Probabilistic Functions of a Marcov Process ASR., Bell System Technical Journal, 4.

Lewis, F.L., 1986, Optimal Estimation, John Wiley & Sons.

Liebert, P.B., 1967, An Introduction to Optimal Estimation, Addison Wesley Reading Mass.

Linsker, R., Towards an Organizing Princeble for a Layerd Perceptural Network, Netural Information Processing System.

Linsker, R., 1988, Self_Organization in a Perceptual Network, IEEE, 3/21, 105.

Lippmann, R.P., Gold, B., A, 1986, Comparison of Hamming and Hopfield Neural Nets for Pattern Classification, MIT Lincoln Laboratory Technical Report.

Lippmann, R.P., 1987, An Introduction to Computing with Neural Nets, IEEE ASSP MAGAZINE, 4/4.

Lyon, R.F., Loeb, E.P., 1987, Isolated Digit Recognition experiments with a Cochlear Model, ICASSP-87, 4.

Makhoul, J., Roucos, S., Vector Quantization in Speech Coding, IEEE Proceedings, 73, 1085.

Marks II, R.J., Atlas, L.E., 1988, Generalization in Layerd Classification Neural Networks, IEEE ISCAS'88.

Martin, T., Acoustic Recognition of a Limited Vocabulary in Continuous Speech, Dept. Elec. Eng. Univ. Pennsylvania, 197-.

McClelland, J.L., Rumelhart, D.E., 1986, Paralel Distributed Processing, MIT Press.

McClelland, J.L., Rumelhart, D.E., 1987, Explorations in Parallel Distributed Processing, Cambridge.

McClelland, J.L., Rumelhart, D.E., 1986, Paralel Distributed Processing 1, MIT Press.

McClelland, J.L., Rumelhart, D.E., 1987, Parallel Distributed Processing 2, MIT Press.

McEliece, R.J., Posner, E.C., 1987, The Capacity of the Hopfield Associative Memory, IEEE Transaction on Information Theory, 7/33, 461.

Minsky, M., Papert, S., 1969, Perceptrons:An Introduction ta Computational Geometry, MIT Press.

Miyata, Y., 1991, A Users Guide to PlaNet Version 5.6, University of Colorado.

Moopenn, A., Lambe, J., 1987, Electronic Implementation of Associative Memory Based on Neural Network Models, IEEE Transactions on Systems Man and Cybernetics, 3/17.

Morris, L.R., 1988, A PC-Based Digital Speech Spectrograph, 8/6, 68-85.

Muller, P., Lazzaro, J., 1986, A Machine for Neural computation of Acoustical Patterns with App.To Real-Time Speech Recognition, AIP Conference Proceedings, 151-.

Muveit, H., Weintraub, M., 1988, 1000-Word Speaker-Independent Continuous-Speech Recognition Using Hidden Markov Models, IEEE CH2651-9,115-118.

Mustafa, A., Pslatis, D., 1987, Optical Neural Computers, Scientific American, 3/3, 88-.

Mustafa, A., Jaques, ST., 1985, Information Capacity of the Hopfield Model, IEEE Transaction on Information Theory, 7/31, 461-.

National Semiconductor Corparation, 1985, Switched-Capacitor Filter Handbook.

Nielsen, R.H., 1988, Neurocomputing: Picking the Human Brain, IEEE Spectrum, 3/25, 36.

Nijhuis, J.A.G., Spaanenburg, L., 1989, On Fault Tolarance of Neural Assosiative Memories, IEE Journal E.

Nijhuis, J.A.G., Spaanenburg, L., 1989, Structure and Application of NNSIM: a general_purpose Neural Network SIMulator, Microprocessing and Microprogramming, 8/27, 189.

Norusis, M.J., 1990, SPSS/PC+Statistics 4.0, SPSS Inc.

Oberne, K., Barron, J.J., 1989, Time to Get Fried Up, BYTE, 8/14, 217.

O'Connor, J.D., 1973, Phonetics, Penguin Books.

Oppenheim, A.V., 1989, Discrete-Time Signal Processing, Prentice Hall.

Paik, E., Gungner, D., 1987, UCLA SFINX A Neural Network Simulation ment, IEEE First Int. Conf. on NN Proceeding, 367-.

Papamichalis, P., Simar, R., 1988, The TMS320C30 Floting-Point Digital Signal Processor, IEEE Micro, 8/6, 13-29.

Parker, D.B., 1986, A Comparison of Algorithms for Neuron-Like Cells, AIP Conference Proceddings 151.

Parson, T., 1986, Voice and Speech Processing, McGraw Hill.

Paulos, J.J., Hollis, P.W., 1988, Neural Networks Using Analog Multipliers, ISCAS'88, 499-.

Pelling, S.M., Moore, R.K., 1986, The Multi-Layer Perceptron as a Tool for Speech Pattern Processing Research, Proc.IoA Autumn Conf. on Speech and Hearing.

Petre, P., 1985, Master:Typewriters that take Dictation, FORTUNE, 1.

Picone, J., 1990, Continuous Speech Recognition Using Hidden Markov Models, IEEE ASSP Magazine.

Picone, J., 1993, Signal Modeling Techniques in Speech Recognition, Proceedings of the IEEE, 81/9, 1215-1247.

Plonski, M., Joyce, C., 1993, RCS, GENESIS, and SFINX: Three "Public-Domain" Simulators for Neural Networks.

Rabiner, L.R., February 1989, A Tutorial on Hidden Markov Models and Selected Applications in Speech Processing, Proceedings of the IEEE, 77/2,

Rabiner, L.R., Schafer, R.W., 1975, Digital Repres-entations of Speech Signals, Proceedings of the IEEE.

Rabiner, L.R., Schafer, R.W., 1978, Digital Processing of Speech Signals, Prentice-Hall.

Rabiner, L.R., 1989, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, 2/77, 257,

Rabiner, L.R., 1994, Application of Voice Processing to Telecommunications, Proceedings of the IEEE, 2/82, 199-228.

Reddy, R., Zue, V., 1983, Recognizing Continuous Speech Remains an Elusive goal, IEEE Spectrum, 11, 84.

Regency, H., 1992, International Conferance on Signal Processing Applications and Technology, DSP Associates.

Reid, C.E., 1992, Signal Processing in C, John Wiley and Sons.

Rich, E., Knight, K., 1991, Artifical Intelligence, Mc Graw Hill.

Robinson, T., 1989, Dyanamic Error Propagation Networks, Doktora Tezi, Cambridge Universitesi, İngiltere.

Robinson, T., Fallside, F., 1990, A Recurrent Error Propagation Network Speech Recognition System, Computer Speech and Language Nov.1990, 5/3.

Robinson, T., Fallside, F., 1990, Phoneme Recognition from the TIMIT database using Recurrnt Error Propagation Networks, Cambridge Univ./F-INENG, TR42.

Robinson, T.1991, Several Improvements to a Recurrent Error Propagation Network Phone Recognition System, Cambridge Univ./F-INENG, TR82,

Robinson, T., 1992, A Real-Time Recurrent Error Propagation Network Word Recognition System, IEEE - ICASSP.

Robinson, T., 1992, Recurrent Nets for Phoneme Probability Estimation, Cambridge Univ./F-INENG.

Robinson, T., 1992, Practical Network Design and Implementation, Cambridge Univ./F-INENG.

Robinson, T., 1992, The State Space and "Ideal Input" Representations of Recurrent Networks, ESCA92.

Robinson, T., 1992, Artifical Neural Networks:The Mole-Grips of the Speech Scientist, ESCA92.

Robinson, A.J., Almedia, L., 1993, "A Neural Network Based, Speaker Independent, Large Vocabulary, Continuous Speech Recognition System: The WERNICE project", Proc. EuroSpeech'93.

Rumelhart, D.E., Hinton, G.E., 1985, Learning Representations by Back-Propagating Errors, NATURE, 1/27, 533.

Rumelhart, D.E., Hinton, G.E., 1986, Parallel Distributed Processing : Vol 1, MIT Cambridge, 318-.

Sage, J.P., Thompson, K., 1986, An Artifical Neural Network Integrated Circuit Based on MNOS/CD Principles, AIP Conferance Proceedings N.151.

Savcı, Y.F., 1994, Sesli İfade Tanıma için Otomatik bir Özellik Çıkarım Sistemi Tasarım ve Gerçekleştirimi, Y.Müh. Tezi, H.Ü. Ankara.

Schalkoff, R.J., 1992, Pattern Recognition: Statistical Structural and Neural Approaches, John Wiley and Sons.

Schalkoff, R., 1992, Pattern Recognition: Statistical Structural and Neural Approaches, John Willey and Sons Inc.

Scroeder, M.R., Hall, J.L., 1974, Model for mechanical to neural transduction in the auditory receptor, Journal of Acoustic Soc. American., 5, 1055.

Seneff, S., 1986, A Computational Model ofr the Peripheral Autitory System:Application to Speecn Recognition Research, ICASSP-86.

Senjnowsiki, T:, Rosenberg, C.R., 1986, NETtalk:A Parallel Network That Learns to Read Aloud, Univ. Techinal Report JHU/EECS-86/01, 1096-.

Shannon, C.E., 1948, A Mathematical Theory of Communication, Bell Systems Technical Journal, 27, 379-623.

Shiffman, S., Wu, A.W., February 1991., Building a Speech Interface to a Medical Diagnostic System, IEEE Expert, 6/1.

Shriver, B.D., 1988, Artificial Neural Systems, IEEE Computer, 3/21, 8.

Sivilotti, M.A., Emerling, M.R., 1986, VLSI architecture for Implementation of Neural Networks, Am. Inst. of Pysics Conferance Proceeding 151, 408-.

Sivilotti, M.A.,Mahowald,M.A.,1987,Real Time Visual Computations Using Analog CMOS Processing Arrays, Advanced Research in VLSI, MIT Press, 295-.

Stanley, H.L., 1984, MultipleValued Logic Its Status and Its Future, IEEE Transactions on Computer, 1/33, 1160-.

Sutton, R., Barto, A., 1981, An Adaptive Network That Constructs and Uses an Internal model of Its World, Cognition and Brain Theory, 4/3, 217-.

Tank, W., Hopfield, J., 1986, Simple \Neural\ Optimization Networks: An A/D Converter, Signal Devision Circuit and a Li IEEE Transactions on Circuits and Systems, 5, 33-.

Tank, W., Hopfield, J., 1989, Collective Computation in Neuronlike Circuits, Scientific American, 8, 62-.

Tazelaar, M., 1989, Neural Networks, BYTE, 8/14, 214-.

Tenorio, M.F., Huges, C.S., 1987, Real Time Noisy Image Segmentation Using an Artifical Neural Network Model, Proceeding of the IEEE 1.Int. Conf.on NN, 4, 357.

Tenorio, M.F., Tom, M.D., 1989, Adaptive Networks as a Model for Human Speech Development, Purdue University, TR-EE 89-54.

Terry, M. Renalds, S., 1988, A Connectionist Approach To Speech Recognition Using Peripheral Auditory Modelling, IEEE CHI2561-9, 699-702.

Texas Instruments, 1991, Digital Signal Processing Applications with the TMS320C30 Evaluation Module, Texas Instruments.

Texas Instruments, 1990, TMS320C30 Evaluation Module Technical Referance, Texas Instruments.

Texas Instruments, 1992, TMS320C3x Users Guide, Texas Instruments.

Torkkola, K., 1988, AUtomatic Alignment of Speech with Phonetic Transcriptions in Real Time, IEEE CH2561-9, 611-614.

Touretzky, D.S., Hinton, G.E., 1985, Symbols Among Neurons:Details of Connectionist Inference Architecture, Proceeding IJCAI Los Angeles, 238-.

Touretzky, D.S., Pomerleau, D., 1989, What's Hidden in the Hidden Layers?, BYTE, 8/14, 227-.

Touretzky, D.S., Elman, J.L., 1990, Connectionist Models. Proceedings of the 1990 Summer School, Morgan Kaufmann Publishers.

Van Der Kam, J., 1986, A Digital Decimating Filter for Analog-to-Digital Conversion of Hi-Fi Audio Signals, Phillips Technical Review, 42, 6/7.

Vemuri, E., 1988, Artifical Neural Networks:An Introduction, IEEE, 1.

Verhaeghe, B., 1992, Toward Continuous-Speech Recognition, BYTE , 17/4, 158-158.

Waibel, A., Hanazawa, T., 1988, Phoneme Recognition: Neural Networks vs. Hidden Markov Models, IEEE CH2561-9, 107-110.

Waibel, A., Hanazawa, T., 1989, Phoneme Recognition Using Time-Delay Neural Networks, IEEE Transactions on Acoustics Speech&Signal Proc, 3, 37-.

Waibel, A., Hampshire, J., 1989, Building Blocks for Speech, BYTE, 8/14, 235-.

Wallance, D., 1986, Memory and Learning in a Class of Neural Models, Proceedings of the Workshop on Lattice Gauge Theor.

Waltz., D.L., 1987, Applications of the Connection Machine, IEEE Computer, 1, 85.

Wan Der Enden, W.M., 1989, Discrete-time Signal Analysis: an Introduction, Prentice-Hall.

Watrous, R., 1988, Speech Recognition Using Connectionist Networks, Dept. of Comp. and Information Science, Uni. Penn.

White, M.W., Holdaway, R.M., 1990, New Strategies for Improving Speech Enhencement, Int. J. Biomed. Comp., 25/2-3, 101.

Widrow, B., Winter, R., 1988, Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition, IEEE Computer, 32/21, 25.

Widrow, B., Winter, R., 1988, Layered Neural Nets for Pattern Recognition, IEEE Transaction on ASSP, 36/7, 1109-1118.

Wu, J., Chan, C., 1993, Isolated Word Recognition by Neural Network Models with Cross-Correlation Coefficients for Speech Dynamics, IEEE Trans. Pattern Analysis and Machine Intelligence , 15/11, 1174-1186.

Yalabık, N., Dağıtan, Ü., 1991, Connected Word Recognition Using Neural Networks, NATO ASI Series, Neurocomputing, 297-300.

Yalabık, N., Yarman, F., 1988, A New Approach to Template Selection for Speaker IndependentWord Recognition, NATO ASI Series, 329-334.

Young, S.R., Hauptmann, A.G., 1989, High Level Knowledge Sources in Usable Speech Recognition Systems, Communications of the ACM, 3/21, 183.

Young, J.F., 1971, Information Theory, London Butterworth and Company.

Young, T.Y., 1974, Classification Estimation and Pattern Recognition, American Elsevier Publishing Company.

Zipser, D., Rabin, D., 1986, P3:A Paralel Network Simulation System, Paralel Distributed Processing-1 MIT Press.

Zurada , J.M., 1992, Artificil Neural Systems, West Publishing Company.

EK. 1. Türkçe Sesli İfade Korpüsü

abajur	dede	ip	mal
abla	demir	it	muhtaç
aç	dere	jale	muhtar
acı	dev	kıraç	nasıl
ad	dokuz	kablo	oku
af	dört	kaçık	on
ah	düz	kalın	örtü
altı	efe	kan	otur
an	emek	kaplan	pak
annem	erek	kas	pamuk
av	evet	kav	pas
ay	ezmek	kay	pat
ayar	fırça	kek	perde
az	fırsat	kep	pide
baba	genç	kirpi	pil
bay	güz	kitabın	pot
beş	hah	koca	resim
ben	hasta	kol	ruj
bin	hayır	kop	sıcak
bir	ırmak	kuru	savaş
büro	ısı	lamba	seher
cadı	iki	lav	sekiz
çamur	ilke	lay	sen
dal	ip	loş	simit

su	tere	viraj	zeytin
şarap	top	yedi	ziyan
aşı	törpü	yöntem	zor
taş	üç	yorum	zor
tanı	ülke	yük	zühtü
tav	umut	yüz	
tay	ütü	zahmet	
tay	vah	zavallı	
tek	van	zeki	

EK-2. TMS320C30 ile IEEE arasında kayan ayrımlı gösterim dönüşümü

```
/*****************************************************************************
    TMS320C30 işleyicisi için kayan ayrımlı sayıların gösteriminde
     ieee standart(ından)ına dönüşüm algoritması örneği.
                                                                        TI.COM
*****************************************************************************/
struct c30float
{
    signed     int mantissa:23;
    unsigned int sign:1;
    signed     int exponent:8;
};
struct ieeefloat
{
    unsigned int mantissa:23;
    unsigned int exponent:8;
    unsigned int sign:1;
};

float fmieee(int i)
/*IEEE gösterim biçiminden TMS320C30'nin gösterim biçimine */
{
    union ieee_union
    {
       unsigned int        in;
       struct  ieeefloat  str;
    } ieee;

    union c30_union
    {
       float                 flt;
       struct c30float str;
    } c30;

    ieee.in = i;
    c30.str.mantissa = ieee.str.mantissa;
/* Mantissa değişmeden kalır   */

/* IEEE Exponent 127 eklenir   */
c30.str.exponent = (ieee.str.exponent == 0) ? -128 : ieee.str.exponent - 127;

    c30.str.sign = 0;

/* İşaret ikili pozitif yapılır*/

/* Eğer negatif ise bunun tersi alınır     */
    if (ieee.str.sign) c30.flt = -c30.flt;
    return (c30.flt);
}

unsigned int toieee(float x);
/* TMS320C30'den  IEEE dönüştürme */
{
    union ieee_union
    {
       unsigned int       in;
       struct   ieeefloat str;
    } ieee;
    union c30_union
    {
       float           flt;
       struct c30float str;
    } c30;
    c30.flt = x;
    ieee.str.sign = c30.str.sign;
/* İşaret her iki formatta da aynı kalır */

/* Eğer C30 NEGATIVE ise  tersi alınır (pozitif yapılır) */
    if (c30.str.sign) c30.flt = -c30.flt;
    ieee.str.exponent = (c30.str.exponent == -128) ? 0 : c30.str.exponent + 127;
    ieee.str.mantissa = c30.str.mantissa;
/*  Mantissa aynı kalır     */
    return (ieee.in);
}
```

ÖZGEÇMİŞ

Adı Soyadı : Harun ARTUNER

Doğum Yeri : Ankara

Doğum Yılı : 1960

Medeni Hali : Bekar

Eğitim ve Akademik Durumu:

Lise : 1973-1977 Yenimahalle Mustafa Kemal Lisesi (Ankara)

Lisans : 1977-1982 Ankara Üniversitesi Fen Fakültesi Fizik Mühendisliği Bölümü

Yüksek Lisans : 1985 -1987 Hacettepe Üniversitesi Bilgisayar Bilimleri-Mühendisliği Bölümü

Yabancı Dil : İngilizce

İş Tecrübesi : 1981-1982 Ankara Üniversitesi Fen Fakültesi Fizik Mühendisliği Bölümünde Teknisyen

1982-1984 Mamak Muhabere Okulunda Asteğmen

1984-...Hacettepe Üniversitesi Bilgisayar Bilimleri- Mühendisliği Bölümünde Araştırma Görevlisi

INTERNET ADRESLERİ

ai.toronto.edu pub/xerion Xerion adlı Nöron ağı simülarörü.

archie.funet.fi Kaynak aramaya yönelik makina.

cayuga.cs.rochester.edu pub/simulator RCS adlı Nöron ağı simülatörü.

cheops.cis.ohio.edu pub/neuroprose Nöron ağlarına ilişkin teknik raporlar.

cnuce-arch.cnr.it:/pub/Linuz/X11/xapps khoros ve ptolemy'nin linux için derlenmiş kütükleri.

cnuce-arch.cnr:pub/Linux-local/ptolemy

cochlea.hut.fi pub/lvq_pak, pub/ref, pub/som_pak, pub/utils. Kohonen SOM ve LVQ algoritma ve teknik raporlar, referanslar.

cs.brown.edu Teknik raporlar.

epcc.ed.ac.uk Paralel işlem merkezi, teknik rapor ve programlar.

ftp.cica.indiana.edu Pc ya da Unix üzerinde çalışan programlar.

ftp.cis.upenn.edu:/pub/ldc (130.91.6.8)

ftp.cs.cmu.edu:/project/fgdata/dict

ftp.dartmounth.edu pub/gnuplot Pc ve Workstation üzerinde çalışan çizim programları.

ftp.e20.physik.tu-muenchen.de:/pub/khoros

ftp.khoros.unm.edu khoros'a ilişkin kaynak kütükler. ve FAQ(frequently asked questions)

ftp.mathworks.com WWW: http://www.mathworks.com/

ftp.microsoft.com Microsoft şirketinin makinası.

genesis.cns.caltech.edu (131.215.137.64) GENESIS adlı Nöron ağı simülataörü.

hpcvaaz.cv.hp.com (15.255.72.15) Hewlett Packard 720 Workstation için destek programlar.

ics.uci.edu pub.machine.learning-databases Teknik rapor ve veritabanları.

jaguar.ncsl.nist.gov

me.uta.edu Nöron ağ programları.

phloem.uoregon.edu:/pub/Sun4/lib/phonemes

pprg.eece.unm.edu : Khoros

pprg.eece.unm.edu Khoros adlı sinyal işleme paket programı.

pt.cs.cmu.edu (128.2.254.155) Aspirin adlı Nöron ağı simülatörü.

ptolemy.berkeley.edu:/pub/ptolemy

research.att.com Sinyal işleme programları.

rtfm.mit.edu:/pub/usenet/news.answers/comp-speech-faq/*

sounds.sdsu.edu:/.1/phonemes

sunsite.unc.edu:/pub/multimedia/sun-sounds/phonemes

sunsite.unc.edu:pub/Linux/X11/xapps

svr-ftp.eng.cam.ac.uk pub/comp.speech Sesli ifade tanıma ve oluşturmaya yönelik program ve bildiriler.

svr-ftp.eng.cam.ac.uk:/comp.speech/FAQ-complete

ti.com Texas Instrument'ın yazılım ve bildirileri.

trickle@trmetu MS-Dos ve UNIX programları.

wilma.cs.brown.edu pub/comp.lang.postscript Brown üniversitesi teknik raporları.

wocket.vantage.gte.com:/pub/standard_dictionary

8. SONUÇ, TARTIŞMA VE ÖNERİLER

Türkçe fonem kümeleme sistemi tasarım ve gerçekleştirimi adlı bu tez kapsamında sesli ifade tanımada kullanılan temel yaklaşımlar incelenmiş, ayrışık sözcük tanıma ve sözcük altı ses birimleri tanıma yaklaşımlarından, Türkçe için fonem tanıma tercihi yapılmıştır. Bu tez kapsamında değişik kesimlerde de dile getirildiği üzere, ismin i, e, de ,den halleri, fiil çekimi gibi nedenlerle çok yoğun biçimde sonek alan bir dil olması bu tercihin temel gerekçesini oluşturmaktadır. Zira Türkçe'de kökler sonekler alarak, diğer dillerde ancak birkaç tümce ile ifade edilebilen anlam yoğunluğunda sözcüklere dönüşebilmektedir. Bilindiği üzere ayrışık sözcük tanıma yaklaşımının kullanıldığı sistemlerde sözlük kullanımı temel alınmakta ve tanımaya çalışılan sesli ifadeler (sözcükler) bu sözlük içinde aranmaktadır. Bu yaklaşımla Türkçe'de, sonekler alanlar da dahil olmak üzere olası tüm sözcükler için bir sözlük oluşturma olanağı bulunmadığından bu yaklaşımın genel amaçlı bir Türkçe sesli ifade tanıma için kullanılması olanaksızdır. Bu zorunluluktan dolayı en küçük sesli ifade birimi olarak fonem alınmıştır. Bu yaklaşım Türkçe'nin fonemik bir dil olması itibarıyla sesli ifadelerden yazılı metinlere geçişi de kolaylaştıracak bir yaklaşım olarak düşünülmüştür. Bu düşünce de yapılan tercihi etkileyen bir etmen olmuştur.

Bu bağlamda, önce, ZÜZ türünde tek heceli 47 ve Ü, ÜZ, ÜZZ, ZÜ, ZÜZ, ZÜZZ türünde iki ve daha çok heceden oluşan 139 sözcüklük deneysel Türkçe korpüs hazırlanmıştır. Korpüs içerisinde tek heceli sözcükler yalın çiftler(*minimal pair*) olarak ünlü seslerin, diğerleri ise daha çok ünsüz seslerin özellik vektörlerinin hesaplanması için kullanılmıştır.

Özellik vektörlerinin belirlenmesi için aşağıdaki özellik vektörleri çıkarma yöntemleri incelenmiştir. Bunlar sırasıyla şunlardır:

- *FFT*'ye dayalı sıklık değerlerine dayalı özellik çıkarma yöntemi
- *Mel* skalasında *FFT*'ye dayalı cepstrum katsayılarına dayalı özellik çıkarma yöntemi
- *LPC* parametrelerine dayalı özellik çıkarma yöntemi
- *LPC* -autocorrelation parametrelerine dayalı özellik çıkarma yöntemi
- *LPC* -reflection coefficient parametrelerine dayalı özellik çıkarma yöntemi,

- *LPC* parametrelerine dayalı filtre dizisi yöntemi
- *LPC*'parametrelerine dayalı *Smoothed Group Delay Spectrum* yöntemi

Bu yöntemlerden Kohonen *Learning Vector Quantization* algoritması ile yürütülen fonem kümeleme işlemlerinde en yüksek başarımı *Mel* skalasında *FFT*'ye dayalı *cepstrum* katsayıları yöntemi vermiştir. Bu nedenle bu tez çalışması kapsamında özellik vektörleri bu yöntem kullanılarak hesaplanmıştır. Türkçe *codebook*'un oluşturulmasında bu yöntem kullanılagelmiştir.

Alıştırma aşamasında, Türkçe korpüsten alınan *phon'*lardan yararlanılarak Türkçe iki boyutlu *codebook* Kohonen *Self Organizing Feature Map* algoritması kullanılarak oluşturulmuştur. *Codebook*'un etiketlenmesi aşamasından sonra Türkçe fonemlerin *fonotopic map* (fonem dağılımlı *codebook*)çizimleri elde edilmiştir. Etiketlenmiş *codebook* tanıma aşamasında referans vektör tablosu olarak kullanılması amaçlanmıştır. Bir sesli ifadeye ilişkin özellik vektörleri, Türkçe sesli ifade tanıma laboratuvarı yazılım kütüphanesinde yeralan *gavisual* grafik tabanlı bir yazılım ile kümeleme işlemlerine tabi tutulmuştur. *Codebook*'ta yeralan vektörlerin, kimi fonemler için fonem altı birimlerden oluşması durumunda kümeleme başarımının artmasını sağladığı gözlenmiştir. Bu gerekçeye dayalı olarak *codebook*'ta yaklaşık 300 vektöre yer verilmiştir.

Fonem olup olmadığı tartışma konusu olan *ğ*'nin yer aldığı örnekler özellikle korpüsün dışında tutulmuştur. *ž*'nin başlı başına *phon* olup olmadığı, ya da diğer *phon*'lardan mı oluştuğu ileriki çalışmalarda konu edilecektir.

Bu çalışma sonrasında yapılması gerekli ilk araştırma elde edilen fonotopik mapları kullanarak yürütülecek sistemli tanımanın başarımının ölçülmesine ilişkin olmalıdır.

Bu çalışmada, *Transputer* donanımı ve bunun üzerinde paralel kod üreten derleyicilerin kullanımında deneyim kazanma yoluna gidilmiş ancak uygulama aşamasına tam geçilememiştir. Sözkonusu bu donanım üzerindeki uygulama, nöron ağı yaklaşımını içine alan tanıma aşamasının *Transputer*'lü sistem üzerinde çalıştırılması biçiminde olmalıdır. Bu bağlamda tanıma kesimindeki programların bağımsız çalışabilen kesimleri ayrı işleyiciler üzerinde çalıştırılarak işlem hızının göreli artması sağlanabilecektir.

Bu çalışmanın devamı niteliğinde yapılabilecek bir diğer çalışma da, Türkçe korpüs'un genişletilmesi sözkonusu edilmelidir. Bunun yanı sıra korpüsün birden çok konuşmacıya

ilişkin verileri taşıması ve bu yolla konuşmacıdan bağımsız sesli ifade tanıma sistemleri üzerinde araştırma yapılması gerekmektedir.

Yapılması gerekli diğer bir çalışma da Türkçe sesli ifade tanıma da elde edilen birikimin, dilden bağımsız sesli ifade tanıma çalışma gruplarınca yürütülen projelerle birleştirilmesidir. Bu bağlamda değişik gruplarca üzerinde çalışılan korpuslerin karşılaştırılarak incelenmesi gerekmekte ve bu gruplarla paralel ve ortak çalışma yolları aranmalıdır. Özellikle çalışma gruplarınca *CD-ROM*'lar üzerinde dağıtılan ses veri tabanlarında Türkçenin de yer alması sağlanmalıdır. Bunu için kayıt koşullarının iyileştirilmesi (sessiz oda ve kayıt aletleri gibi) ve belirli bir standarta ulaştırılması gerekmektedir.

Sesli ifade tanıma çalışmalarından elde edilen birikimden, Türkçe yapay sesli ifade oluşturma konusunda yararlanılması sağlanmalıdır. Bu bağlamda elde edilen *codebook*'un sesli ifade oluşturma için denenmesi, hem *codebook*'un sınanması hem de yapay sesli ifade oluşturmaya katkı bakımından yararlı olabilecektir.

Sesli iletişim ortamının, örneğin telefon hattı, kalabalık bir ortam gibi, niteliğine göre araştırma genişletilmelidir.

Bu çalışmanın devamında yapılması gerekli bir diğer araştırma, tanınan sesli ifadelerden yazılı metinlere geçişi sağlayacak anlayışlı ya da uzman sistem kesimine ilişkin olmalıdır. Bu bağlamda bu çalışmanın Türkçeye ilişkin gerçekçi bir sözlük ile kural tabanının hazırlanması gerekli olacaktır., Türkçe üzerinde yürütülen anlamsal ve biçimsel *(semantic and morphologic)* çalışmalarla birlikte düşünülmesi gerekecektir.

KAYNAKLAR

Aksan, D., 1980, Her Yönüyle Dil -Ana Çizgileriyle Dilbilim, Türk Dil Kurumu Yayınları,

Aleksander, I., 1989, Neural Computing Architecture, North Oxford Academic.

Alspector, J., Allen, R.B., 1987, A Neuromorphic VLSI Learning System, Proc. of the 1987 Stanford Conf.: Adv.Res. in VLSI.

Alspector, J., 1988, Research Result in VLSI Implementations of Neural Networks, Conferance Acoustical Soc. of Japan.

Amari, S.I., 1972, Characteristics of Random Nets of Analog Neuron-Like Elements, IEEE Transactions on Systems Man and Cybernetics, 11/2.

Anderson, J.A., 1983, Cognitive and Psychological Computation with Neural Models, IEEE Transactions on Systems Man and Cybernetics, 1/13.

Auger, J.M., 1991, Parallel Implementation on Transputers of Kohonen's Algorithm, NATO ASI Series, Neurocomputing, 215-226.

Barto, A., 1981, Assosiative Search Network: A Reinforcement Learning Associtive Memory, Biological Cybernetics, 4, 201.

Banks, S., 1990, Signal Processing, İmage Processing and, Pattern Rcognitiıon, Prentice Hall.

Baum, E.B., 1986, Internal Representations for Associative Memory, Univercity of California, 86-.

Bengio, Y., Cardin, R., 1989, Programmable Execution of Multi Layered Networks For automatic Speech Recognition., Communications Of The ACM, 2/32, 195.

Blelloch, G., Rosenberg, C.R., 1987, Network Learning on the Connection Machine, Ablex Publishing Corp.

Bose, N.K., 1993, Neural Network Design Using Voronoi Diagrams, IEEE Trans. Neural Networks, 4/5, 778-787.

Brause, R., 1989, Using Neural Networks, Euromicro-Microprocessing and Microprogramming Journal, 8/27, 179.

Buda, O.R., Hart, P.E., 1973, Pattern Classification and Scene Analysis, John Wiley & Sons.

Caianiello, E.R., 1989, Parallel Architectures and Neural Networks, Word Scientific.

Camp, D., 1993, A Users Guide for the Xerion Neural Network Simulator. Version 3.1, University of Toronto.

Carpenter, G.A., Grossberg, S., 1986, Neural Dynamics of Category Learning and Recognition:Attention Memory Consolidation and Amnesia, AAAS Symposium Series, and Amnesia.

Carpenter, G.A., Grossberg, S., 1986, Assosiative Learning adaptive Pat. Recog. and Cooperative-competitive desicion making by NN, SPIE, 634-.

Carpenter, G.A., Grossberg, S., 1988, The ART of Adaptive Pattern Recognition by a Self_Organizing Neural Network, IEEE Computer, 3/21, 77.

Chen, C.H., 1973, Statistical Pattern Recognition, Hayden Book Company,

Chua, L.O., Yang, L., 1988, Cellular Neural Networks:Theory, IEEE Transactions on Circuits and Systems, 1/35, 1257.

Chua, L.O., Yang, L., 1988, Cellular Neural Networks:Applications, IEEE Transactions on Circuits and Systems, 1/35, 1273.

Cohen, M.A., Grossberg, S., 1983, Absolute Stability of Global Pattern Formation & Parallel Memory Storage by Competitive NN, IEEE Transactions on Systems Man and Cybernetics, 1/13.

Cooke, M., Crawford, M., 1993, Vısual Reprasentation of Speech Signals, John Wiley and Sons Ltd.

d'Alessandro, C., Sylvain, J., 1988, Decomposition of The Speech Signal into Short-Time Waveforms Using Spectral Segmentation, IEEE CH 2651-9, 351-354.

Davis, S.B., Mermelstein, P., 1980, Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, IEEE transaction on ASSP, 28/4, 357-367.

Delgutte, B., 1984, Speech Coding in the Auditory Nerve:II.Processing Schemers for Vowel-Like Sounds, Journal of Acoust. Soc. Am. 75, 879-.

Demirezen, M., 1986, Phonemics and Phonology: Theory through Analysis, Bizim Büro Yayınevi.

Demirezen, M., 1987, Articulatory Phonetics and the Principles of Sound Production, Yargı Yayınevi.

Devijer, P.A., Kittler, J., 1977, Pattern Recognition: A Statical Approach, Prentice Hall.

Devijver, P.A., Kittler, J., 1977, PAttern Recognition: A Statical Approach, Prentice Hall.

Doddington, G.R., Schalk, T.B., 1981, Speech Recognition: Turing Theory to Practice, IEEE SPECTRUM, 9/18, 26.

Ellis, E.M., Robinson, A.J., 1993, A Phonetic Tactile Speech Listening System, Cambridge Univ./F-INENG, TR122.

Elman, J.L., Zipser, D., 1987, Learning the Hidden Structure of Speech, Univ. of California at San Diego ICS Report 8701, 2/2.

Ergenç, İ., 1989, Türkiye Türkçesinin Görevsel Sesbilimi.

Erman, L.D., Hayes, R.F., 1987, THE HEARSAY-II Speech-Understanding System:Integrating Knowledge to Resolve Uncertainty, Computing Surveys, 12/2, 214.

Fahlman, S.E., Hinton, G.E., 1987, Connectionist Architectures for Artifical Intelligence, Computer, 1, 100.

Fahlman, S.E., 1988, An Empirical Study of Learning Speed in Back-Propagation Networks, CMU Techincal Report, 5, 88-.

Fant, G., 1959, The Acoustics of Speech, Proceedings of the Third International Congress on Acoustics.

Feldman, J.A., 1982, Dynamic Connections in Neural Networks, Biological Cybernatics, 27-.

Feldman, J.A., Ballard, D.H., 1982, Connectionist Models and Their Properties, Cognitive Science, 6, 205-.

Feldman, J.A., Goddard, N.H., 1988, Computing with Structured Connectionist Networks, Communications of ACM, 2/31, 170-.

Feldman, J.A., Fanty, M.A., 1988, Computing with Structured Neural Networks, IEEE, 3/21, 91-.

Flanagan, J.L., 1972, Voices of Men and Machines, Journal of Acoustical Society of America.

Fukushima, K., Ito, T., 1983, Neocognitron:A Neural Network Model Model for a Mechanism of Visual Pattern Recognition, IEEE Transactions on Systems Man and Cybernetics, 1, 13-.

Fukushima, K., 1988, A Neural Network for Visual Pattern Recognition, IEEE Computer, 3/21.

Gallager, R.G., 1968, Information Theory and Reliable Communication, John Wiley & Sons.

Gallant, S.I., 1988, Connectionist Expert Systems, Communication of ACM, 2/31, 152-.

Genvis, A.S., Morgan, N.H., 1988, Applications of Neural-Network (NN) Signal Processing in Brain Research, IEEE Transactions on Acoustics Speech and Sig.PR, 7, 36-.

Glass, J.R., Zue, V.W., 1988, Multi-Level Acoustic Segmentation of Continuous Speech, IEEE CH2561-9, 429-432.

Goddard, N. H., 1989, The Rochester Connectionist Simulator: User's Manual, University of Rochester Technical Report.

Gold, B., 1987, Hopfield Model Applied to Vowel and Consonant Discrimination, MIT Lincoln Laboratory Tecnical Report, 4.

Gold, B., 1988, A Neural Network for Isolated-Word Recognation, IEEE CH2561-9,44-47.

Graf, H.P., Jakel, L.D., 1986, VLSI Implementation of a Neural Network Memory with Several Hunderts of Neurons, Am. Inst. of Pysics Conferance Proceeding 151, 182-.

Graf, H.P., deVegvar, P., 1987, A CMOS Implementation of a Neural Network Model, Proc. Stanford Conf. Advanced Res. in VLSI, 351-.

Graf, H.P., Jakel, L.D., 1988, VLSI Implementation of a Neural Network Model, IEEE Computer, 3/21, 41-.

Grant, P.M., Sage, J.Q., 1986, A Comparison of Neural Network and Matched Filter Processing for Detecting Lines in Ymages, Neural Networks for Computing, AIP.

Gürgen, F., 1992, Phoneme Recognition Neural Networks, ISCIS 7,569-573.

Habib, M.K., Akel, H., 1988, Logic Gate Formed Neuron Type Processing Element, IEEE ISCAS'88, 491-.

Haffner, P.A., Waibel, A., 1988, Fast Back Propagation Learning Methods for Neural Networks in Speech, ATR Telephone Lab. Tech. Report.

Hammerstrom Dan, D., 1988, A Connectivity Analysis of a Class of Simple Associative Neural Networks, Technical Report No. CS/E-86-009 Oregon Gra.Cent.

Hinton, G.E., Rumelhart, D.E., 1988, Neural Network Architecture for Artificial Intelligence, Tutorial AAAI.

Hirai, Y., 1983, A Model of Human Associative Processor (HASP), IEEE Transactions on Systems Man and Cybernatics, 1, 13.

Hoggle, G., 1992, An FFT-Based Speech Recognition System, Journal of the Franklin Institute, 329/3, 555-562.

Hopfield, J.J., 1982, Neural Networks and Physical Systems with Emergent Collevtive Computational Abilities, Proc. Natl. Acad. Sci. USA, 4/79, 2554-.

Hopfield, J.J., 1984, Neurons with Graded response have Collective Computational Properties Like Those of 2-State Neurons, Procedings National Academy of Science, 5/81, 3088.

Hopfield, J.J., 1984, Neuronswith Graded Response Have Collective Computational PropertiesLike Those of Two State Neurons, Proc. Natl. Acad. Sci. USA, 5, 81-.

Hopfield, J.J., Tank, D.W., 1985, Neural Computation of Decisions in Optimization Problems, Biological Cybernetics, 141-.

Hopfield, J.J., Tank, D.W., 1986, Computing with Neural Circuits: A Model, SCIENCE, 8, 625-.

Hopfield, J.J., Tank, D.W., 1987, Collective computation in neuronlike circuits, Scientific American, 6, 104-.

Horio, Y., Nakamura, S., 1988, Speech Recognition Network with SC Neuron-Like Components, IEEE ISCAS'88, 495-.

Huang, W., Lippmann, R., 1988, A Neural Net Approach to Speech Recognition, IEEE 2561-9, 99-102.

Hubel, D.H., Wiesel, T.N., 1979, Brain Mechanisms of Vision, Scientific American, 8/24, 150.

Hutchinson, J., Koch, C., 1988, Computing Motion Using Analog and Binary Resistive Networks, IEEE Computer, 3/21, 52.

İnce, N., 1992, Digital Speech Processing: Speech Coding, Synthesis and Recognition, Kluwer Academic Publishers.

James, D.A., Young, S.J., 1994, A Fast Lattice-Based Approach to Vocabulary Independent Wordspotting, ICASSP.

James, M.R., 1992, Design of Low-cost, Real-time Simulation Systems for Large Neural Networks, MS thesis at the University of Sydney.

Jones, M., Woodland, P.C., 1993, Using Relative Duration ın Large Vocabulary Speech Recognition, Proc. EuroSpeech'93.

Joseph, W.P., 1993, Signal Modelling Techniques in Speech Recognition, Proceedings of the IEEE, 81/9, 1215-1247.

Judd, J.S., 1987, Complexity of Connectionist Learning with Various Node Functions, Proceeding IEEE Int'l. Conf. on Neural Networks, 6.

Kadirkamanathan, K., Niranjan, M., 1992, Application of an Archıtectureally Dynamic Network for Speech Pattern Classification, Proceedings of the Instutude of Acoustics, 14/6, 343-350.

Kangas, J., 1994, On the Analysis of Pattern Sequences by Self-Organizing Maps, Doktora Tezi, Helsinki Üniversitesi, Finlandiya.

Kitano, H., 1991, An Experimental Speech to Speech Dialog Translation System, IEEE Computer, 24/6.

Klatt, D.H., 1977, Reviev of the ARPA Speech Understanding Project, Journal of Acoustic Soc. American, 12, 1345,

Kohonen, T. , 1982, Self-Organized Formation of Topologically Correct Feature Maps, Biological Cybernetics, 43, 59-69,

Kohonen, T., 1984, Self-Organization and Associative Memory, Springer-Verlag.

Kohonen, T., Mas, I.K., 1984, Phonotopic Maps-Insightful Representation of Phonological Peatures for Speech Representation, Proceedings IEEE Inter. Conf. on Pattern Recognition.

Kohonen, T., 1986, Dynamically Expanding Context with Application to the Correction of Symbol Strings in the Rec. CS., Proceeding 9. Int. Conf. Pattern Recognition IEEE.

Kohonen, T., 1988, The Neural Phonetic Typewriter, IEEE Computer, 3/21, 11.

Kohonen, T., Torkkola, K., 1988, Phonetic Typewriter for Finnish and Japanese, IEEE CH2561-9, 607-610.

Korb, T., Zell, A., 1989, A Declarative Neural Network Description Language, Microprocessing and Microprogramming, 8/27, 181-.

Kosko, B., 1987, Constraction an Associative Memory, BYTE, 137-.

Kosko, B., 1992, Neural Networks for Signal Processing, Prentice Hall.

Kuhn, R., 1992, A Cache-Based Language Model for Speech Recognition, IEEE Transactions on Pattern Recognition and Machine Intelligence, 14/6, 691-692.

Lang, K.J., Withbrock, M.J., 1988, Learning to Tell Two Spirals Apart, Proceedings of the Connectionist Models Summer School.

Lang, K.J., Waibel, A.H., 1990, A Time-Delay Neural Network Architecture for Isalated Word Recognition, Neural Networks, 3, 33-43.

Lea, W.A., 1980, Trends in Speech Recognition, Prentice Hall,.

Lee, K.F., 1989, Automatic Speech Recognition. The Development of the SPHINX System, Kluwer Academic Publishers.

Leighton, R. R., 1992, The Aspirin/MIGRAINES Neural Network Software User's Manual, Mitre Corparation.

Leung, H., Zue, V.W., 1988, Some Phonetic Recognation Experiments Using Artifical Neural Nets, IEEE CH2561-9, 422-425.

Levinson, S.A., Rabiner, L.R., 1983, An Introduction to the Application of the theory of Probabilistic Functions of a Marcov Process ASR., Bell System Technical Journal, 4.

Lewis, F.L., 1986, Optimal Estimation, John Wiley & Sons.

Liebert, P.B., 1967, An Introduction to Optimal Estimation, Addison Wesley Reading Mass.

Linsker, R., Towards an Organizing Princeble for a Layerd Perceptural Network, Netural Information Processing System.

Linsker, R., 1988, Self_Organization in a Perceptual Network, IEEE, 3/21, 105.

Lippmann, R.P., Gold, B., A Comparison of Hamming and Hopfield Neural Nets for Pattern Classification, MIT Lincoln Laboratory Technical Report.

Lippmann, R.P., 1987, An Introduction to Computing with Neural Nets, IEEE ASSP MAGAZINE, 4/4.

Lyon, R.F., Loeb, E.P., 1987, Isolated Digit Recognition experiments with a Cochlear Model, ICASSP-87, 4.

Makhoul, J., Roucos, S., Vector Quantization in Speech Coding, IEEE Proceedings, 73, 1085.

Marks II, R.J., Atlas, L.E., 1988, Generalization in Layerd Classification Neural Networks, IEEE ISCAS'88.

Martin, T., Acoustic Recognition of a Limited Vocabulary in Continuous Speech, Dept. Elec. Eng. Univ. Pennsylvania, 197-.

McClelland, J.L., Rumelhart, D.E., 1986, Paralel Distributed Processing, MIT Press.

McClelland, J.L., Rumelhart, D.E., 1987, Explorations in Parallel Distributed Processing, Cambridge.

McClelland, J.L., Rumelhart, D.E., 1986, Paralel Distributed Processing 1, MIT Press.

McClelland, J.L., Rumelhart, D.E., 1987, Parallel Distributed Processing 2, MIT Press.

McEliece, R.J., Posner, E.C., 1987, The Capacity of the Hopfield Associative Memory, IEEE Transaction on Information Theory, 7/33, 461.

Minsky, M., Papert, S., 1969, Perceptrons:An Introduction ta Computational Geometry, MIT Press.

Miyata, Y., 1991, A Users Guide to PlaNet Version 5.6, University of Colorado.

Moopenn, A., Lambe, J., 1987, Electronic Implementation of Associative Memory Based on Neural Network Models, IEEE Transactions on Systems Man and Cybernetics, 3/17.

Morris, L.R., 1988, A PC-Based Digital Speech Spectrograph, 8/6, 68-85.

Muller, P., Lazzaro, J., 1986, A Machine for Neural computation of Acoustical Patterns with App.To Real-Time Speech Recognition, AIP Conference Proceedings, 151-.

Muveit, H., Weintraub, M., 1988, 1000-Word Speaker-Independent Continuous-Speech Recognition Using Hidden Markov Models, IEEE CH2651-9,115-118.

Mustafa, A., Pslatis, D., 1987, Optical Neural Computers, Scientific American, 3/3, 88-.

Mustafa, A., Jaques, ST., 1985, Information Capacity of the Hopfield Model, IEEE Transaction on Information Theory, 7/31, 461-.

National Semiconductor Corparation, 1985, Switched-Capacitor Filter Handbook.

Nielsen, R.H., 1988, Neurocomputing: Picking the Human Brain, IEEE Spectrum, 3/25, 36.

Nijhuis, J.A.G., Spaanenburg, L., 1989, On Fault Tolarance of Neural Assosiative Memories, IEE Journal E.

Nijhuis, J.A.G., Spaanenburg, L., 1989, Structure and Application of NNSIM: a general_purpose Neural Network SIMulator, Microprocessing and Microprogramming, 8/27, 189.

Norusis, M.J., 1990, SPSS/PC+Statistics 4.0, SPSS Inc.

Oberne, K., Barron, J.J., 1989, Time to Get Fried Up, BYTE, 8/14, 217.

O'Connor, J.D., 1973, Phonetics, Penguin Books.

Oppenheim, A.V., 1989, Discrete-Time Signal Processing, Prentice Hall.

Picone, J., 1990, Continuous Speech Recognition Using Hidden Markov Models, IEEE ASSP Magazine.

Picone, J., 1993, Signal Modeling Techniques in Speech Recognition, Proceedings of the IEEE, 81/9, 1215-1247.

Paik, E., Gungner, D., 1987, UCLA SFINX A Neural Network Simulation ment, IEEE First Int. Conf. on NN Proceeding, 367-.

Papamichalis, P., Simar, R., 1988, The TMS320C30 Floting-Point Digital Signal Processor, IEEE Micro, 8/6, 13-29.

Parker, D.B., 1986, A Comparison of Algorithms for Neuron-Like Cells, AIP Conference Proceddings 151.

Parson, T., 1986, Voice and Speech Processing, McGraw Hill.

Paulos, J.J., Hollis, P.W., 1988, Neural Networks Using Analog Multipliers, ISCAS'88, 499-.

Pelling, S.M., Moore, R.K., 1986, The Multi-Layer Perceptron as a Tool for Speech Pattern Processing Research, Proc.IoA Autumn Conf. on Speech and Hearing.

Petre, P., 1985, Master:Typewriters that take Dictation, FORTUNE, 1.

Plonski, M., Joyce, C., 1993, RCS, GENESIS, and SFINX: Three "Public-Domain" Simulators for Neural Networks.

Rabiner, L.R., February 1989, A Tutorial on Hidden Markov Models and Selected Applications in Speech Processing, Proceedings of the IEEE, 77/2,

Rabiner, L.R., Schafer, R.W., 1975, Digital Repres-entations of Speech Signals, Proceedings of the IEEE.

Rabiner, L.R., Schafer, R.W., 1978, Digital Processing of Speech Signals, Prentice-Hall.

Rabiner, L.R., 1989, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, 2/77, 257,

Rabiner, L.R., 1994, Application of Voice Processing to Telecommunications, Proceedings of the IEEE, 2/82, 199-228.

Reddy, R., Zue, V., 1983, Recognizing Continuous Speech Remains an Elusive goal, IEEE Spectrum, 11, 84.

Regency, H., 1992, International Conferance on Signal Processing Applications and Technology, DSP Associates.

Reid, C.E., 1992, Signal Processing in C, John Wiley and Sons.

Rich, E., Knight, K., 1991, Artifical Intelligence, Mc Graw Hill.

Robinson, T., 1989, Dyanamic Error Propagation Networks, Doktora Tezi, Cambridge Universitesi, İngiltere.

Robinson, T., Fallside, F., 1990, A Recurrent Error Propagation Network Speech Recognition System, Computer Speech and Language Nov.1990, 5/3.

Robinson, T., Fallside, F., 1990, Phoneme Recognition from the TIMIT database using Recurrnt Error Propagation Networks, Cambridge Univ./F-INENG, TR42.

Robinson, T.1991, Several Improvements to a Recurrent Error Propagation Network Phone Recognition System, Cambridge Univ./F-INENG, TR82,

Robinson, T., 1992, A Real-Time Recurrent Error Propagation Network Word Recognition System, IEEE - ICASSP.

Robinson, T., 1992, Recurrent Nets for Phoneme Probability Estimation, Cambridge Univ./F-INENG.

Robinson, T., 1992, Practical Network Design and Implementation, Cambridge Univ./F-INENG.

Robinson, T., 1992, The State Space and "Ideal Input" Representations of Recurrent Networks, ESCA92.

Robinson, T., 1992, Artifical Neural Networks:The Mole-Grips of the Speech Scientist, ESCA92.

Robinson, A.J., Almedia, L., 1993, "A Neural Network Based, Speaker Independent, Large Vocabulary, Continuous Speech Recognition System: The WERNİCE project", Proc. EuroSpeech'93.

Rumelhart, D.E., Hinton, G.E., 1985, Learning Representations by Back-Propagating Errors, NATURE, 1/27, 533.

Rumelhart, D.E., Hinton, G.E., 1986, Parallel Distributed Processing : Vol 1, MIT Cambridge, 318-.

Sage, J.P., Thompson, K., 1986, An Artifical Neural Network Integrated Circuit Based on MNOS/CD Principles, AIP Conferance Proceedings N.151.

Savcı, Y.F., 1994, Sesli İfade Tanıma için Otomatik bir Özellik Çıkarım Sistemi Tasarım ve Gerçekleştirimi, Y.Müh. Tezi, H.Ü. Ankara.

Schalkoff, R.J., 1992, Pattern Recognition: Statistical Structural and Neural Approaches, John Wiley and Sons.

Schalkoff, R., 1992, Pattern Recognition: Statistical Structural and Neural Approaches, John Willey and Sons Inc.

Scroeder, M.R., Hall, J.L., 1974, Model for mechanical to neural transduction in the auditory receptor, Journal of Acoustic Soc. American., 5, 1055.

Seneff, S., 1986, A Computational Model ofr the Peripheral Autitory System:Application to Speecn Recognition Research, ICASSP-86.

Senjnowsiki, T:, Rosenberg, C.R., 1986, NETtalk:A Parallel Network That Learns to Read Aloud, Univ. Techinal Report JHU/EECS-86/01, 1096-.

Shannon, C.E., 1948, A Mathematical Theory of Communication, Bell Systems Technical Journal, 27, 379-623.

Shiffman, S., Wu, A.W., February 1991., Building a Speech Interface to a Medical Diagnostic System, IEEE Expert, 6/1.

Shriver, B.D., 1988, Artificial Neural Systems, IEEE Computer, 3/21, 8.

Sivilotti, M.A., Emerling, M.R., 1986, VLSI architecture for Implementation of Neural Networks, Am. Inst. of Pysics Conferance Proceeding 151, 408-.

Sivilotti, M.A.,Mahowald,M.A.,1987,Real Time Visual Computations Using Analog CMOS Processing Arrays, Advanced Research in VLSI, MIT Press, 295-.

Stanley, H.L., 1984, MultipleValued Logic Its Status and Its Future, IEEE Transactions on Computer, 1/33, 1160-.

Sutton, R., Barto, A., 1981, An Adaptive Network That Constructs and Uses an Internal model of Its World, Cognition and Brain Theory, 4/3, 217-.

Tank, W., Hopfield, J., 1986, Simple \Neural\ Optimization Networks: An A/D Converter, Signal Devision Circuit and a Li IEEE Transactions on Circuits and Systems, 5, 33-.

Tank, W., Hopfield, J., 1989, Collective Computation in Neuronlike Circuits, Scientific American, 8, 62-.

Tazelaar, M., 1989, Neural Networks, BYTE, 8/14, 214-.

Tenorio, M.F., Huges, C.S., 1987, Real Time Noisy Image Segmentation Using an Artifical Neural Network Model, Proceeding of the IEEE 1.Int. Conf.on NN, 4, 357.

Tenorio, M.F., Tom, M.D., 1989, Adaptive Networks as a Model for Human Speech Development, Purdue University, TR-EE 89-54.

Terry, M. Renalds, S., 1988, A Connectionist Approach To Speech Recognition Using Peripheral Auditory Modelling, IEEE CHI2561-9, 699-702.

Texas Instruments, 1991, Digital Signal Processing Applications with the TMS320C30 Evaluation Module, Texas Instruments.

Texas Instruments, 1990, TMS320C30 Evaluation Module Technical Referance, Texas Instruments.

Texas Instruments, 1992, TMS320C3x Users Guide, Texas Instruments.

Torkkola, K., 1988, AUtomatic Alignment of Speech with Phonetic Transcriptions in Real Time, IEEE CH2561-9, 611-614.

Touretzky, D.S., Hinton, G.E., 1985, Symbols Among Neurons:Details of Connectionist Inference Architecture, Proceeding IJCAI Los Angeles, 238-.

Touretzky, D.S., Pomerleau, D., 1989, What's Hidden in the Hidden Layers?, BYTE, 8/14, 227-.

Touretzky, D.S., Elman, J.L., 1990, Connectionist Models. Proceedings of the 1990 Summer School, Morgan Kaufmann Publishers.

Van Der Kam, J., 1986, A Digital Decimating Filter for Analog-to-Digital Conversion of Hi-Fi Audio Signals, Phillips Technical Review, 42, 6/7.

Vemuri, E., 1988, Artifical Neural Networks:An Introduction, IEEE, 1.

Verhaeghe, B., 1992, Toward Continuous-Speech Recognition, BYTE , 17/4, 158-158.

Waibel, A., Hanazawa, T., 1988, Phoneme Recognition: Neural Networks vs. Hidden Markov Models, IEEE CH2561-9, 107-110.

Waibel, A., Hanazawa, T., 1989, Phoneme Recognition Using Time-Delay Neural Networks, IEEE Transactions on Acoustics Speech&Signal Proc, 3, 37-.

Waibel, A., Hampshire, J., 1989, Building Blocks for Speech, BYTE, 8/14, 235-.

Wallance, D., 1986, Memory and Learning in a Class of Neural Models, Proceedings of the Workshop on Lattice Gauge Theor.

Waltz., D.L., 1987, Applications of the Connection Machine, IEEE Computer, 1, 85.

Wan Der Enden, W.M., 1989, Discrete-time Signal Analysis: an Introduction, Prentice-Hall.

Watrous, R., 1988, Speech Recognition Using Connectionist Networks, Dept. of Comp. and Information Science, Uni. Penn.

White, M.W., Holdaway, R.M., 1990, New Strategies for Improving Speech Enhencement, Int. J. Biomed. Comp., 25/2-3, 101.

Widrow, B., Winter, R., 1988, Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition, IEEE Computer, 32/21, 25.

Widrow, B., Winter, R., 1988, Layered Neural Nets for Pattern Recognition, IEEE Transaction on ASSP, 36/7, 1109-1118.

Wu, J., Chan, C., 1993, Isolated Word Recognition by Neural Network Models with Cross-Correlation Coefficients for Speech Dynamics, IEEE Trans. Pattern Analysis and Machine Intelligence , 15/11, 1174-1186.

Yalabık, N., Dağıtan, Ü., 1991, Connected Word Recognition Using Neural Networks, NATO ASI Series, Neurocomputing, 297-300.

Yalabık, N., Yarman, F., 1988, A New Approach to Template Selection for Speaker IndependentWord Recognition, NATO ASI Series, 329-334.

Young, S.R., Hauptmann, A.G., 1989, High Level Knowledge Sources in Usable Speech Recognition Systems, Communications of the ACM, 3/21, 183.

Young, J.F., 1971, Information Theory, London Butterworth and Company.

Young, T.Y., 1974, Classification Estimation and Pattern Recognition, American Elsevier Publishing Company.

Zipser, D., Rabin, D., 1986, P3:A Paralel Network Simulation System, Paralel Distributed Processing-1 MIT Press.

Zurada , J.M., 1992, Artificil Neural Systems, West Publishing Company.

EK. 1. Türkçe Sesli İfade Korpüsü

abajur	dede	ip	mal
abla	demir	it	muhtaç
aç	dere	jale	muhtar
acı	dev	kıraç	nasıl
ad	dokuz	kablo	oku
af	dört	kaçık	on
ah	düz	kalın	örtü
altı	efe	kan	otur
an	emek	kaplan	pak
annem	erek	kas	pamuk
av	evet	kav	pas
ay	ezmek	kay	pat
ayar	fırça	kek	perde
az	fırsat	kep	pide
baba	genç	kirpi	pil
bay	güz	kitabın	pot
beş	hah	koca	resim
ben	hasta	kol	ruj
bin	hayır	kop	sıcak
bir	ırmak	kuru	savaş
büro	ısı	lamba	seher
cadı	iki	lav	sekiz
çamur	ilke	lay	sen
dal	ip	loş	simit

su	tere	viraj	zeytin
şarap	top	yedi	ziyan
aşı	törpü	yöntem	zor
taş	üç	yorum	zor
tanı	ülke	yük	zühtü
tav	umut	yüz	
tay	ütü	zahmet	
tay	vah	zavallı	
tek	van	zeki	

EK-2. TMS320C30 ile IEEE arasında kayan ayrımlı gösterim dönüşümü

```
/*************************************************************************
   TMS320C30 işleyicisi için kayan ayrımlı sayıların gösteriminde
    ieee standart(ından)ına dönüşüm algoritması örneği.
                                                    TI.COM
*************************************************************************/
struct c30float
{
   signed     int mantissa:23;
   unsigned int sign:1;
   signed     int exponent:8;
};
struct ieeefloat
{
   unsigned int mantissa:23;
   unsigned int exponent:8;
   unsigned int sign:1;
};

float fmieee(int i)                    /*IEEE gösterim biçiminden TMS320C30'nin gösterim biçimine */
{
   union ieee_union
   {
            unsigned int        in;
            struct  ieeefloat  str;
   } ieee;

   union c30_union
   {
            float               flt;
```

```
        struct c30float str;
    } c30;

    ieee.in = i;
    c30.str.mantissa = ieee.str.mantissa;                    /* Mantissa değişmeden kalır   */
                                                             /* IEEE Exponent 127 eklenir   */
    c30.str.exponent = (ieee.str.exponent == 0) ? -128 : ieee.str.exponent - 127;

    c30.str.sign = 0;                                        /* İşaret ikili pozitif yapılır*/
                                                             /* Eğer negatif ise bunun tersi alınır     */
    if (ieee.str.sign) c30.flt = -c30.flt;
    return (c30.flt);
}

unsigned int toieee(float x)                                 /* TMS320C30'den  IEEE dönüştürme */
{
    union ieee_union
    {
        unsigned int       in;
        struct   ieeefloat str;
    } ieee;
    union c30_union
    {
        float          flt;
        struct c30float str;
    } c30;
    c30.flt = x;
    ieee.str.sign = c30.str.sign;                            /* İşaret her iki formatta da aynı kalır */
                                                             /* Eğer C30 NEGATIVE ise  tersi alınır (pozitif yapılır) */
    if (c30.str.sign) c30.flt = -c30.flt;
    ieee.str.exponent = (c30.str.exponent == -128) ? 0 : c30.str.exponent + 127;
    ieee.str.mantissa = c30.str.mantissa;                    /*  Mantissa aynı kalır     */
```

```
    return (ieee.in);
}
```

ÖZGEÇMİŞ

Adı Soyadı : Harun ARTUNER

Doğum Yeri : Ankara

Doğum Yılı : 1960

Medeni Hali : Bekar

Eğitim ve Akademik Durumu:

Lise : 1973-1977 Yenimahalle Mustafa Kemal Lisesi (Ankara)

Lisans : 1977-1982 Ankara Üniversitesi Fen Fakültesi Fizik Mühendisliği Bölümü

Yüksek Lisans : 1985 -1987 Hacettepe Üniversitesi Bilgisayar Bilimleri-Mühendisliği Bölümü

Yabancı Dil : İngilizce

İş Tecrübesi : 1981-1982 Ankara Üniversitesi Fen Fakültesi Fizik Mühendisliği Bölümünde Teknisyen

1982-1984 Mamak Muhabere Okulunda Asteğmen

1984-... Hacettepe Üniversitesi Bilgisayar Bilimleri-Mühendisliği Bölümünde Araştırma Görevlisi

INTERNET ADRESLERİ

ai.toronto.edu pub/xerion Xerion adlı Nöron ağı simülarörü.

archie.funet.fi Kaynak aramaya yönelik makina.

cayuga.cs.rochester.edu pub/simulator RCS adlı Nöron ağı simülatörü.

cheops.cis.ohio.edu pub/neuroprose Nöron ağlarına ilişkin teknik raporlar.

cnuce-arch.cnr.it:/pub/Linuz/X11/xapps khoros ve ptolemy'nin linux için derlenmiş kütükleri.

cnuce-arch.cnr:pub/Linux-local/ptolemy

cochlea.hut.fi pub/lvq_pak, pub/ref, pub/som_pak, pub/utils. Kohonen SOM ve LVQ algoritma ve teknik raporlar, referanslar.

cs.brown.edu Teknik raporlar.

epcc.ed.ac.uk Paralel işlem merkezi, teknik rapor ve programlar.

ftp.cica.indiana.edu Pc ya da Unix üzerinde çalışan programlar.

ftp.cis.upenn.edu:/pub/ldc (130.91.6.8)

ftp.cs.cmu.edu:/project/fgdata/dict

ftp.dartmounth.edu pub/gnuplot Pc ve Workstation üzerinde çalışan çizim programları.

ftp.e20.physik.tu-muenchen.de:/pub/khoros

ftp.khoros.unm.edu khoros'a ilişkin kaynak kütükler. ve FAQ(frequently asked questions)

ftp.mathworks.com WWW: http://www.mathworks.com/

ftp.microsoft.com Microsoft şirketinin makinası.

genesis.cns.caltech.edu (131.215.137.64) GENESIS adlı Nöron ağı simülataörü.

hpcvaaz.cv.hp.com (15.255.72.15) Hewlett Packard 720 Workstation için destek programlar.

ics.uci.edu pub.machine.learning-databases Teknik rapor ve veritabanları.

jaguar.ncsl.nist.gov

me.uta.edu Nöron ağ programları.

phloem.uoregon.edu:/pub/Sun4/lib/phonemes

pprg.eece.unm.edu : Khoros

pprg.eece.unm.edu Khoros adlı sinyal işleme paket programı.

pt.cs.cmu.edu (128.2.254.155) Aspirin adlı Nöron ağı simülatörü.

ptolemy.berkeley.edu:/pub/ptolemy

research.att.com Sinyal işleme programları.

rtfm.mit.edu:/pub/usenet/news.answers/comp-speech-faq/*

sounds.sdsu.edu:/.1/phonemes

sunsite.unc.edu:/pub/multimedia/sun-sounds/phonemes

sunsite.unc.edu:pub/Linux/X11/xapps

svr-ftp.eng.cam.ac.uk pub/comp.speech Sesli ifade tanıma ve oluşturmaya yönelik program ve bildiriler.

svr-ftp.eng.cam.ac.uk:/comp.speech/FAQ-complete

ti.com Texas Instrument'ın yazılım ve bildirileri.

trickle@trmetu MS-Dos ve UNIX programları.

wilma.cs.brown.edu pub/comp.lang.postscript Brown üniversitesi teknik raporları.

wocket.vantage.gte.com:/pub/standard_dictionary