

79263

İSTANBUL TEKNİK ÜNİVERSİTESİ \* FEN BİLİMLERİ ENSTİTÜSÜ

SOLUNUM SESLERİNİN ANALİZİ

VE SINIFLANDIRILMASI

YÜKSEK LİSANS TEZİ

Müh. Tanju ENGİN

Tezin Enstitüye Verildiği Tarih : 5 Eylül 1991

Tezin Savunulduğu Tarih : 12 Eylül 1991

Tez Danışmanı : Yard.Doç. Dr.H.Ümit AYGÖLÜ

Diğer Jüri Üyeleri : Prof.Dr.Ertuğrul YAZGAN  
Doç.Dr. Ahmet H. KAYRAN

**T. C.**  
**Yükseköğretim Kurulu**  
**Dokümantasyon Merkezi**

EYLÜL 1991

## ÖNSÖZ

Bu konuda çalışmama olanak veren sayın hocam Prof. Dr. Bülent Sankur'a, tez konusunu önererek bana yepyeni bir çalışma alanı yaratan sayın hocam Yrd.Doç.Dr. Yasemin P. Kahya'ya, hastahane kayıtları sırasında bilgi birikimlerini bizlerle paylaşmaktan çekinmeyerek bizlere destek olan Prof.Dr. Rauf Saygın'a, Prof.Dr. Olcay Neyzi'ye ve Doç.Dr. Sema Umut'a, gerek donanımda, gerekse yazılımda bana yardımcı olan proje arkadaşım Çağatay Güler'e, bana yazıcısını vererek tezin bilgisayar çıktısının alınması aşamasında yardımcı olan Kerem Kaner'e ve her türlü yardımları için Yrd.Doç.Dr. H. Ümit Aygölü'ne teşekkür ederim.

## İÇİNDEKİLER

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| ÖNSÖZ.....                                                            | ii |
| ÖZET.....                                                             | v  |
| SUMMARY.....                                                          | vi |
| BÖLÜM 1. GİRİŞ.....                                                   | 1  |
| BÖLÜM 2. SOLUNUM FİZYOLOJİSİ VE SOLUNUM SESLERİ.....                  | 7  |
| 2.1. Solunum Sistemi ve Akciğerler.....                               | 7  |
| 2.2. Solunum Sesleri.....                                             | 9  |
| 2.2.1. Normal Soluk Sesleri.....                                      | 11 |
| 2.2.2. Islık Sesleri.....                                             | 14 |
| 2.2.3. Çıtırtı Sesleri.....                                           | 16 |
| 2.3. Konuşma Sesleri.....                                             | 18 |
| 2.4. Akciğer ve Solunum Yolları Hastalıkları....                      | 19 |
| BÖLÜM 3. İŞARET İŞLEME VE ÖRÜNTÜ TANIMA YÖNTEMLERİ....                | 22 |
| 3.1. Zaman Serilerinin Doğrusal Olarak<br>Modellenmesi.....           | 22 |
| 3.2. AR Katsayılarının En Küçük Kareler<br>Yöntemi İle Kestirimi..... | 26 |
| 3.2.1. Doğrudan En Küçük Kareler Kestirimi....                        | 28 |
| 3.2.1.1. Kovaryans ve Özilinti Yöntemi....                            | 30 |
| 3.2.1.2. Kısımsal İlinti Yöntemi.....                                 | 32 |
| 3.2.2. Özyineli En Küçük Kareler Kestirimi....                        | 33 |
| 3.2.3. Kafes Yöntemi.....                                             | 35 |
| 3.3. Örüntü Tanımanın Temelleri.....                                  | 36 |
| 3.3.1. k-En Yakın Komşu Yöntemi.....                                  | 39 |
| 3.3.2. En Küçük Uzaklık Sınıflandırıcıları....                        | 42 |
| BÖLÜM 4. SOLUNUM SESLERİNİN ANALİZİ VE<br>SINIFLANDIRILMASI .....     | 49 |
| 4.1. Ölçüm Düzeni.....                                                | 49 |
| 4.2. Analiz Ve Sınıflandırma Yöntemi.....                             | 50 |
| 4.3. Deneysel Sonuçlar.....                                           | 58 |

|                                   |     |
|-----------------------------------|-----|
| SONUÇLAR VE ÖNERİLER.....         | 71  |
| KAYNAKLAR.....                    | 73  |
| EK A. DONANIM.....                | 76  |
| EK B. BİLGİSAYAR PROGRAMLARI..... | 79  |
| ÖZGEÇMİŞ.....                     | 106 |



## ÖZET

Akciğer hastalıklarının tanısında en çok kullanılan yöntemler, dinleme, ııınçekim ve işlev testi yöntemleridir. Bu yöntemler arasında ucuzluğu, kolaylığı ve herhangi bir cerrahi müdahale gerektirmemesi açısından en ilgi çeken dinleme yöntemidir. Ancak soluk seslerinin dinlenmesi kolay bir yöntem olmasına karşın bir çok doktor tarafından tanı için yeterli sayılmamaktadır. Bunun nedenlerinin başında hastalıklı akciğerler için tanımlanan seslerin nesnel değil öznel olması gelmektedir. Örneğin, astımda duyulan müziksel sesler ısıık, akciğer yangısında duyulan ani patlama sesleri çıtırtı diye tanımlanmaktadır [1].

Öznel tanımlamaların ortaya çıkardığı problemleri ortadan kaldırmak için en akılcı yaklaşım , hem normal hem de hastalıklı seslerin parametrik gösterimlerini bulmak, daha sonra bu parametreleri sesleri sınıflandırma ve hastalığı tanıma amacıyla kullanmaktır.

Bu çalışmanın amacı sık görülen akciğer ve solunum yolları hastalıklarında duyulan solunum seslerinin bilgisayar yardımıyla analizi ve sınıflandırılmasıdır. Bu amaçla göğüs kafesinden duyulan solunum sesleri bilgisayara kaydedilmiş, daha sonra kaydedilen verilere özbağılanımlı (AR) modelleme uygulanmıştır. Sesleri sınıflandırmak için Itakura uzaklık ölçüsü kullanılmıştır. Yapılan deneyler önerilen sınıflandırıcının, yüzde 85'lik bir başarı oranı ile zatürreeli sesleri sağlıklı seslerden ayırabildiğini göstermiştir.

## SUMMARY

### ANALYSIS AND CLASSIFICATION OF RESPIRATORY SOUNDS

Auscultation, function testing and roentgenography are the three widely used methods in the identification and diagnosis of lung disease and respiratory disorder. Among these, auscultation is the most attractive method in that it is cheap, simple and non-invasive. The basis of the method is to stimulate the chest by an acoustical input and then to auscultate the sounds on the chest wall and trachea. The sound source itself could be generated in the chest through breathing at and above a certain flow rate, through percussion, or with speech utterances. The pattern variations occurring in these sounds provide to the physician important information about the pathological situations, i.e. lung lesions, obstruction and collapse of the bronchial paths, the status of the alveoli etc. However auscultation is regarded by many physicians of low diagnostic value. The reason of this approach to auscultation is the subjective definitions of adventitious lung sounds. For example, the musical sounds heard in asthma are called wheezes, or suddenly bursting sounds heard in pulmonary fibrosis are called crackles. An experiment conducted among the qualified physicians [2] has revealed significant variability between the observers, such that the diagnostic opinions were falling midway between chance and full agreement on almost all of the symptoms. The difficulty comes from the uncertainty in relating and transferring subjective qualifications and sound assessments, and the confusing terminology [3].

In order to avoid the problems of subjective definitions that change from one physician to another, a reasonable approach is to use a parametric representation of both normal and abnormal sounds and then, use these parameters to compare and classify lung sounds for diagnostic purposes. This method, though seems easy to implement in theory, is difficult to achieve in practical applications, as these waveforms and their spectral content may depend upon various conditions, such as flow rate, regional volume distribution, surrounding tissue, position of the patient, the stage of the illness.

Respiratory sounds are divided roughly into two classes, breath sounds and adventitious sounds. Breath sounds are low-pitched, non-musical rustlings that are synchronous with the movement of air in and out of the lungs, and they arise as the flow of air changes from laminar to turbulent above a certain flow rate. Normal breath sounds themselves can be classified into four types called vesicular, bronchial, broncho-vesicular and tracheal [4]. Their amplitudes, however, have been observed to display significant variations among sites of measurement in any one patient, as well as across patients. The above mentioned breath sounds are considered normal if heard over certain areas of the throat, and abnormal otherwise. There are, however, several types of breath sounds which, when present, always indicate abnormality. For example, the cogwheel breath sound which contains several short pauses during the inspiratory phase is thought to be caused by nonuniform inflation of normal alveoli. Asthmatic breath sound is characterized by its prolonged expiratory phase and by its short inspiratory phase. It is the result of obstruction of airways and is heard primarily in patients with asthma, but also in patients with bronchitis and emphysema.

The adventitious sounds, on the other hand, are much more varied, though often a more direct association exists between these sounds and their physiological causes. These sounds have been described by various adjectives such as wet, crepitant, crackling, stridor, tinkling, rharale, fricative, amphoric, cavernous, etc. For example consecutive fine crackles at the beginning and terminal stages of pneumonia, with the trickled exuded matter causing alveolar walls to stick together in expiration, while they snap out in inspiration, causing crepitation. Wet rales or subcrepitant sounds, on the other hand, occur both in inspiration and expiration. Typical occurrences of wet rales are with acute, chronic bronchitis, lung abscess, diffuse fibrosis, bronchiectasis, diffuse interstitial lung illnesses. Fricative sounds occur if serous membranes have lost their lubricant fluid or if there exists fibrous exudate. Stridors are whistling type of sounds that occur in laryngeal oedema or acute laryngitis, chronic mediastinitis, aortic aneurysm. Wheezing sounds seen in bronchial asthma and chronic, acute bronchitis are due to the constriction of bronchial lumens which narrows the air pathways due to muscle spasms or mucosa swelling.

The goal of this work is the early detection of pneumonia and eventually other lung diseases with the

help of signal processing and computerization methods. For this purpose, a data acquisition and measurement system, shown in Figure 1, was used [8]. Respiratory sounds were preprocessed by a bandpass filter in the 80–2000 Hz band. The lower cutoff frequency was set to eliminate the heart sounds and muscle friction noise. A 4th order Butterworth highpass filter and an 8th order Bessel lowpass filter were designed, both with almost linear phase characteristics. A preamplifier with a gain of 45 dB and flat frequency response was implemented before the analog-to-digital (A/D) converter. The signal was digitized with 12 bit accuracy (MacAdiosII Jr) at 5 kHz sampling rate and inputted into the computer (Apple Macintosh II-cx). The quality of the sound was monitored before conversion with headphones (Philips SBC 484). In order to adjust the flow rate and synchronize on the inspiration–expiration cycles, the flow signal was also recorded by a Fleisch type flowmeter (Gould Godart No.2) [5].

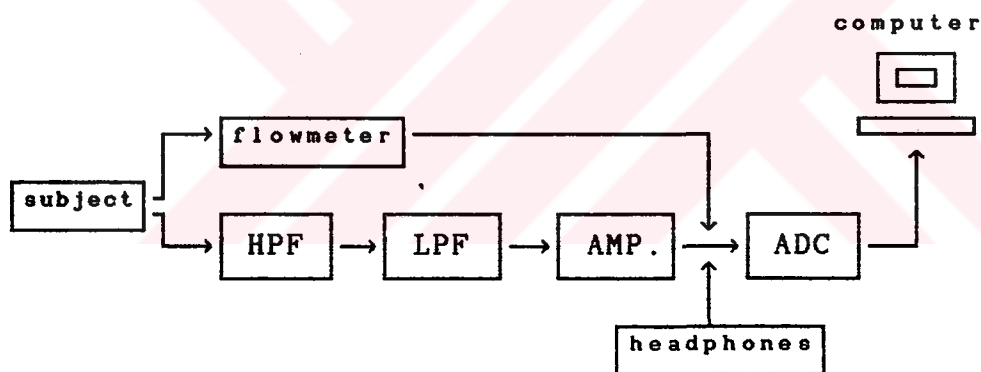


Figure 1. Measurement set up for respiratory sound monitoring and recording.

In order to avoid friction noise, and to isolate the ambient noise, the microphones were encapsulated within manufactured plastic casings, and then attached to the chest wall and trachea with an adhesive tape. This type of air coupling also makes the measurement less dependent upon the static force applied [6]. The microphones were Panasonic P9930 with a bandwidth of 20–18000 Hz. The measurement sites were right basilar, left basilar, and tracheal. From each of the three sites recordings of 12.8 s duration were made, a period covering 4–5 respiration cycles. Flow signal was also recorded to synchronize on the inspiratory and

expiratory cycles.

The recorded respiratory data was divided into frames of 256 samples and weighted with a Hamming window. Successive frames had 64 samples in common or overlapped by 25%. This produced a smoothing effect and prevented spurious leaps in the estimated parameters. Assuming that classes of respiratory sounds can be modelled as white-noise driven all-pole systems, an autoregressive (AR) model was built for each class of signal as:

$$s_f^{(i)}(n) = \sum_{k=1}^p a_{k,f}^{(i)} s_f^{(i)}(n-k) + e_f^{(i)}(n) ; n = 1, \dots, N$$

(1)

where  $s_f^{(i)}(n)$  was the respiratory sound signal observed at time  $f n$ ,  $a_{k,f}^{(i)}$  was the  $k$ 'th LPC coefficient,  $p$  was the model order,  $N_f$  was the frame length and  $e_f^{(i)}(n)$  was the one-step prediction error. The superscript  $i$  denoted the case of the subject and  $f$  denoted the flow cycle. The major concern in selecting the model order  $p$  was the class separability, an order of  $p = 8$  seemed adequate. As inspiration and expiration cycles display different spectral characteristics, separate AR models were applied to each cycle. Durbin's algorithm was used for the calculation of AR coefficients. The discrimination between cycles was achieved using flowmeter data. For this purpose, a modified zero crossing method was used. The calculated AR coefficients from data sets were labeled through their concerning flow cycles. Classification was based on the  $k$ -nearest neighbor algorithm employing the  $p$  AR coefficients as the feature vectors and the Itakura (I) distance measure [7] between the classes, which is specified as:

$$d_{m,1}^{(i)} = \log \left( \frac{\sum_{n=1}^N \underline{b}_{i,1}^T \underline{s}_n \underline{s}_n^T \underline{b}_{i,1}}{\sum_{n=1}^N \underline{a}_m^T \underline{s}_n \underline{s}_n^T \underline{a}_m} \right) \quad (2)$$

where  $d^{(i)}$  is the I distance of the  $m$ 'th frame to the  $l$ 'th reference vector  $b_l^{(i)}$  of the  $i$ 'th class,  $s$  is the observation sequence of  $p$  samples  $s_n$  and  $a_l^{(i)}$  is the LPC vector identified from the test frame. The I measure, i.e., a measure for spectral mismatch, can also be obtained with the ratio of the variance of the prediction errors, i.e.,

$$d_{m,l}^{(i)} = \log \left( \frac{\sum_{n=1}^N e_{i,l}^2(n)}{\sum_{n=1}^N e_m^2(n)} \right) \quad (3)$$

where the denominator is the total prediction error of the test frame and the numerator is the total prediction error when the test frame is filtered through the  $l$ 'th coefficient set in the reference library. There is no need of an extra calculation for the denominator as it is already calculated in the Durbin's algorithm.

$k$ -nearest neighbor algorithm requires the comparison of the test sample to each of the reference points, so reducing the size of the reference library increases the computational speed considerably as well as reducing the storage space requirements. In order to reduce the library size without compromising the information content, the condensed nearest neighbor algorithm was employed. This algorithm works on the principle of classifying a frame with the current reference library and adding the coefficients to the library if the classification is incorrect and discarding the coefficients otherwise. The condensing routine starts with an initial reference library containing  $k+1$  reference AR coefficient sets, chosen randomly from each pattern class and inspiration-expiration cycle.  $k$  was chosen to be 1, 3 and 5. After the condensing routine, the test frames were classified using  $k$ -nearest neighbor algorithm. For each pattern class and flow phase, the lowest  $k$  distances were determined. The sum of these  $k$  distances represented the overall distance of the test frame to the specific pattern class. These overall distances computed for each class were compared and the smallest distance was determined. The test frame was classified as belonging to the pattern class which generated the smallest distance. Two pattern classes, healthy and pneumonic

breath cycles, were used in the classification process. The performance of the classifier was calculated using leave-one-out method, which provides good estimates of the probability of error in case of small sample populations. The experiments showed that suggested classifier could distinguish the pneumonic breath cycles providing a percentage of correct classifications of 85 %.



## BÖLÜM 1. GİRİŞ

Son yıllarda, süregelen tıkanmalı akciğer hastalıkları ve solunum yolu rahatsızlıkları, modern hayatın yol açtığı hava kirliliği yüzünden oldukça önemli hastalıklar haline gelmişlerdir. Bu tür hastalıkların geç ya da yanlış belirlenmesi sorunların daha da büyümesine yol açmaktadır.

Akciğer hastalıklarının ve solunum yolu bozukluklarının belirlenmesinde en çok kullanılan üç yöntem dinleme, işlev testi ve ışınçekim yöntemleridir. Ancak ciğerlerdeki hava yollarını etkileyen hastalıkların ışınçekim yardımıyla kolaylıkla belirlenememesi, daha ucuz, kolay ve cerrahi müdahale gerektirmeyen dinleme yöntemine olan ilgiyi arttırmıştır.

Dinleme yönteminin temeli göğüs kafesinin akustik olarak uyarılması ve sonuçta oluşan sesleri yine göğüs kafesinin çeşitli yerlerinden dinlenmesidir. Akciğerde bir hastalık olduğu durumlarda göğüs kafesinin aktarım işlevi normal bir akciğere göre değişeceğinden duyulan sesler de normal duruma göre farklılık gösterirler. Seslerdeki bu örüntü değişimleri doktora havayollarının durumu, akciğer örsentileri, bronşiyel yollardaki tıkanıklıklar, akciğer keseciklerinin durumu hakkında bilgi vermektedir. Uyarma işaretleri, soluk sesleri, vuruş sesleri ve konuşma sesleri olarak kabaca üçe ayrılabilir [3]. Soluk sesleri, normal ve hastalıklı durumlarda hava akışının havayollarında oluşturduğu akustik işaretlerdir. Vuruş sesleri, göğüs kafesine

parmakla vurmak yoluyla oluřan seslerdir. Konuřma sesleri ise ses kasları ve ses yolu tarafından oluřturulan akustik seslerdir. Ağızdan yayılan bu sesler aynı zamanda havayolları ve akciğerler yoluyla göğüs duvarından da duyulurlar. Konuřma sesleri de kendi aralarında normal sesler, bronkofoni, fısıltı ve egofoni olarak ayrılabilirler. Göğüs kafesi alçakgeçiren bir süzgeç gibi davrandığından normal bir kiřide yüksek frekanslı sesler duyulmaz, ancak hava keseciklerinin sıvıyla dolması ya da göğüs zarının yangılanması gibi hastalıklı durumlarda yüksek frekansların iletimi artar. Uyarı iřaretinin konuřma olduđu durumlarda, konuřma sesleri göğüs kafesinden daha az bozulmuř olarak duyulabilir.

Dinleme amacıyla yaygın olarak stetoskop kullanılmaktadır. Fakat bu yolla yapılan belirlemeler ve sınıflandırmalar, gerek stetoskopun bant geniřliđinin yeterli olmaması, gerekse duyulan seslerin öznel biçimde yorumlanması yüzünden her zaman dođru olmamaktadır. Soluk seslerini tanımlayan terminolojinin karmařıklığı da yanlış yorum ve tanırlara bařka bir neden olarak gösterilebilir. Örneđin, astımlı bir hastanın soluması sırasında çıkardığı ses ısıklık benzeri bir ses olarak nitelendirilmekte, göğüs zarı yangılanmasında (zatül-cenp) duyulan sesler iki elin birbirine sürtünmesi ile çıkan seslere benzetilmektedir. Doktorlar arasında yapılan bir testte solunum hastalıklarının belirlenmesinde büyük görüş ayrılıkları olduđu gözlenmiřtir [2]. Stetoskopa alternatif olarak kullanılan elektronik stetoskoplar ve mikrofonların özellikleri mekanik stetoskoptan daha iyi olmasına rađmen belirleme yine doktorun öznel yargılarına ve nitelermelerine bađlıdır. Çok kullanılan diđer bir yöntem de akciğer iřlev testi ya da spirometre yöntemidir. Bu test akciğerin hacmi ve

hava yollarının direnci gibi bilgiler vermektedir. Spirometrede hastanın da işbirliğine gereksinim duyulduğundan bebekler ve çok küçük yaştaki çocuklar için uygun değildir. Işınçekim yöntemi de çok kullanılan bir yöntem olmasına karşın hem pahalıdır, hem de tıkanıklığa bağlı hastalıklar bu yöntemle kolaylıkla belirlenemeyebilir.

Son yıllarda solunum fizyolojisi üzerine yapılan çalışmalar dinleme yönteminin daha iyi anlaşılmasını sağlamıştır. 1960'ların sonlarında Forgacs' ın [3] solunum sisteminin bilinen anatomisi ve fizyolojisi ile ilgili bazı açıklamalar getirmesi ve Kraman' ın [9] bu konudaki çalışmaları sayesinde solunum sisteminin akustik karakteristiği daha iyi anlaşılmıştır. Ancak tüm bu çalışmalara rağmen soluk seslerine getirilen öznel tanımlamalar ve bu tanımlamaların konuşulan dillere göre farklılık göstermesi tanı zorluklarına neden olmaya devam etmiştir. Öznel yaklaşımlara dayanan tanıyı daha nesnel bir hale getirmek için en akla yakın çözüm soluk seslerinin parametrelerle temsil edilmesi ve daha sonra bu parametreleri karşılaştırarak tanıya gidilmesidir. Yine son yıllarda gelişen veri toplama ve işaret işleme yöntemleri soluk seslerinin analizine ve sınıflandırılmasına olanak vermiştir. Cohen ve grubu [4] tarafından yapılan çalışmada normal soluk sesleri dinlenildikleri yere göre sınıflandırılmaya çalışılmıştır. Bu çalışmada AR parametreleriyle modelleme ve Mahalanobis uzaklık ölçüsü ile örüntü tanıma yöntemleri kullanılmıştır. Yine Cohen tarafından yapılan benzer bir çalışmada [1] uyarı ses olarak konuşma sesi kullanılmıştır. Iyer ve grubunun yaptığı çalışmada [10] ise akciğer seslerinin kaynak ve iletim modellerinin belirlenmesine çalışılmıştır. Bu çalışmada da AR modelleme kullanılmış ve sonuçta göğüs kafesindeki seslerden yararlanılarak seslerin kaynağının

ve iletim ortamının kestirilebileceği gösterilmiştir. Akciğer seslerinin analizinde yaygın olarak kullanılan bir yöntem de spektrum analizi yöntemidir. Çalışmalarını bu yönde yapan Fenton ve grubu [11] astım hastalarından kaydedilen seslerin güç spektrumunda yüksek frekanslarda tepecikler belirlediğini görmüşler ve bu tepeciğin astımın belirtisi olan ıslık sesi olarak nitelendirilebilmesi için tepenin genliği ile ortalama güç arasında bir ilişki belirlemeye çalışmışlardır. Bu çalışmada soluk borusundan (trachea) alınan seslerin astım hakkında, ciğerlerden alınan seslere göre daha çok bilgi taşıdığı anlaşılmıştır. Benzer bir çalışma da Pasterkamp ve grubu [12] tarafından yapılmıştır. Yeni doğmuş sağlıklı çocuklar üzerinde yapılan çalışmada soluk sesinin şiddetinin soluk alma ve vermenin evrelerine göre değişimi incelenmiştir. Sonuçta küçük çocuklarla yetişkinlerin spektrumlarının yapısal olarak aynı oldukları, ancak çocuklarda, dışsoluma sırasında ses şiddetinin biraz daha az olduğu görülmüştür. Akciğer seslerinin hava akışı ile olan ilişkisi de incelenen konular arasındadır. Shykoff ve grubu tarafından yapılan çalışmada [13] çeşitli hava akış hızlarında kaydedilen soluk seslerinin incelenmesinde içsoluma sırasında sesin genliğinin ağızdaki hava akışı ile karesel olarak değiştiği gözlenmiştir. Soluk seslerinin spektrumlarının ve genliklerinin içsoluma ve dışsolumaya göre farklılıklar göstermesi analizin akış evreleriyle eşzamanlı yapılmasını gerekli kılmaktadır. Genelde kullanılan yöntem soluk sesleri ile birlikte hava akışının da kaydedilmesidir. Cohen ve grubu yaptıkları bir çalışmada akış evresi bilgisinin enerji zarfı yöntemiyle soluk sesinden de elde edilebileceğini göstermiştir [14].

Soluk seslerinin analizinde en önemli aşamalardan birisi seslerin kayıt aşamasıdır. Carbonneau'nun yaptığı bir çalışmada uygun bir kayıt sistemi şöyle betimlenmektedir [15] :

1- Uygun bir kayıt sistemi düşük genlikteki işaretlere duyarlı olmalıdır. Forgacs'a göre [3] soluk seslerini ağızdan duyulabilmesi için en az 0.35 l/s, göğüs kafesinden duyulabilmesi için en az 0.75 l/s hava akış hızı gereklidir.

2- Sistemin frekans yanıtı ilgilenilen aralıkta düz olmalıdır. Soluk sesleri, 1kHz'den daha yüksek frekanslarda ihmal edilebilir güce sahiptirler.

3- Alçak frekanslar iyi süzölmelidir. 75-80 Hz ve daha düşük frekanslarda soluk seslerinin genliği kalp seslerinin genliğine göre çok daha azdır. Kalp seslerinin ve kas seslerinin soluk seslerine karışması analiz ve sınıflandırmada problem yaratacağından bunları engellemenin bir yolu çok alçak frekanslı bu sesleri yüksek-geçiren bir süzgeçle süzmektir. Başka bir yol da ECG işaretinin referans işaret alınarak kalp seslerinin uyarlamalı süzgeç yardımıyla süzülmesidir [16].

4- Kayıt sırasında kullanılan mikrofonlar hava bağlaşımlı olmalı ve deriye değmesi engellenmelidir. Bu yolla mikrofon üzerine uygulanan duruk kuvvetin ölçümler üzerindeki etkisi ortadan kaldırılabilir. Yapılan denemelerde en uygun mikrofon tipinin hava bağlaşımlı elektret kondansatör mikrofon olduğu görülmüştür.

Yukarıda anlatılanlardan da anlaşılacağı gibi akciğer hastalıklarının tanısı oldukça karmaşık bir iştir. Akciğer seslerinin analizi, yapılması gerekenlerden sadece birisidir. Akciğer seslerinin denegin

yaşına, soluma örüntüsüne, göğüs kafesinin kalınlığına, ve kayıt alınan yerlere ve hava akışına bağlı olması, tanı aşamasında bir çok parametrenin gözönüne alınmasını gerektirmektedir.

Bu çalışmada, sağlıklı ve hastalıklı deneklerden kaydedilen akciğer seslerinin tanınması ve sınıflandırılması için bir yöntem önerilmiştir. Sınıflandırma yönteminin açıklanmasından önce solunum fizyolojisi ile ilgili gerekli bilgiler ve kullanılan işaret işleme yöntemleri anlatılmıştır. Sonuç bölümünde elde edilen sonuçlar ve gelecekte yapılacak araştırmalar için öneriler sunulmuştur.

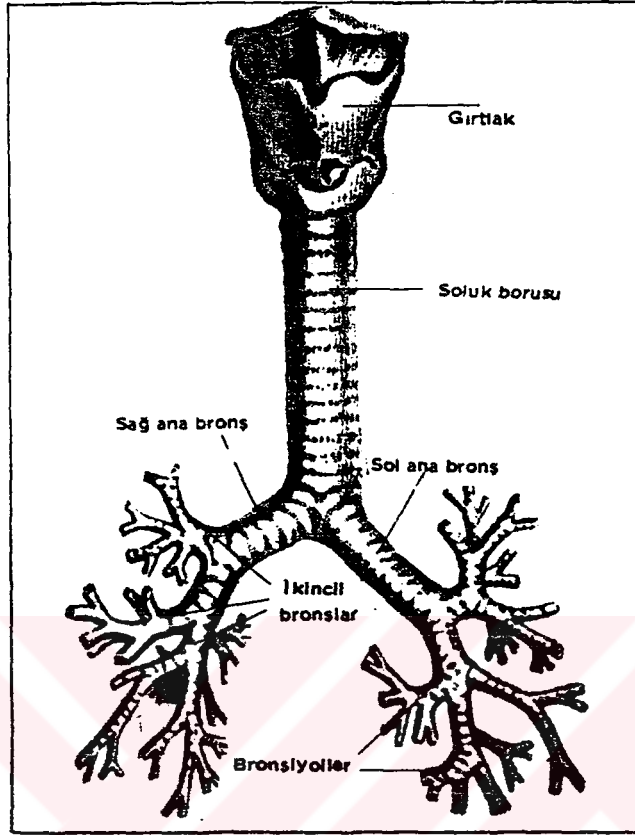


## BÖLÜM 2. SOLUNUM FİZYOLOJİSİ VE SOLUNUM SESLERİ

### 2.1. Solunum Sistemi Ve Akciğerler

Vücudun metabolizmayı gerçekleştiren çeşitli kesimlerine oksijen sağlamak ve karbondioksiti uzaklaştırmak amacıyla akciğerlere çekilen, ya da akciğerlerden dışarı atılan havaya soluk, soluk alıp verme eylemine de solunum denir. Hava akciğerlere burun, ağız, yutak, gırtlak ve soluk borusundan oluşan üst solunum yollarından geçerek girer. Üst solunum yollarının görevi solukla alınan havayı süzmek, temizlemek ve ısı ayarlaması yapmaktır. Soluk borusu göğüs boşluğuna girdikten sonra, sağ ve sol bronşlara ayrılır. Akciğerlere giren bronşlar yeniden bölünürler. Her bölünmede bronşların sayısı artar ve boyutları küçülür. Yirmi kadar bölünmeyle bir milyon kadar bronşçuk oluşur. En sondaki bronşçuklar bir defa daha bölünerek solunum bronşçukları denilen kısa kanalları oluştururlar. Bu kanalları küme halinde toplanmış hava geçitleri izler. Soluk borusu ve bronşların yapısı Şekil 2-1'de görülmektedir.

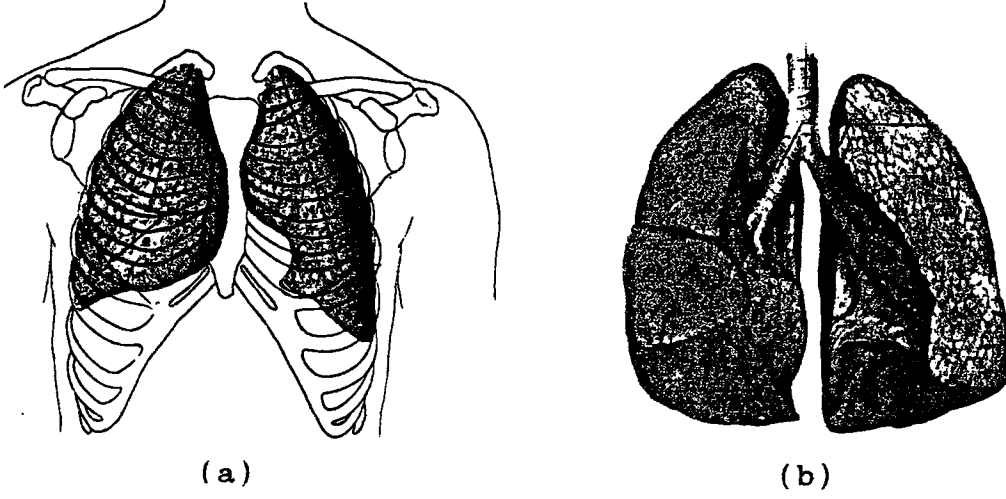
Solunum bronşçuğunun akciğere girdiği kesime akciğer lopçuğu denir. Her lopçuk bir çok hava keseciğinden oluşur. Gaz değişiminin gerçekleştiği başlıca kesim burasıdır. Akciğerlerden her biri çift kat bir zarla kaplıdır. Buna plevra zarı denir. Bu zarın dış tabakası göğüs kafesine döşenmiştir. Akciğerler tepesinden tabanına kadar ikiye bölünmüş birer yarım koni biçimindedir. Bir akciğer bir tepe (apex), bir



Şekil 2-1. Soluk borusu ve bronşların yapısı.

taban (basis), iki yüz (facies) ve üç kenara (margo) sahiptir. Tepe akciğerin en dar kısmı olup köprücük kemiği hizasındadır. Taban akciğerin alttaki genişlemiş kısmıdır.

Yaklaşık olarak sağ akciğer 650 gr., sol akciğer 550 gr.ağırlığındadır. En geniş solunum anında iki akciğer birden 5-6 litre hava alır. Sağ akciğerin hacmi soldakinden  $\frac{1}{5}$  oranında fazladır. Çok esnek olan akciğerler soluk alırken büyük bir genişleme, soluk verirken de büyük bir daralma yeteneğine sahiptirler. Akciğerlerin göğüs kafesi içinde ve soluk borusu ile birlikte yerleşimleri Şekil 2-2a ve 2-2b ' de verilmiştir.



Şekil 2-2. Akciğerlerin (a) göğüs kafesi içinde ve (b) solukborusu ile birlikte yerleşimleri.

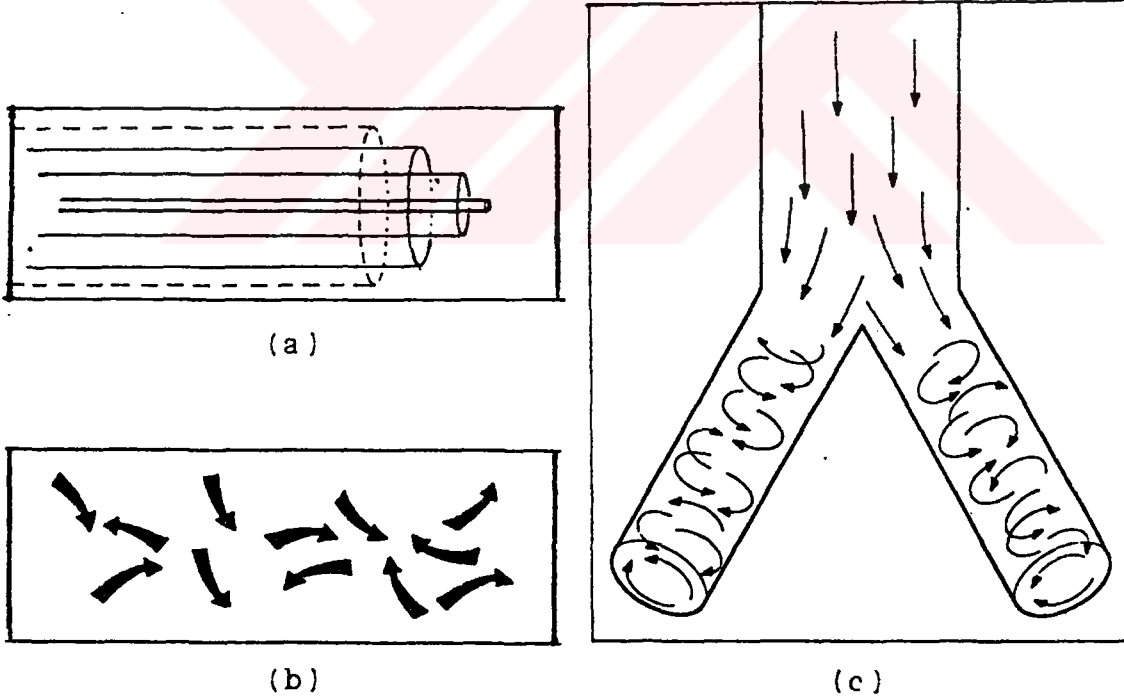
Solunum sisteminin bir çok hastalığı bronşlarla ilgilidir. Örneğin astım, çeperlerindeki düz kasların kasılmaları sonucu bronşların daralmaları, bronşit, bronşların yangılanması sonucu daha fazla mukus salgılamalarıdır. Öte yandan zatülcenp ve zatürree gibi hastalıklar akciğer loblarının, ya da zarının yangılanması sonucu oluşurlar.

## 2.2. Solunum Sesleri

Solunum ve akciğer sesleri, ciğer ve solunum yollarındaki gaz basıncının hızlı değişimi ve katı dokuların titreşimi sonucunda oluşurlar. Ancak bu sesler, göğüs kafesinin alçak-geçiren bir özellik göstermesi yüzünden, toplam enerjilerinin ve frekans bileşenlerinin bir kısmını iletim sırasında yitirirler.

Akciğerlerden duyulan sesler hava akış örüntüleri ile yakından ilintilidir. Bilinen üç çeşit akış örüntüsü

Şekil 2-3' de görülmektedir. İçiçe akışta gazlar bir borunun duvarlarına paralel olarak içiçe akarlar. En uçta bulunan hava yollarında ve keseciklerdeki gaz akışı bu türden, sessiz bir akıştır. Burgaçsal akış, kritik bir akış hızının üstüne çıkıldığında içiçe akan gazların akışın olağan yönüne ters ya da kesen bir yönde rasgele hareket etmeleri sonucunda oluşan akıştır. Gaz basıncının bu türden hızlı değişimler göstermesi sonucunda oluşan ses, 200 Hz ile 2000 Hz arasında değişen rasgele genlikli gürültü sesidir. Girdapsal akış ise boruda bulunan eğriliklerin ya da köşelerin sebep olduğu dairesel akıştır. Girdaplı bir hava akışı dar bir yarığa rasgeldiğinde belirlenebilen perdeden bir ses tonu oluşturur [3].



Şekil 2-3. Hava akış örüntüleri (a) içiçe akış (b) burgaçsal akış (c) girdapsal akış.

Akciğer sesleri en genel halde normal soluk sesleri ve anormal sesler olarak ikiye ayrılırlar. Soluk sesleri duyuldukları bölgeye göre, anormal sesler de oluşum şekilleri ve yapılarına göre çeşitli gruplara ayrılırlar. Akciğer hastalıklarının tanısında yararlanılan bir başka ses türü de konuşma sesidir.

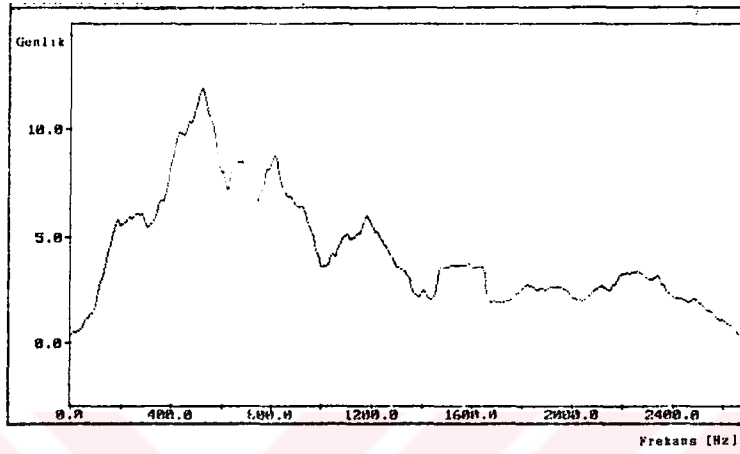
### 2.2.1. Normal Soluk Sesleri

Normal soluk sesleri oluşum şekillerine göre ikiye ayrılırlar. Bunlardan ilki içsoluma sırasında keseciklere giren havanın oluşturduğu sestir ve "yerel gürültü", ya da "vesiküler gürültü" adını alır. Diğer, havanın gırtlaktan geçmesi sonucu oluşan sestir ve "gırtlak gürültüsü" adını alır [1]. Bu sesleri "göğüs duvarından duyulan soluk sesleri" ve "ağızdan duyulan soluk sesleri" olarak ayırmakta olasıdır [3].

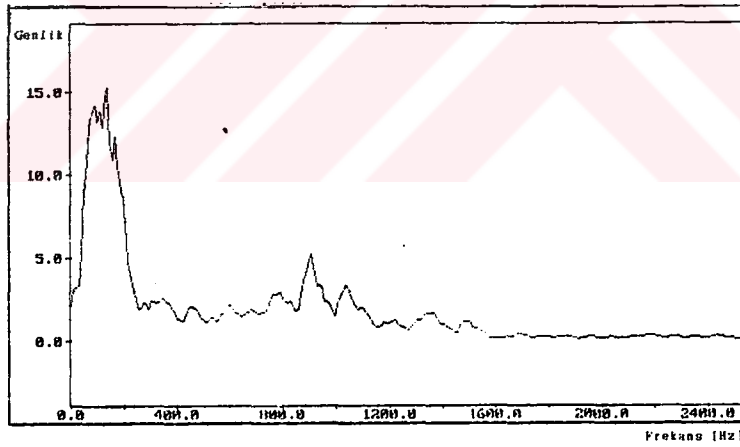
Ağızdan duyulan sesler ile göğüs duvarından duyulan sesler arasında önemli farklar vardır. Göğüs duvarından duyulan sesler, göğüs kafesinin alçakgeçiren bir süzgeç özelliği göstermesi yüzünden, ağızdan ya da soluk borusundan duyulan seslere göre daha düşük bir frekans bandında sınırlanmışlardır. Şekil 2-4a ve 2-4b 'de sağlıklı bir denekten alınan, soluk borusu ve sol tabana ait soluk seslerinin frekans spektrumları verilmiştir. Şekilden de görüldüğü gibi soluk borusundan alınan soluk sesi daha yüksek frekans bileşenleri içermektedir. Soluk borusu sesleri tüm solunum çevrimi boyunca duyulmasına rağmen göğüsden duyulan sesler dışsolumanın yarısından sonra iyice zayıflayarak duyulmaz olurlar.

Soluk seslerinin kaynağı konusunda 19. yy'dan beri süregelen tartışmalarda soluk seslerinin önce en uç

bronşiyellerde ve keseciklerde meydana geldiği ileri sürülmüştür. Başka bir düşünce de, soluk seslerinin, üst solunum yollarından akan gaz sesinin akciğerlerden



(a)



(b)

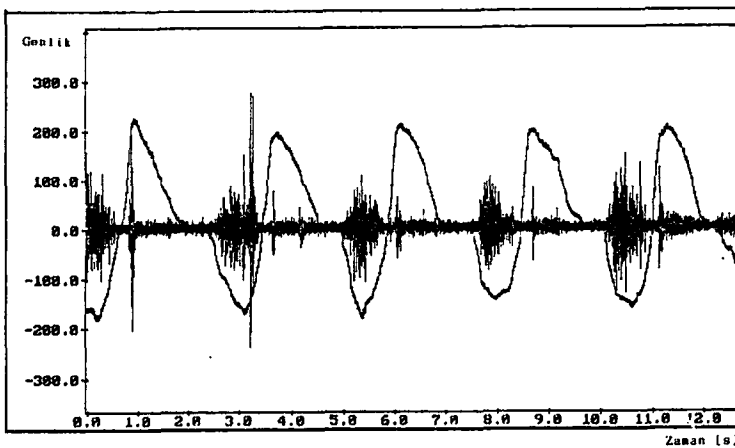
Şekil 2-4. Sağlıklı bir denegın soluk sesi frekans spektrumları (a) soluk borusu (b) sol taban.

geçerken değişime uğraması sonucunda oluştuğu yolundadır. Yapılan son çalışmalar göğüs duvarından

duyulan soluk seslerinin oluşmasında ana bronşdan geçen havanın ve akciğer yapısının etkisi olduğunu göstermiştir. Bu düşünceye göre hem ağızdan, hem de göğüsden duyulan seslerin kaynağı bir ucunda ağız, diğer ucunda da bölünmeli bronş yapısının bulunduğu burgaçsal solunum bölgesidir [3].

Soluk seslerinin oluşumunda etkili olan bir başka etmen de hava akışıdır. Akış işareti Şekil 2-5'de görüldüğü gibi soluk sesi ile birlikte değişim gösterir. Ortalama içsoluma sesi genliği ile akış işareti sağlıklılarda 1 l/s'ye kadar doğrusal bir ilişki gösterirler. Bu sınırın üstünde sesin genliği akışın genliğine üssel olarak bağlıdır [3]. Soluk seslerini ağızdan duyulabilmesi için en az 0.35 l/s, göğüs kafesinden duyulabilmesi için en az 0.75 l/s hava akış hızı gereklidir.

Göğüs kafesinden dinlenen seslerdeki değişimler akciğerlerin durumu hakkında bilgi verirler. Örneğin hava yolları ve göğüs kafesi arasındaki akciğer dokusu



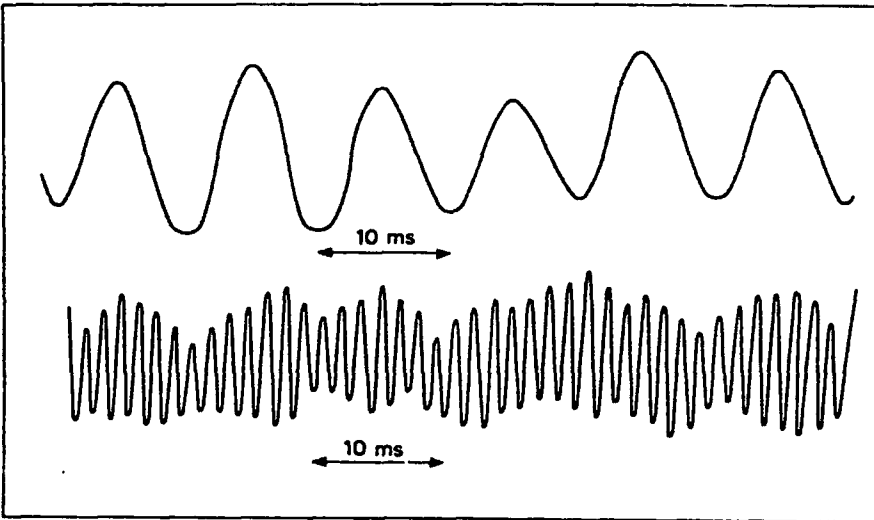
Şekil 2-5. Akış işareti ile soluk sesinin birlikte değişimi.

tıkanma ya da bağdokulaşma sonucunda havasız kaldığında göğüs kafesinden duyulan sesler tıpkı soluk borusundan duyulan soluk sesleri gibi yüksek frekans bileşenleri içerirler. Amfizem gibi hastalıklarda da akış hızı azaldığından soluk sesi duyulmaz olur.

### 2.2.2. ıslık Sesleri

Ağızdan ve akciğerlerden duyulan ve belirlenebilir süresi 100 ms'den fazla olan müziksel seslere ıslık sesleri adı verilir.

ıslık sesinin en alçak frekanslı bileşeni sesin perdesini belirler. Buna temel frekans da denir. Frekans spektrumları incelendiğinde ıslık sesleri spektrumunda, genellikle 2000 Hz'in altında ortaya çıkan dar tepecikler olarak gözükürler [11]. ıslık sesleri yüksek perdeli, alçak perdeli, içsolumaya ait, dışsolumaya ait, kısa, uzun, tek sesli ve çok sesli ıslıklar olarak



Şekil 2-6. Alçak ve yüksek perdeli ıslık sesi dalga biçimleri.

sınıflandırılabilirler. Şekil 2-6'da alçak ve yüksek perdeli ıslık sesi dalga biçimleri görülmektedir.

Islık seslerinin oluşumunu açıklayan en uygun model çocuk düdüğü modelidir. Bu modelde sabit frekansta oluşan ses, sesi oluşturan gazın yoğunluğundan bağımsızdır. Oluşan notanın perdesi hava yolunun uzunluğuna değil, titreşen yapının kütlesine ve elastik özelliklerine bağlıdır. Bu modelden yola çıkılarak ıslık seslerinin, hava yollarını tıkayan tümörlerin, sümük maddesinin, ya da bronşiyel kasılmaların hava yollarıyla birleştiği yerlerdeki gaz akışı sonucunda titreşime girmesinden dolayı oluştuğu söylenebilir.

Tek sesli ıslıklar tek bir notadan, ya da farklı zamanlarda başlayıp biten bir çok notadan oluşurlar. Tek sesli ıslıklar genellikle bronküsün bir tümör tarafından tıkanması sonucunda meydana gelirler. Eğer tıkanma katı halde ise ıslık tüm solunum çevrimi boyunca duyulabilir, tıkanma esnek ise bu ses bazı durumlarda duyulmayabilir. Tekli ya da çoklu tek sesli ıslıklar astım hastalığının karakteristik belirtileridir. Astım hastalığında hava akışını engelleyen bronşiyel kasılmalar, ya da yutulan sümük maddesinin neden olduğu ıslıkların kaynağı nadiren belirlenebilir. Geniş merkezi hava yollarında oluşan ıslık sesleri genelde ağızdan da duyulabilen, yüksek genlikli seslerdir. Alt hava yollarında oluşan sesler ise ağıza ulaşmadan süzöldükleri için sadece göğüs duvarından duyulabilirler.

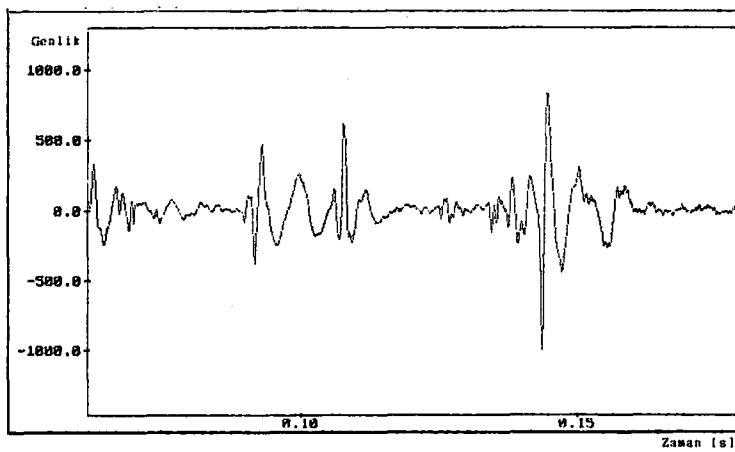
Çok sesli ıslıklar aynı anda başlayıp aynı anda biten notalardan oluşurlar. Bu türden ıslıklar merkez solunumdalının dinamik olarak sıkıştırılması sonucunda, daha çok dışsoluma evresinde meydana

gelirler. Çok sesli ıslıkların en belirgin duyulduğu hastalık süregelen tıkanmalı bronşit hastalığıdır.

### 2.2.3. Çıtırtı Sesleri

Ağızdan, ya da göğüs duvarından duyulan, kısa patlamalı, müziksel olmayan süreksiz seslere çıtırtı sesleri adı verilir.

Çıtırtılar, hava yollarında bulunan fazla sıvılardan geçen havanın kabarcık yapması, ya da hava yollarının ani olarak açılıp kapanmaları sonucunda oluşurlar [1]. Çıtırtı sesleri kendi aralarında yüksek perdeli, alçak perdeli, az sayıda, çok sayıda, içsolumaya ait, dışsolumaya ait, erken, ya da geç çıtırtılar gibi sınıflara ayrılırlar. Tipik bir çıtırtının süresi yaklaşık 20 ms kadardır. Şekil 2-7'de zatürreeli bir hastadan alınan çıtırtı dalga şekilleri görülmektedir.



Şekil 2-7. Zatürreeli bir denekten alınan çıtırtı sesi dalga biçimleri.

Geç içsoluma çıtırtıları akciğer bağdokulaşmasında görülen ve içsolumanın sonlarına doğru ortaya çıkan çıtırtılardır. Bu tür çıtırtılar, akciğerin havasız kalan kısımlarındaki hava yollarının içsolumanın geç evrelerine kadar kapalı kalması ve sonra aniden açılarak patlamalı bir ses oluşturmaları sonucu meydana gelirler.

İçsolumanın başlarında duyulan erken içsoluma çıtırtıları süregelen tıkanmalı akciğer hastalıklarının, özellikle süregelen bronşitin belirtisidir. Bu çıtırtılar geç içsoluma çıtırtılarının aksine daha kalın ve daha az sayıdadırlar ve alt loblardan olduğu kadar ağızdan da duyulabilirler. Çıtırtıların genlikleri ve aralarındaki uzaklık çoğunlukla aynıdır.

Çıtırtıların, soluk seslerinin frekans spektrumlarını fazla etkilememeleri, bu dalga biçiminin zaman bölgesinde incelenmesini gerekli kılmaktadır. Şekil 2-7'den de görüldüğü gibi çıtırtılar soluk dalga biçiminde duran olmayan bölgeler yaratmaktadırlar. Duran olmayan çıtırtıların duran soluk seslerinden ayrılması için bir yöntem doğrusal olmayan süzgeçler kullanmaktır [17]. Çıtırtılar ayrıldıktan sonra parametrik olarak modellenenebilirler. Parametrik modellemede bir yaklaşım çıtırtıların üssel olarak sönümlenen sinüsoidaller olarak modellenmesi olabilir. Böyle bir model

$$x_k = \sum_{i=1}^N A_i e^{-\alpha_i k} \cos(w_i k + \theta_i) \quad (2.1)$$

denklemleri ile verilebilir [18] .

### 2.3. Konuşma Sesleri

Akciğer ve solunum yolları hastalıklarının tanısında kullanılan başka bir yöntem de konuşma seslerinin göğüs duvarından dinlenmesidir. Bu yöntemde konuşma sesi göğüs duvarını uyaran işaret olarak kullanılır ve göğüsten dinlenen sesin değişimine göre tanı hakkında bir karara varılır. Bu yöntemde dayanan tanı seslerini dört grup altında toplamak olasıdır :

a) Normal Konuşma Sesleri Sağlıklı kişilerde, göğüs duvarının hemen her yerinden duyulabilen, belirsiz, boğuk ve az anlaşılır konuşma sesleridir.

b) Bronkofoni Keseciklerin sıvıyla, ya da katı bir dokuyla dolduğu bölgelerden duyulan belirli, anlaşılabilen seslerdir. Bronkofoni genelde hastaya "kırk" gibi düşük frekanslı kelimeler söyletilerek test edilir.

c) Fısıltı Keseciklerin sıvı veya katı bir dokuyla dolduğu bölgelerden duyulan anlaşılabilir fısıltı sesleridir. Normalde fısıltı seslerinin oluşmasında ses tellerinin etkisi olmadığından oluşan sesler yüksek frekanslıdır ve normal bir insanda göğüs duvarından duyulmaz.

d) Egofoni Göğüs zarı sıvı birikiminde duyulan yüksek genlikli konuşma sesidir. Egofoni testi için hastaya /a/ harfi söyletilir. Egofoni olduğu durumlarda /a/ sesi göğüs duvarından /e/ sesi olarak duyulur.

#### 2.4. Akciğer ve Solunum Yolu Hastalıkları

Solunum seslerinin analizinde kullanılan yöntemler kadar incelenen hastalık türlerinin belirtilerinin bilinmesi de önemlidir. Bu bölümde sık rastlanılan hastalıklar, duyulan solunum sesleri açısından incelenenecektir.

Amfizem Akciğerdeki hava keseciklerinin aşırı derecede gerilip genişlemelerine verilen addır. Gerilen kesecikler arasındaki duvarlar bir süre sonra yırtılır ve büyük kesecikler oluşur. Genelde bronşitle beraber ya da bronşitten sonra görülür. Bu tür hastalarda dışsoluma uzundur. Normal bir insanda zorlu bir içsolumadan sonra yapılan zorlu bir dışsoluma 4 saniyeden önce tamamlanırken amfizemlilerde bu süre 4 saniyeyi geçer. Akciğerden duyulan seslerin genlikleri azalır. Ancak yapılan deneyler aynı akış hızında solutulan normal ve amfizemli hastaların akciğer seslerinin genliklerinin aynı olduğunu göstermiştir. Bu da sesteki azalmanın akış hızına bağlı olduğunu göstermektedir [19]. Ses azalmasıyla beraber hekimlikte "kuru raller" olarak bilinen hırıltıyı andıran (ronflan raller) ve ince düdük sesini andıran (sibilan raller) sesler de duyulabilir.

Zatürree (Pnömoni) Bir veya birden fazla akciğer lobunun mikropların etkisiyle yangılanmasına verilen addır. Genelde bakteriyel ve solunum zatürreesi olarak ikiye ayrılır. Solunum zatürreesi solunum sistemindeki bir anormalliğin akciğerleri hastalığa elverişli kılmasının sonucudur. Duyulan sesler hastalığın evrelerine göre değişir. Hastalığın ilk evresinde sadece içsolumada ince çıtırtılar duyulur. Bu sesler tıpkı kulak önünde ovuşturulan saç veya kartopu yaparken

sıkılan karın çıkardığı seslere benzer. Akciğer keseciklerinde bulunan yapışkan eksüda maddesi dışsoluma sırasında kesecik duvarlarını birbirlerine yapıştırır. Ardından gelen içsolumada keseciklere giren hava yapışmış duvarları birbirinden ayırırken bir dizi çıtırtı oluşur. Bu çıtırtıya hekimlik dilinde "krepitan" ya da "ince raller" denir. Hastalığın bazı durumlarında soluk borusu ve bronşlar yolu ile sırta kadar gelen ses hem içsolumada, hem de dışsolumada duyulur. Bu durumda göğüs duvarından alınan sesin frekans spektrumu incelenirse yüksek frekans bileşenlerinin arttığı görülür.

Zatülcenp (Plörezi) Akciğerleri saran zarın ıvegen veya süreğen yangılanmasına verilen addır. Kuru ve sulu zatülcenp olarak ikiye ayrılır. Kuru zatülcenp göğüs zarı üstünde bağdoku maddesinin birikimi sonucunda bir tabaka oluşmasıdır. Plevra zarının tabakaları arasında su birikmesiyle ortaya çıkan yangı ise sulu zatülcenp adını alır. Hastalığın başlangıcında "frotman" denilen ve yangılanma sonucu yüzeyleri kayganlığını yitirmiş olan göğüs zarı yapraklarının her solunumda birbirlerine sürtünmesi sonucunda oluşan sesler duyulur. Hem içsolumada, hem de dışsolumada duyulan bu sesler meşin gıcirtısına benzer. Hastalığın ileri hallerinde, plevra yaprakları arasında sıvı toplandığında sesler kaybolur.

Bronkopnömoni Bronşçukların yangılanması olan bronşiyolitın ileri bir dönemidir. Genelde keseciklerin yangıyla dolması olan zatürree ile birlikte görülür. Hem içsolumada, hem de dışsolumada akciğer tabanlarında, bronşiyellerde biriken salgının, havanın geçişi ile parçalanması sonucu oluşan ve "sukrepitan raller" olarak adlandırılan kalın çıtırtılar duyulur. Bu çıtırtılar bardağa boşaltılan gazozdan çıkan hışırtı seslerine

benzer.

Bronşial Astım Akciğer içi bronşlarının yangısal veya alerjik nedenlerle daralması veya tıkanması sonucunda oluşan ve nöbetler halinde gelip geçen nefes darlığına verilen addır. Nöbet sırasında dışsoluma uzar ve göğüs duvarından hırıltı ve ıslık benzeri sesler duyulur. Çoğu zaman bu sesler göğüs duvarı dinlenmeden, dışarıdan da duyulabilir.



### BÖLÜM 3. İŞARET İŞLEME VE ÖRÜNTÜ TANIMA YÖNTEMLERİ

Doğada bulunan konuşma işaretleri, haberleşme işaretleri ve biomedikal işaretler gibi bir çok işaretler rasgele süreçler olarak kabul edilebilirler. Genelde bu tür işaretler doğrusal ve durağan olmayan işaretlerdir. Fakat yapılan çeşitli yaklaşıklıklarla durağan ve doğrusal işaretler için geliştirilmiş olan analiz yöntemleri bu tür işaretlerde de kullanılabilir.

Bu bölümde ilk önce solunum seslerinin analizinde kullanılacak olan işaret işleme yöntemleri anlatılacaktır. Zaman serilerinin doğrusal olarak modellenmesine kısaca değinildikten sonra özbağımlı (autoregressive -AR), özbağımlı-yürüyen ortalamalı (autoregressive-moving average - ARMA) ve yürüyen ortalamalı (moving average - MA) modelleme üzerinde durulacak ve AR modellerin en küçük kareler yöntemiyle belirlenmesi anlatılacaktır. Ayrıca temel örüntü tanıma kavramlarına değinildikten sonra, soluk örüntülerinin sınıflandırılmasında kullanılan en yakın komşu yöntemi ve uzaklık ölçüleri açıklanacaktır.

#### 3.1. Zaman Serilerinin Doğrusal Olarak Modellenmesi

Durağan işaretlerin modellenmesine en iyi yaklaşımlardan biri parametrik modellemedir. Parametrik modellemede süreç oransal ve nedensel olarak modellenir. Modellemeden sonra işaret parametrelerle temsil edilebilir.

$\{s(k)\}$  doğrusal bir sistemin çıkışı,  $\{u(k)\}$  sistemin bilinmeyen girişi,  $\{\xi(k)\}$  eklenir dış gürültü olmak üzere  $k$  anındaki çıkış işareti

$$s(k) = -\sum_{i=1}^p a_i s(k-i) + \sum_{i=0}^q b_i u(k-i) + \sum_{i=0}^p a_i \xi(k-i) \quad (3.1)$$

fark denklemi ile verilir [20]. Çoğunlukla  $\{\xi(k)\}$  dış gürültü dizisinin beyaz gürültü tarafından sürülen bir sistemin çıkışı olduğu düşünülürse

$$\sum_{i=0}^n c_i \xi(k-i) = \sum_{i=0}^m d_i n(k-i) \quad (3.2)$$

denklemi geçerli olur.

$$A(z^{-1}) = \sum_{i=0}^p a_i z^{-i} \quad ; \quad a_0 = 1$$

$$B(z^{-1}) = \sum_{i=0}^q b_i z^{-i}$$

$$C(z^{-1}) = \sum_{i=0}^n c_i z^{-i}$$

$$D(z^{-1}) = \sum_{i=0}^m d_i z^{-i} \quad (3.3)$$

olmak üzere denklem (3.1)

$$S(z) = \frac{B(z^{-1})}{A(z^{-1})} U(z) + \frac{D(z^{-1})}{C(z^{-1})} N(z) \quad (3.4)$$

şeklinde yazılabilir. Denklem (3.4)'deki  $\{s(k)\}$  dizisi

$$\begin{aligned} \beta_s^T &= [a_0, a_1, \dots, a_p, b_0, b_1, \dots, b_q] \\ \beta_n^T &= [c_0, c_1, \dots, c_n, d_0, d_1, \dots, d_m] \end{aligned} \quad (3.5)$$

vektörleri ile modellenenabilir [20]. Burada  $\beta_s$  sistem parametre vektörü,  $\beta_n$  gürültü parametre vektörüdür. Denklem (3.4) de  $z^{-1}$  dönüşümü cinsinden verilen  $\{u(k)\}$  dizisi genellikle beyaz gürültü olarak alınır. Solunum sesleri gibi, girişi tam olarak bilinmeyen sistemlerde bu en akla yakın seçimdir. Bu durumda  $\beta_s$  vektörü, beyaz gürültü dizisi ile renkli süreç arasında doğrusal bir dönüşüm gerçekleştiren bir parametre vektörü olarak düşünülebilir. Denklem (3.4)'de dış gürültüler gözardı edilerek  $N(z) = 0$  varsayılırsa

$$s(k) = -\sum_{i=1}^p a_i s(k-i) + \sum_{i=0}^q b_i u(k-i)$$

$$S(z) = \frac{B(z^{-1})}{A(z^{-1})} U(z) \quad (3.6)$$

denklemleriyle verilen ARMA modeli elde edilir. ARMA modeli aynı zamanda kutup-sıfır modeli olarak da bilinir. Bu modelin bir özel durumu  $b_i = 0, 1 \leq i \leq q$  olduğu durumdur. Bu durumda denklem (3.6)

$$s(k) = -\sum_{i=1}^p a_i s(k-i) + b_0 u(k) \quad (3.7)$$

ve  $G = b_0$  alınarak her iki tarafın  $z$  dönüşümü alınır

$$S(z) = \frac{G}{A(z^{-1})} U(z) \quad (3.8)$$

şekline dönüşür. Elde edilen bu modele özbağımlı (AR) model adı verilir. Yine ARMA modelinde  $a_i = 0, 1 \leq i \leq p$  alınır denklemin (3.6)

$$s(k) = \sum_{i=0}^q b_i u(k-i)$$

$$S(z) = B(z^{-1})U(z) \quad (3.9)$$

şekline dönüşür. Elde edilen bu modele de yürüyen ortalamalı (MA) model adı verilir. Denklemlerden de anlaşılacağı gibi, AR modeli tüm-kutup, MA modeli tüm-sıfır modellerdir. Üç model içinde en çok kullanılan model AR modelidir. ARMA modelinin derecesinin küçük olması, MA modelinin ise durağan olmayan işaretleri en iyi modelleyen model olmasına karşın AR modelleri her ikisinin de yerine kullanılabilir [20]. Ayrıca AR

modeller, hesaplama karmaşıklığı açısından diğerlerine göre daha az karmaşık, model derecesinin değişmesine daha az duyarlıdır.

### 3.2. AR Katsayılarının En Küçük Kareler Yöntemi ile Kestirimi

Zaman serilerinin analizinde önemli olan parametre kestirimi için bir çok yöntem olmasına karşın en çok bilinen ve kullanılan yöntem en küçük kareler yöntemidir. Bu yöntemin en çok kullanılmasının nedeni yakınsama özellikleri ve yuvarlatma hatalarına karşı olan gürbüzlüğüdür.

En küçük kareler yönteminde incelenen sistemin parametreleri, öngörü hatasını en aza indirecek şekilde belirlenirler. Sistemin girişi bilinmediğinden sistemin çıkışı ancak daha önceki çıkışların doğrusal bileşiminden yararlanılarak öngörülebilir. Sistemin  $k$  anındaki çıkışının öngörülmüş değeri olan  $\hat{s}(k)$  geçmiş çıkış değerlerine

$$\hat{s}(k) = - \sum_{i=1}^p a_i s(k-i) \quad (3.10)$$

denklemini ile bağlıdır. Böyle bir sistemde öngörü hatası ve karesel hatanın beklenen değeri  $E_T$ ,

$$e(k) = s(k) - \hat{s}(k) = s(k) + \sum_{i=1}^p a_i s(k-i) \quad (3.11)$$

$$E_T = E[e^2(k)] = E\left[\left(s(k) + \sum_{i=1}^p a_i s(k-i)\right)^2\right] \quad (3.12)$$

denklemleri ile verilir. Karesel hatayı en küçük yapacak olan katsayılar

$$\frac{\partial E_T}{\partial a_m} = 0 \quad , \quad 1 \leq m \leq p \quad (3.13)$$

denklemlerinden bulunabilir. Denklem (3.12)'deki  $E_T$ , denklem (3.13)'de yerine konursa

$$\sum_{i=1}^p a_i E[s(k-i)s(k-m)] = -E[s(k)s(k-m)] \quad , \quad 1 \leq m \leq p \quad (3.14)$$

ile verilen normal denklem elde edilir. Denklem (3.12) ve (3.14)'ten yararlanılarak minimum karesel hata

$$E_T = E[s^2(k)] + \sum_{i=1}^p a_i E[s(k)s(k-i)] \quad (3.15)$$

olarak elde edilebilir. Eğer  $s(k)$  durağan bir işaretse

$$E[s(k-i)s(k-m)] = R(m-i) \quad (3.16)$$

denklemini geçerli olur. Burada  $R(m)$  özilinti işlevidir. Denklem (3.16), denklem (3.14) ve (3.15)'de yerlerine konursa

$$\sum_{i=1}^p a_i R(m-i) = -R(m) \quad 1 \leq m \leq p \quad (3.17)$$

$$E_T = R(0) + \sum_{i=1}^p a_i R(i) \quad (3.18)$$

şekline dönüşür.  $s(k)$ 'nin durağan olmadığı durumda ise

$$E[s(k-i)s(k-m)] = R(k-i, k-m) \quad (3.19)$$

denklemini geçerli olur. Burada  $R(t_1, t_2)$ ,  $t_1-t_2$  aralığında  $s(k)$ 'nin durağan olmayan özilinti işlevini göstermektedir.  $k=0$  anında normal denklemler,

$$\sum_{i=1}^p a_i R(-i, -m) = -R(0, -m) \quad 1 \leq m \leq p \quad (3.20)$$

$$E_T = R(0, 0) + \sum_{i=1}^p a_i R(0, i) \quad (3.21)$$

şeklini alırlar. Normal denklemlerin çözülmesi amacıyla bir çok yöntem önerilmesine rağmen burada en çok kullanılan üç tanesi anlatılacaktır.

### 3.2.1. Doğrudan En Küçük Kareler Kestirimi

AR modeli,

$$s(k) = -\sum_{i=1}^p a_i s(k-i) + u(k) \quad (3.22)$$

şeklinde tanımlansın. Bu denklem vektör gösterilimi ile yazılırsa,

$$\underline{a}^T = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_p \end{bmatrix} \quad (3.23)$$

$$\underline{s}_k^T = \begin{bmatrix} s(k-1) & s(k-2) & \dots & s(k-p) \end{bmatrix}. \quad (3.24)$$

olmak üzere

$$s(k) = \underline{a}^T \underline{s}_k + u(k) \quad (3.25)$$

şeklini alır.  $\underline{s}^{*T}$ ,  $\underline{e}$ , vektörleri ve  $\underline{U}$  matrisi aşağıdaki gibi tanımlansın

$$\underline{s}^{*T} = \begin{bmatrix} s(k) & s(k-1) & \dots & s(k-N+1) \end{bmatrix} \quad (3.26)$$

$$\underline{e}^T = \begin{bmatrix} e_k & e_{k-1} & \dots & e_{k-N+1} \end{bmatrix} \quad (3.27)$$

$$\underline{U} = \begin{bmatrix} s(k-1) & s(k-2) & \dots & s(k-p) \\ s(k-2) & s(k-3) & \dots & s(k-p-1) \\ \vdots & \vdots & & \vdots \\ s(k-N) & s(k-N-1) & \dots & s(k-p-N+1) \end{bmatrix}. \quad (3.28)$$

Burada  $N$  gözlem aralığıdır. Öngörü hata vektörü  $\underline{e}$ , gözlem vektörü  $\underline{s}^*$ , gözlem matrisi  $\underline{U}$  ve kestirilen katsayı vektörü  $\underline{a}$  cinsinden yazılırsa

$$\underline{e} = \underline{s}^* - \underline{U} \hat{\underline{a}} \quad (3.29)$$

denklemini elde edilir.  $s(k)$  işaretinin  $N$  gözleme dayalı  $\hat{\underline{a}}$  katsayıları ile öngörülmesinin en küçük kareler maliyeti

$$J_N = (\underline{s}^* - \underline{U} \hat{\underline{a}})^T (\underline{s}^* - \underline{U} \hat{\underline{a}}) = \underline{e}^T \underline{e} \quad (3.30)$$

ile verilir. Maliyeti en küçük yapacak  $\hat{\underline{a}}$  katsayıları  $J_N$ 'nin türevi sıfıra eşitlenerek ve buradan  $\hat{\underline{a}}$  çözülerek aşağıdaki gibi bulunabilir,

$$\frac{\partial J_N}{\partial \hat{\underline{a}}} = 0 \quad (3.31)$$

$$\hat{\underline{a}} = (\underline{U}^T \underline{U})^{-1} \underline{U}^T \underline{s}^* \quad (3.32)$$

### 3.2.1.1. Kovaryans Ve Özilinti Yöntemi

Denklemin (3.32)'deki  $\underline{U}^T \underline{U}$  ve  $\underline{U}^T \underline{s}^*$  matris çarpımları yaklaşık olarak

$$\underline{W} = \begin{bmatrix} R(1,1) & R(1,2) & \dots & R(1,p) \\ R(2,1) & R(2,2) & \dots & R(2,p) \\ \vdots & \vdots & \ddots & \vdots \\ R(p,1) & R(p,2) & \dots & R(p,p) \end{bmatrix} \quad (3.33)$$

ve

$$\underline{R}^T(0) = \begin{bmatrix} R(0,1) & R(0,2) & \dots & R(0,p) \end{bmatrix} \quad (3.34)$$

şeklinde yazılabilir. Burada  $\underline{W}$  matrisi kovaryans matrisi olarak adlandırılır. Durağan olmayan bir rasgele süreç için AR model parametreleri, denklem (3.33) ve (3.34)'un denklem (3.32)'de yerine konmasıyla elde edilen

$$\hat{\underline{a}} = - \underline{W}^{-1} \underline{R}(0) \quad (3.35)$$

denklemleriyle elde edilebilir. Bu yöntem kovaryans yöntemi adı verilir. Eğer  $s(k)$  işareti geniş anlamda durağansa kovaryans matrisi özilinti matrisine dönüşür. Bu durumda denklem (3.33) ve (3.34)

$$\underline{R} = \begin{bmatrix} R(0) & R(1) & \dots & R(p-1) \\ R(1) & R(0) & \dots & R(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & \dots & R(0) \end{bmatrix} \quad (3.36)$$

$$\underline{r}^T = \begin{bmatrix} R(1) & R(2) & \dots & R(p) \end{bmatrix} \quad (3.37)$$

şeklini alır. Parametre vektörü ise

$$\hat{\underline{a}} = -\underline{R}^{-1}\underline{r} \quad (3.38)$$

formülü ile bulunabilir. Bu yöntem de özilinti yöntemi olarak adlandırılır.

### 3.2.1.2. Kısımsal İlinti Yöntemi

Gerek kovaryans, gerekse özilinti yönteminde matris tersinin hesaplanması ek bir hesaplama yükü getirmektedir. Bu yükten kurtulmak için denklem (3.36)'teki  $\underline{R}$  matrisinin bakışımli bir Toeplitz matrisi olma özelliğinden yararlanılan özyineli Durbin algoritması kullanılabilir. Aynı zamanda kısımsal ilinti yöntemi olarakta bilinen bu algoritma aşağıdaki denklem takımıyla verilir

$$E_0 = R(0), \quad (3.39)$$

$$k_i = - \left[ R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j) \right] / E_{i-1}, \quad (3.40)$$

$$a_i^{(1)} = k_i, \quad (3.41)$$

$$a_j^{(1)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)}, \quad 1 \leq j \leq i-1, \quad (3.42)$$

$$E_i = (1 - k_i^2) E_{i-1}, \quad 1 \leq i \leq p. \quad (3.43)$$

Denklemler sonucunda bulunan

$$a_j = a_j^{(p)}, \quad 1 \leq j \leq p \quad (3.44)$$

katsayıları istenen AR parametreleridir. Denklem (3.43)'teki  $E_i$ ,  $i$ . dereceden bir öngörücü için hatayı verir.  $k_i$  büyüklüklerine yansıma ya da kısmısal ilinti katsayıları denir.  $k_i$  ve  $E_i$ , süzgecin kararlı olması için

$$E(i) > 1, \quad 1 \leq i \leq p \quad (3.45)$$

$$|k_i| \leq 1, \quad 1 \leq i \leq p \quad (3.46)$$

denklemlerini sağlamalıdır.

### 3.2.2. Özyineli En Küçük Kareler Kestirimi

Doğrudan kestirim yöntemlerinde veriyi modelleyen parametrelerin kestirilebilmesi için tüm verinin bilinmesi gereklidir. Bazı durumlarda parametrelerin sürekli olarak kestirilmesi gerekebilir, yani, yeni veri geldikçe parametrelerin güncelleştirilmesi istenebilir. Bu gibi durumlarda kullanılan en etkili yöntem özyineli en az kareler yöntemidir. Yöntem, katsayı vektörünün ilklenmesi ve daha sonra her yeni veri geldiğinde bu vektörün gerçek kestirime doğru güncelleştirilmesi esasına dayanır. Özyineli en az kareler algoritması, ilk koşullar

$$P_{=0} = \delta^{-1} I \quad (3.47)$$

$$\hat{\underline{a}}_0 = \underline{0} \quad (3.48)$$

olmak üzere

$$\underline{t}_k = \frac{\underline{P}_{k-1} \underline{s}_k}{1 + \underline{s}_k^T \underline{P}_{k-1} \underline{s}_k} \quad (3.49)$$

$$\alpha_k = s(k) - \hat{\underline{a}}_{k-1} \underline{s}_k \quad (3.50)$$

$$\hat{\underline{a}}_k = \hat{\underline{a}}_{k-1} + \underline{t}_k \alpha_k \quad (3.51)$$

$$\underline{P}_k = \underline{P}_{k-1} - \underline{t}_k \underline{s}_k^T \underline{P}_{k-1}, \quad k=1,2,\dots \quad (3.52)$$

özyineli denklem takımıyla verilir. Bu denklemlerde  $\delta$  küçük pozitif bir sayı olup  $\underline{s}_k$

$$\underline{s}_k^T = \begin{bmatrix} s(k-1) & s(k-2) & s(k-p) \end{bmatrix} \quad (3.53)$$

ve  $\underline{P}_k$

$$\underline{P}_k = \left[ \sum_{i=1}^k \underline{s}_i \underline{s}_i^T \right]^{-1} \quad (3.54)$$

olarak tanımlanır. Denklemlerden de anlaşılacağı gibi yöntem matris tersi alma işlemini kullanmadığından

doğrudan kestirime oranla hesaplama karmaşıklığı açısından daha az karmaşıktır [21].

### 3.2.3. Kafes Yöntemi

Kafes yöntemi, ilinti matrisinin elemanlarının bulunması ve doğrusal bir denklem takımının çözülmesi basamaklarını birleştirerek özyineli bir denklem haline getiren bir yöntemdir. Derecesi altıdan daha büyük modeller için kafes yöntemi hızlı ve yakınsak çözümler sağlamaktadır. Kafes yönteminin esası

$$f_k^i = s_k - \sum_{n=1}^i a_n s_{k-n} \quad (3.55)$$

$$b_k^i = s_{k-i} - \sum_{n=1}^i c_n s_{k-i+n} \quad (3.56)$$

denklemleriyle verilen ileri ve geri öngörü hatalarının karelerinin toplamının minimum yapılmasına dayanır. Denklemdaki  $a_n$  ve  $c_n$  katsayıları ileri ve geri öngörücünün katsayılarıdır. Durağan işaretler için  $a_n = c_n$  olur. İleri ve geri öngörü hataları arasında

$$f_k^i = f_k^{i-1} - k_i b_{k-1}^{i-1} \quad (3.57)$$

$$b_k^i = b_{k-1}^{i-1} - k_i f_k^{i-1} \quad (3.58)$$

ilişkisi vardır. Denklem (3.57) ve (3.58)  $k_i$  katsayısı  $i$ . dereceden bir öngörünün kısmasal ilinti katsayısı olup Durbin algoritması yardımıyla bulunabilir. Bir başka yöntem de  $k_i$  katsayılarının

$$k_i = \frac{2 \sum_{k=0}^{N-1} f_k^{i-1} b_{k-1}^{i-1}}{\sum_{k=0}^{N-1} [f_k^{i-1}]^2 + \sum_{k=0}^{N-1} [b_{k-1}^{i-1}]^2} \quad (3.59)$$

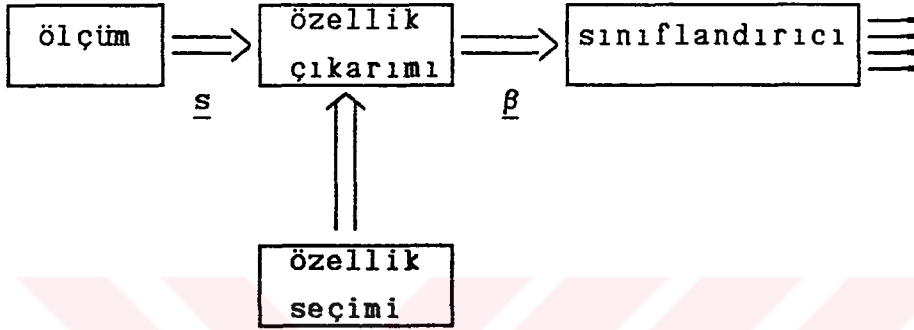
denkleminde bulunmasıdır [22]. Denklem (3.57), (3.58) ve (3.59),  $f_k^0 = b_k^0 = s_k$  alınarak özyineli olarak  $i=1,2,\dots,p$  için çözülürse  $\{k_i\}$ ,  $i=1, 2,\dots,p$  kısmasal ilinti katsayıları elde edilebilir. Daha sonra Durbin algoritmasında denklem (3.41) ve (3.42) kullanılarak  $\{a_i\}$ ,  $i=1,2,\dots,p$  parametreleri hesaplanabilir.

### 3.3. Örüntü Tanımanın Temelleri

Örüntü tanımanın ana amacı bilinmeyen bir örneği sonlu sayıdaki sınıflardan birine atamaktır. Sınıflandırma, hangi sınıftan oldukları bilinen etiketli örneklerden alınan bilgilere dayanır.

Örüntü tanımanın temel basamakları Şekil 3-1'de verilmiştir. Ölçüm bölümünde örüntüler ölçüm değerleri ile temsil edilirler. Yani sınıflandırılacak örüntüleri üreten fiziksel sistemi tanımak için gerekli ön bilgi ölçüm yoluyla elde edilir. Ancak bu yolla elde edilen bilgiler çoğu zaman fazlalıklar içerirler. Bu yüzden ölçümden sonra ilk yapılan işlem veri sıkıştırmadır.

Veri sıkıştırma gereksiz bilgi gözardı edilerek işareti daha az bilgi ile temsil etmek olasıdır. Veri sıkıştırma en çok kullanılan yöntem özellik çıkarımıdır.



Şekil 3-1. Bir örüntü tanıma sisteminin temel yapısı.

Özellik çıkarımı, ölçüm sonucu işaretin özünü temsil eden, fazlalıklardan arınmış parametre topluluğunun elde edilmesi olarak tanımlanabilir. Tipik özellik vektörleri arasında işaretin zaman serisi analizine dayanan ve işaretin yeniden elde edilmesinde kullanılabilen AR ve ARMA katsayıları olabileceği gibi, işaretin elde edilmesinde kullanılamayan fakat sınıflar arasında belirli bir ayrım sağlayan herhangi bir parametre de olabilir.

Sınıflandırıcı bloğu, özellikleri çıkarılan işareti bilinen sınıflardan birisine atar. Sınıflardan birisi "bilinmeyen" sınıfı olabilir. Bu durumda sınıflandırıcı, sınıflandıramadığı işaretleri bilinmeyen işaretler olarak kabul eder. Sınıfların sınıflandırıcıya tanıtılması işlemine "eğitim" işlemi denir. Ölçüm sonucu elde edilen  $\underline{s}$  vektörüne "örüntü vektörü", örüntü

vektöründen çıkarılan  $\underline{\beta}$  vektörüne "özellik vektörü" adı verilir. Sınıflandırıcı, sınıflandırmayı gerçekleştirmek için  $\underline{\beta}$  nın işlevleri olan "ayrım işlevleri" ni, ya da "karar işlevleri" ni kullanır.  $g(\underline{\beta})$  olarak gösterilen karar işlevleri  $\underline{\beta}$ 'nin tek değerli işlevleri olup skalarlardır. Genelde M sayıda sınıf için M tane karar işlevi kullanılır ve örüntü vektörü  $\underline{s}$ , özellik vektörü  $\underline{\beta}$  olan bir işaret eğer

$$g_i(\underline{\beta}) > g_j(\underline{\beta}), \quad i, j = 1, 2, \dots, M \quad (3.60)$$

$$i \neq j$$

eşitsizliği sağlanıyorsa i. sınıf olan  $w_i$  sınıfına atanır.  $g_i(\underline{\beta}) = p(\underline{\beta}|w_i)P(w_i)$  olduğunda karar kuralı

$$p(\underline{\beta}|w_i)P(w_i) > p(\underline{\beta}|w_j)P(w_j) \quad i, j = 1, 2, \dots, M \quad (3.61)$$

$$i \neq j$$

eşitsizliğine dönüşür. Bu karar kuralına "Bayes" karar kuralı adı verilir.  $g_i(\underline{\beta}) = p(\underline{\beta}|w_i)$  olduğunda ise karar kuralı

$$p(\underline{\beta}|w_i) > p(\underline{\beta}|w_j) \quad i, j = 1, 2, \dots, M \quad (3.62)$$

$$i \neq j$$

şekline dönüşür. Bu karar kuralına da "en büyük benzerlik" karar kuralı adı verilir [23].

### 3.3.1. k - En Yakın Komşu Yöntemi

Basit olması ve karmaşık karar verme sınırları kullanabilmesi nedeniyle pratikte en çok kullanılan sınıflandırma yöntemi en yakın komşu (EYK) yöntemidir. Bu yöntemde bilinmeyen bir işaret kendisine en yakın olan komşusunun sınıfına atanır.

EYK algoritmasının bir uygulaması olan k - en yakın komşu (k-EYK) yönteminde ise bilinmeyen işaretin hangi sınıfa ait olduğu, etrafındaki k sayıdaki komşu arasında yapılan oylama ile belirlenir.

Özellik vektörü  $\underline{\beta}$ 'nin özellik uzayında R ile belirlenen bölge içine düşmesi olasılığı

$$P = \int_R p(\underline{\beta}) d\underline{\beta} \approx p(\underline{\beta})V \quad (3.63)$$

denklemi ile bulunabilir. Bu denklemde V, R bölgesinin hacmidir. R bölgesinin çok küçük olduğu durumlarda denklem (3.63)'teki yaklaşıklık geçerli olmaktadır. Bu denklemden  $p(\underline{\beta})$  çekilirse

$$p(\underline{\beta}) = \frac{P}{V} \quad (3.64)$$

elde edilir. Elde bulunan n tane özellik vektöründen k tanesi R bölgesi içine düşüyorsa P ve  $p(\underline{\beta})$  olasılıkları

$$\hat{P} = \frac{k}{n} \quad (3.65)$$

$$\hat{p}(\underline{\beta}) = \frac{k/n}{V} \quad (3.66)$$

olarak kestirilebilirler. Denklem (3.66)'deki  $V$ 'nin seçilmesinde izlenen bir yol,  $k$ 'nın,  $n$ 'nin bir işlevi olarak belirlenmesidir. Eğer  $k = k_n = \sqrt{n}$  olarak kabul edilirse, seçilen hacim  $V_n$ ,  $V$ 'yi  $\underline{\beta}$ 'nin  $k_n$  tane komşusunu içerene kadar büyüterek belirlenebilir. Sınıflara bağlı koşullu olasılıklar ve sınıfların olasılıkları

$$\hat{p}(\underline{\beta}|w_i) = \frac{k_i/N_i}{V} \quad (3.67)$$

$$\hat{P}(w_i) = \frac{N_i}{N} \quad (3.68)$$

denklemleri ile kestirilebilir. Bu denklemlerde  $k_i$ , belirlenen  $V_n$  hacmindeki  $k_n$  komşudan  $w_i$  sınıfına ait olan komşu sayısı,  $N_i$ ,  $w_i$  sınıfına ait özellik vektörlerinin sayısı,  $N$  ise eğitim setindeki toplam özellik vektörlerinin sayısıdır. Yani  $M$  tane sınıftan oluşan bir eğitime setinde

$$N = \sum_{i=1}^M N_i \quad (3.69)$$

olmaktadır. Denklem (3.67) ve (3.68) Bayes ilişkisinden yararlanılarak Bayes kuralında yerlerine konursa

$$\hat{P}(w_i)\hat{p}(\underline{\beta}|w_i) > \hat{P}(w_j)\hat{p}(\underline{\beta}|w_j) , \forall j \quad (3.70)$$

$$\frac{N_i}{N} \frac{k_i/N_i}{V} > \frac{N_j}{N} \frac{k_j/N_j}{V} , \forall j \quad (3.71)$$

$$k_i > k_j , \forall j \quad (3.72)$$

karar kuralı elde edilir. Bu kurala göre  $\underline{\beta}$  vektörü, çevresindeki  $k_n$  özellik vektörü içinde en fazla vektöre sahip olan sınıfa atanır.

k-EYK yöntemi kolay olmasına karşın, tüm eğitim vektörlerinin saklanması zorunluluğu bellek gereksinimi doğurmaktadır. Bellek gereksinimini ve işlem zamanını azaltmak için Hart [25] tarafından EYK yöntemi için önerilen sıkıştırılmış en yakın komşu (SEYK) kuralı k-EYK yöntemine de uygulanabilir. Yöntemde önce sınıflandırmada kullanılacak dayanak kütüphane her sınıftan rasgele seçilen  $k+1$  sayıda özellik vektörü ile ilklenir. Diğer özellik vektörlerinden k-EYK kuralına göre doğru olarak sınıflandırılanlar geçici olarak gözardı edilir, yanlış olarak sınıflandırılanlar ise dayanak kütüphaneye eklenir. Bu işlem gözardı edilen vektörler için, ya tüm vektörler dayanak kütüphaneye atılana, ya da artık hiç bir vektör dayanak kütüphaneye atılmayana kadar tekrarlanır. Yani sonuçta gözardı edilebilecek hiç bir vektör kalmaz veya, gözardı edilen vektörlerin hepsi dayanak kütüphanedeki vektörler yardımıyla doğru olarak sınıflandırılır.

### 3.3.2. En Küçük Uzaklık Sınıflandırıcıları

Sınıfların kendi aralarında iyi kümелendiği ve birbirlerinden ayırtedilebildiği durumlarda sınıflandırma, sınıflandırılacak vektörün kümelere olan uzaklığı gözönüne alınarak yapılabilir. Bu sınıflandırmaya "en küçük uzaklık sınıflandırması" adı verilir.

$\underline{\beta}_1$  ve  $\underline{\beta}_2$  gibi iki özellik vektörü arasındaki benzerliğin bir ölçüsü olan uzaklık ölçüsünün genelde aşağıdaki özellikleri sağlaması istenir [23] :

- i)  $d(\underline{\beta}_1, \underline{\beta}_1) = d(\underline{\beta}_2, \underline{\beta}_2) = 0$ ,
- ii)  $d(\underline{\beta}_1, \underline{\beta}_2) > 0$ ,  $\underline{\beta}_1 \neq \underline{\beta}_2$ ,
- iii)  $d(\underline{\beta}_1, \underline{\beta}_2) = d(\underline{\beta}_2, \underline{\beta}_1)$ ,
- iv)  $d(\underline{\beta}_1, \underline{\beta}_2) + d(\underline{\beta}_2, \underline{\beta}_3) \geq d(\underline{\beta}_1, \underline{\beta}_3)$ .

i. ve ii. özellikler pozitif tanımlılık özelliği, iii. özellik bakışlılık özelliği, iv. özellik ise üçgen eşitsizliği olarak adlandırılır.

Örüntü tanımada kullanılan en basit uzaklık ölçüsü Euclid uzaklık ölçüsüdür ve

$$\underline{\beta}_1^T = [\beta_{11} \ \beta_{12} \ \dots \ \beta_{1p}] \quad (3.73)$$

$$\underline{\beta}_2^T = [\beta_{21} \ \beta_{22} \ \dots \ \beta_{2p}] \quad (3.74)$$

olmak üzere

$$d_E(\underline{\beta}_1, \underline{\beta}_2) = \left[ \sum_{k=1}^p (\beta_{1k} - \beta_{2k})^2 \right]^{1/2} \quad (3.75)$$

olarak verilir. Denklem (3.75), M sınıflık bir özellik uzayı için vektörel biçimde ve en genel haliyle yazılırsa,  $x$  bilinmeyen örüntüyü,  $w_i$  i. sınıfı göstermek üzere

$$d_E^2(x, w_i) = (\underline{\beta}_x - \underline{\mu}_i)^T (\underline{\beta}_x - \underline{\mu}_i), \quad i = 1, 2, \dots, M \quad (3.76)$$

şeklini alır. Burada,  $\underline{\beta}_x$  sınıflandırılacak vektör,  $\underline{\mu}_i$  ise i. kümeye ait dayanak vektörlerinin ortalamasıdır. Yani,  $N_i$ , i. kümeye ait özellik vektörlerinin sayısı ise,

$$\underline{\mu}_i = \frac{1}{N_i} \cdot \sum_{j=1}^{N_i} \underline{\beta}_{i,j}, \quad i = 1, 2, \dots, M \quad (3.77)$$

olmaktadır. Sınıflandırıcı bu uzaklığı en küçük yapan dayanak vektörün ait olduğu sınıfa karar verir.

Denklem (3.76)'den de anlaşılabacağı gibi Euclid uzaklığı tüm sınıflara aynı önemi vermektedir. Eğer sınıfların istatistiksel durumları hakkında herhangi bir ön bilgi biliniyorsa, bu ön bilgi kullanılarak özellik vektörlerine bir ağırlıklandırma uygulanabilir. Örneğin, ön bilgi, bazı özellik vektörlerinin yüksek değişimliliğe sahip oldukları ise, uzaklık hesaplanırken bu vektörler daha az güvenilir olarak ele alınabilirler. Ağırlıklandırma kovaryans matrisiyle yapıldığında denklem (3.76),

$$\Sigma_i = \frac{1}{N_i} \sum_{j=1}^{N_i} (\underline{\beta}_j - \underline{\mu}_i)(\underline{\beta}_j - \underline{\mu}_i)^T \quad (3.78)$$

i. kümenin kovaryans matrisi olmak üzere

$$d_M^2(x, w_i) = (\underline{\beta}_x - \underline{\mu}_i)^T \Sigma_i^{-1} (\underline{\beta}_x - \underline{\mu}_i), \quad i = 1, 2, \dots, M \quad (3.79)$$

şeklini alır. Elde edilen bu uzaklık ölçüsüne Mahalanobis uzaklık ölçüsü adı verilir.

Örüntü tanımada sıkça kullanılan bir başka uzaklık ölçüsü de Itakura-Saito (I-S) ölçüsüdür. Özellikle AR olarak modellenen işaretlerin sınıflandırmasında kullanılan I-S ölçüsü

$$d_{IS} = \int_{-\pi}^{\pi} \left[ \frac{|X(w)|^2}{|A(w)|^2} + \log \frac{|A(w)|^2}{|X(w)|^2} - 1 \right] \frac{dw}{2\pi} \quad (3.80)$$

denklemleri ile verilir [26]. Denklemlerde  $|X(w)|^2$  işaretin kısa süreli spektrumu,  $|A(w)|^2$  ise bir AR model spektrumudur. AR ile modellenen bir işaretin spektrumu

$$|A(w)|^2 = \frac{\sigma^2}{|1 + a_1 e^{-jw} + a_2 e^{-j2w} + \dots + a_p e^{-jpw}|^2} \quad (3.81)$$

şeklinde bulunabilir. Burada  $\sigma$  modelin kazancı,  $a_i$  ise p. dereceden bir AR modelinin i. doğrusal öngörü (LPC)

katsayısıdır.

I-S ölçüsü bir işaretin spektrumunu dayanak AR modelleri ile karşılaştırmaktadır. İki AR model arasındaki uzaklığı ölçen ve Itakura tarafından önerilen Itakura (I) ölçüsü, I-S ölçüsünün bir uzantısı olup A ve B gibi iki AR model arasındaki uzaklık

$$d_I(A,B) = \log \int_{-\pi}^{\pi} \frac{|B(w)|^2}{|A(w)|^2} \frac{dw}{2\pi} \quad (3.82)$$

veya

$$d_I(A,B) = \log \int_{-\pi}^{\pi} \frac{|1+b_1 e^{-jw} + b_2 e^{-j2w} + \dots + b_p e^{-jpw}|^2}{|1+a_1 e^{-jw} + a_2 e^{-j2w} + \dots + a_p e^{-jpw}|^2} \frac{dw}{2\pi} \quad (3.83)$$

denklemleri ile verilir. Itakura ölçüsünün basite indirgenmiş hali

$$d_I(A,B) = \log \left( \frac{\underline{b}^T \underline{R}_A \underline{b}}{\sigma_A^2} \right) \quad (3.84)$$

şeklindedir. Denklemdaki  $\underline{b}^T = [1, b_1, b_2, \dots, b_p]$  vektörü B modelinin doğrusal öngörü katsayıları,  $\underline{R}_A$  matrisi A modelinin Toeplitz özelliği gösteren özilinti matrisi,  $\sigma_A$  ise modelin kazancıdır. Bu uzaklık

ölçüsünden elde edilen değer, bilinmeyen işaretin  $\underline{b}^T$  ile verilen dayanak AR süzgecinden geçirildiğinde elde edilen toplam karesel hata ile, aynı işaretin LPC analizi sırasında elde edilen toplam karesel hatanın logaritmik oranlarına eşittir. Yani

$$d_I(A,B) = \log \frac{\sum_{k=1}^N \underline{b}^T \underline{s}_{-k} \underline{s}_{-k}^T \underline{b}}{\sum_{k=1}^N \underline{a}^T \underline{s}_{-k} \underline{s}_{-k}^T \underline{a}} \quad (3.85)$$

veya

$$d_I(A,B) = \log \frac{\sum_{k=1}^N e_B^2(k)}{\sum_{k=1}^N e_A^2(k)} \quad (3.86)$$

denklemleri geçerlidir. Bu denklemlerde  $\underline{s}_{-k}$  (px1)'lik gözlem vektörü, N çerçeve uzunluğu,  $\underline{b}^T$  dayanak LPC katsayı vektörü,  $\underline{a}^T$  sınıflandırılacak işaretin LPC vektörüdür.

Itakura ölçüsü bir çok uygulamada başarıyla kullanılmasına karşın, diğer bir çok uzaklık ölçüsü gibi gürültüye çok duyarlıdır. Bu duyarlılık özellikle dayanak kütüphanenin gürültüsüz işaretlerle eğitildiği, buna karşın sınıflandırılacak işaretlerin gürültülü oldukları durumda ortaya çıkar. Bu duyarlılığı azaltıp başarıyı arttırmak için Soong ve Sondhi [25] tarafından önerilen frekans ağırlıklandırılmalı Itakura ölçüsü

$$\frac{1}{|A_a(w)|^2} = \frac{1}{|1 + \alpha a_1 e^{-jw} + \alpha^2 a_2 e^{-j2w} + \dots + \alpha^p a_p e^{-jpw}|^2} \quad (3.87)$$

olmak üzere

$$d_{WI}(A,B) = \log \int_{-\pi}^{\pi} \frac{1}{|A_a(w)|^2} \frac{|B(w)|^2}{|A(w)|^2} \frac{dw}{2\pi} \quad (3.88)$$

denklemleri ile verilir. Denklemde  $\alpha$ , 0 ile 1 arasında bir sayıdır. Ağırlıklandırılmış Itakura ölçüsü, işaret gürültü oranına (IGO) göre, frekans bölgesinde LPC polinomunun kutuplarına karşılık gelen dorukların band genişliklerini büyütür. IGO'nun fazla olduğu durumlarda  $\alpha$  sıfıra, az olduğu durumlarda  $\alpha$  birine yakın alınır.

Gerek I-S, gerekse I ölçüleri oransal olduklarından, bir uzaklık ölçüsünün sahip olması gereken bakışsımlılık özelliğini sağlamazlar. Bakışsımlılık özelliği bakışsımlı olmayan iki uzaklığın basit geometrik, ya da aritmetik ortalamaları alınarak sağlanabilir. A ve B gibi iki AR model arasındaki I-S ölçüsünün bakışsımlı şekli

$$d_{\cosh} = \frac{1}{2} (d_{IS}(A,B) + d_{IS}(B,A)) \quad (3.89)$$

olarak elde edilebilir. "cosh" uzaklığı olarak adlandırılan bu uzaklık ölçüsü genelde desibel cinsinden

ifade edilir. Bu amaçla tanımlanan

$$\cosh(D) - 1 = d_{\cosh} \quad (3.90)$$

denkleminde  $D$  çekilirse  $\cosh$  uzaklığının desibel cinsinden ifadesi olan

$$D = \ln \left( 1 + d_{\cosh} + \sqrt{d_{\cosh}(2 + d_{\cosh})} \right) \quad (3.91)$$

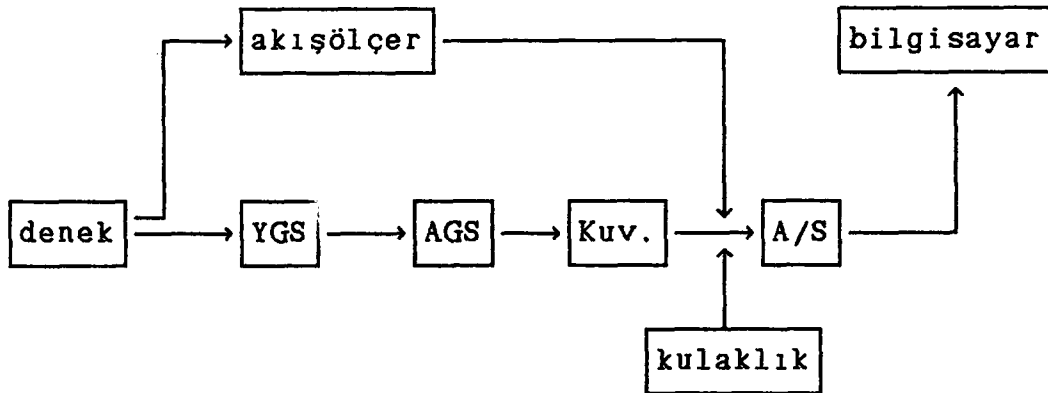
elde edilir [7].

## BÖLÜM 4. SOLUNUM SESLERİNİN ANALİZİ VE SINIFLANDIRILMASI

### 4.1. Ölçüm Düzeni

Solunum seslerinin analizi ve sınıflandırılması amacıyla veri toplamakta kullanılan sistemin öbek çizeneği Şekil 4-1' de verilmiştir.

Şekilden de görüldüğü gibi denekten alınan solunum sesleri önce, kesim frekansı 80 Hz olan bir yüksekgeçiren süzgeçten (YGS) geçirilmiştir. YGS'nin kullanılmasındaki amaç solunum seslerine karışan kalp seslerinin ve kas sürtünmesi gürültülerinin elenmesidir. YGS olarak doğrusal evreli 4. dereceden bir Butterworth süzgeci tasarlanmıştır. YGS çıkışındaki işaret, kesim frekansı 2000 Hz olan bir örtüşme önleyici alçak-



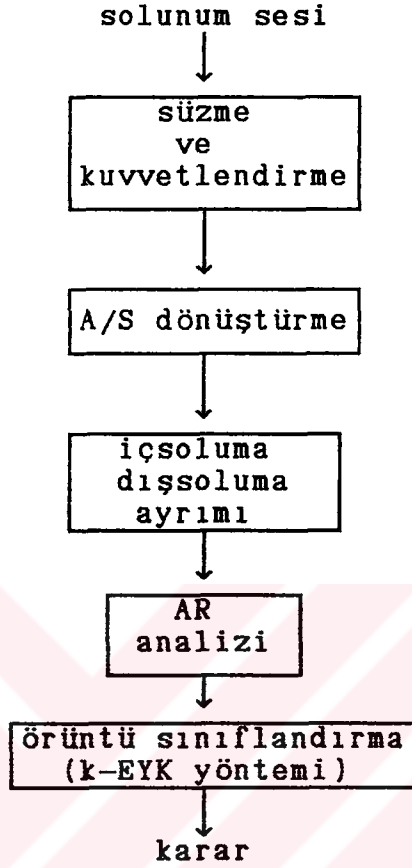
Şekil 4-1. Solunum sesleri için ölçüm ve kayıt düzeni.

geçiren süzgeç (AGS) ile süzölmüştür. AGS olarak kazancı 24 dB olan, doğrusal evre yanıtı, 8. dereceden bir Bessel süzgeç tasarlanmıştır. Bant genişliği 80-2000 Hz'e sınırlı analog işaret sayısala dönüştürülmeden önce kazancı 36 dB olan bir önkuvvetlendirici ile kuvvetlendirilmiş, böylece toplam kazanç 60 dB'e çıkarılmıştır. AGS, YGS ve kuvvetlendiricinin devre şemaları EK-A' da verilmiştir. Önişlemesi yapılan işaret, akış evrelerinin sezilmesi amacıyla Fleisch tipi bir akışölçer ile denekten alınan akış işareti ile beraber 12 bitlik analog-sayısal (A/S) çevirici kartıyla (MACAdiosII-Jr) 5000 Hz hızında örneklenmiştir. Solunum seslerindeki anlamlı bant genişliğinin yaklaşık 1500 Hz olduğu gözönüne alınırsa, 5000 Hz'lik bir örnekleme hızının yeterli olduğu görülmektedir. Kayıt sırasında, yapılan kayıttın niteliğini kontrol etmek amacıyla stereo kulaklık (Philips SBC 484) kullanılmıştır.

Sayısal işarete dönüştürülen işaretler, saklama ve analiz amacıyla doğrudan bilgisayara (Macintosh II-cx) aktarılmıştır. Seslerin kaydında kullanılan, 20 - 18000 Hz bant genişliğine sahip kondansatör mikrofonlar (Panasonic P9930) sürtünme sesini önlemek amacıyla plastik kılıfların içine konmuş ve tıbbi bant kullanılarak kayıt yerine yapıştırılmıştır. Bu türden bir hava bağlaşımının, mikrofonta uygulanan duruk kuvvetin ölçümlere olan etkisini ortadan kaldırdığı anlaşılmıştır [6]. Plastik kılıfın çizeneği Ek-A' da verilmiştir.

#### 4.2. Analiz Ve Sınıflandırma Yöntemi

Solunum seslerinin analizi ve sınıflandırmasında izlenen yol Şekil 4-2'de gösterilmiştir.



Şekil 4-2. Solunum seslerinin analizinde ve sınıflandırılmasında izlenen yol.

Solunum işaretlerinin içsoluma ve dışsoluma noktalarında duragansızlıklar göstermesi, AR modellemeye dayalı analizde soluk çevrimlerine göre eşzamanlamayı gerekli kılmaktadır. Her iki evreye göre farklı modeller oluşturulması, farklı evrelerde oluşabilecek anormal seslerin sezilmesinde de kullanılabilir. Soluma evrelerinin ayırımı, solunum sesi ile birlikte kaydedilen akış işaretinin eşik yardımıyla sıfırlanması yoluyla yapılmıştır. Bu amaçla akış işareti ikiye ayrılmış ve iki akış işaretindeki maksimum genlikler bulunarak ortalamaları alınmıştır. Yapılan deneylerde

bir akış evresinin genelde maksimum noktasının yaklaşık %12'sinde sona erdiği gözlenmiştir. Buna dayanarak bulunan ortalama maksimum değerin %12'si eşik değeri olarak belirlenmiş ve pozitif ve negatif eksendeki eşik değerleri arasında kalan akış işareti sıfırlanmıştır. Daha sonra, eşikten geçirilmiş işaretin sıfırdan pozitif eksene çıktığı ve tekrar sıfıra düştüğü iki nokta arası dışsoluma, sıfırdan negatif eksene çıktığı ve sıfıra düştüğü iki nokta arası içsoluma olarak belirlenmiştir. Bir içsoluma, ya da dışsolumanın yaklaşık 1 saniye sürdüğü gözönüne alınarak iki nokta arasının 0.2 saniyeden (1000 örnek) fazla olduğu aralıklar yalancı evreler olarak kabul edilip gözardı edilmişlerdir.

Solunum seslerinin analizinde, seslerin beyaz gürültüyle sürülen tüm-kutup sistemler olarak modellenebileceği varsayılmış, her sınıf ve içsoluma-dışsoluma olmak üzere, her akış aralığı için ayrı bir özbağlanımlı model oluşturulmuştur. Oluşturulan model

$$s_f^{(i)}(n) = \sum_{k=1}^p a_{k,f}^{(i)} s_f^{(i)}(n-k) + e_f^{(i)}(n) ; n = 1, \dots, N$$

$f = \begin{matrix} \text{içsoluma,} \\ \text{dışsoluma} \end{matrix}$

(4.1)

denklemleriyle verilebilir. Bu denklemde  $s_f^{(i)}(n)$ , n. zamanda gözlenen solunum işareti,  $a_{k,f}^{(i)}$  k. LPC katsayısı, p model derecesi, N çerçeve uzunluğu,  $e_f^{(i)}(n)$  öngörü hatasıdır. Denklemde i indisi işaretin sınıfını, f indisi de akış aralığını belirtmektedir. Solunum seslerinin 50 ms'lik sürede durağan oldukları varsayılmış ve çerçeve uzunluğu  $N = 256$  alınmıştır. Analiz sırasında, kestirilen parametrelerdeki aşırı sapmaları önlemek amacıyla, arka arkaya gelen

çerçeveler arasında 64 örneklik (% 25) bir örtüşmeye izin verilmiş ve her çerçeve Hamming penceresiyle ağırlıklandırılmıştır. AR parametrelerinin hesaplanmasında Durbin algoritması kullanılmıştır.

Model derecesi  $p$ 'nin seçiminde Akaike ölçütünden yararlanılmıştır. Akaike ölçütüne göre, AR modellerde model derecesi

$$AIC(p) = q N \ln(E_p) + 2p \quad (4.2)$$

işlevini enküçükleyecek şekilde seçilmektedir. Denklemden  $E_p$  en küçük karesel ortalama hata,  $q$  ise pencereleme etkisini karşılayan bir katsayıdır. Dikdörtgen pencere için  $q=1$ , Hamming penceresi için  $q=0.4$  alınabilir [20]. Denklem (4.2)'yi en küçük yapan  $p$  değeri genelde  $1 \leq p \leq 3\sqrt{N}$  aralığında aranmaktadır.

Optimum model derecesinin bulunmasında şu yol izlenmiştir : her denegin bir ölçüm bölgesi için, tam bir soluk çevrimine karşı düşen bölgede, her bir çerçeve için optimum dereceler hesaplanmış, daha sonra bunların ortalaması alınmıştır. Sonuçlar Tablo 4-1' de verilmiştir. Daha sonra bulunan tüm optimum model derecelerinin ortalaması alınarak ortalama optimum AR model derecesi  $p_{opt} = 11$  olarak bulunmuştur. Ancak yapılan deneylerde sekizden yüksek model derecelerinde sınıflandırıcının başarımının değişmediği gözlemlendiğinden  $p = 8$  olarak alınmıştır.

Akış evrelerine göre ayrılan ve AR analizi yapılan solunum işaretlerinin sınıflandırılmasında genelleştirilmiş k-EYK yöntemi kullanılmıştır. Genelleştirilmiş

Tablo 4-1. Akaike ölçütü sonuçları

|         | Soluk Borusu | Sol Taban | Sağ Taban |
|---------|--------------|-----------|-----------|
| Denek A | 14           | 15        | 9         |
| Denek B | 8            | 10        | 9         |
| Denek C | 10           | 10        | 12        |
| Denek D | 9            | 9         | 13        |
| Denek E | 10           | 14        | 18        |
| Denek F | 13           | 12        | 11        |
| Denek G | 13           | 11        | 11        |
| Denek H | 8            | 13        | 11        |
| Denek I | 13           | 12        | 10        |
| Denek J | 16           | 9         | 8         |

Denek A,D,G : Sağlıklı

Denek B,E,H,I : Zatürree

Denek C : Akciğer zarı yangısı

Denek F : Amfizem

Denek J : Astım

k-EYK yönteminde bilinmeyen bir işaret, her bir sınıftaki k en yakın komşusuna olan toplam uzaklığı gözönüne alınarak sınıflandırılır. İşaret en az toplam uzaklığı veren sınıfa atanır. Bu çalışmada özellik vektörü olarak AR analizinden elde edilen LPC katsayıları, uzaklık ölçüsü olarak da Itakura ölçüsü kullanılmıştır. m. çerçeve ile i.sınıftaki l. dayanak özellik vektörü  $\mathbf{b}_{-i,1}^T$  arasındaki Itakura uzaklığı

$$d_{m,1}^{(i)} = \log \left( \frac{\sum_{n=1}^N \underline{b}_{i,1}^T \underline{s}_n \underline{s}_n^T \underline{b}_{i,1}}{\sum_{n=1}^N \underline{a}_m^T \underline{s}_n \underline{s}_n^T \underline{a}_m} \right) \quad (4.3)$$

denklemleriyle verilir. Bu denklemlerde  $\underline{s}_n$ , (px1) elemanlık gözlem vektörü,  $\underline{a}_m^T$ , m. çerçevenin AR analizinden elde edilen LPC vektörüdür. Denklem (4.3), denklem (3.86)'dan yararlanarak

$$d_{m,1}^{(i)} = \log \left( \frac{\sum_{n=1}^N e_{i,1}^2(n)}{\sum_{n=1}^N e_m^2(n)} \right) \quad (4.4)$$

şeklinde yazılabilir. Paydaki toplam, m. çerçevenin, katsayıları  $\underline{b}_{i,1}^T$  olan bir AR süzgeçten geçirilmesi sonucunda ortaya çıkan toplam karesel hata, paydadaki toplam ise aynı çerçevenin LPC analizi sırasında elde edilen toplam karesel hatadır. Katsayıların Durbin algoritması ile hesaplanması sırasında paydadaki büyüklük de hesaplandığından payda için ayrıca bir işleme gerek yoktur.

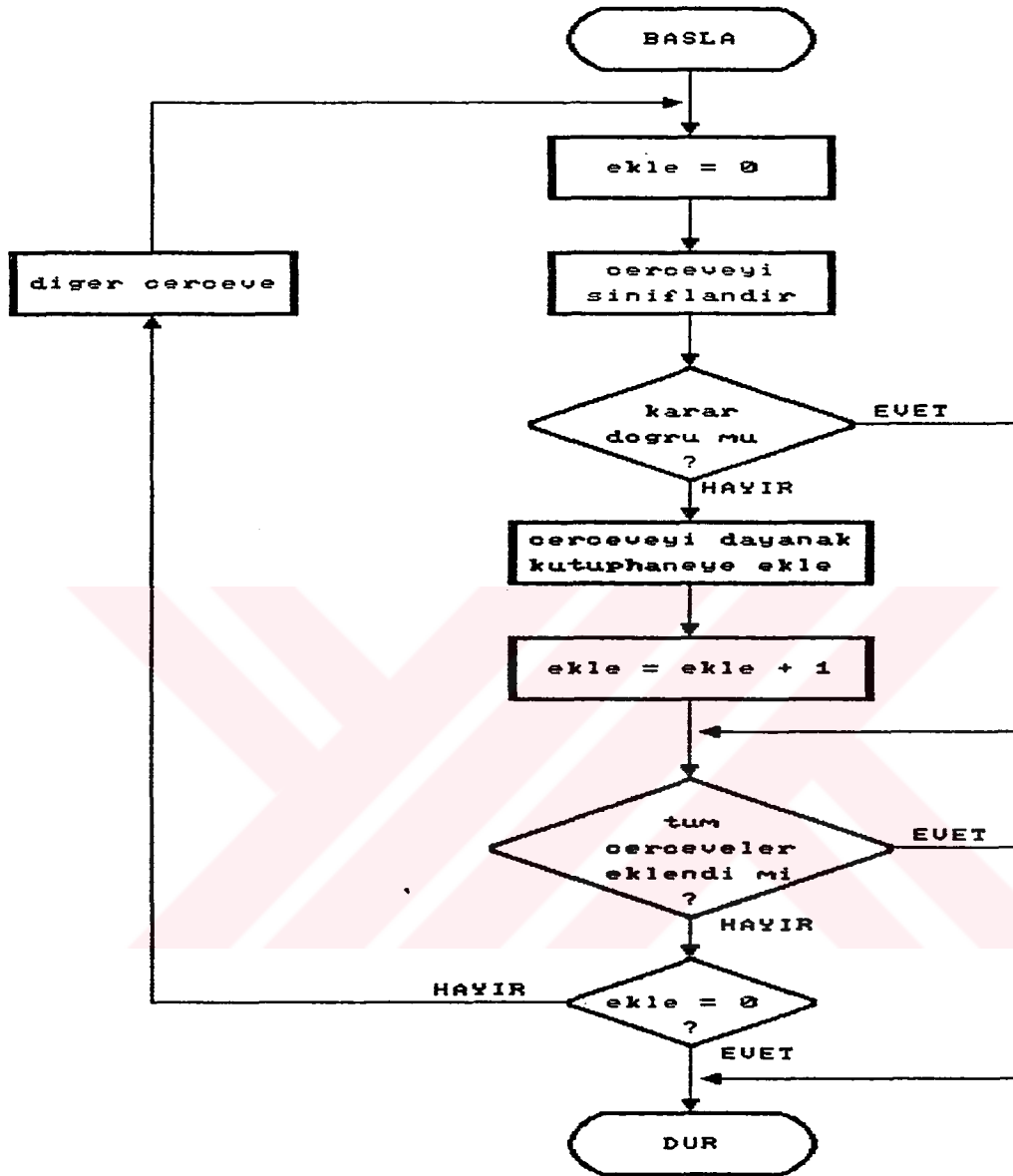
Solunum seslerinin k-EYK ile sınıflandırılması iki aşamada gerçekleşmektedir:

Dayanak Kütüphanenin Oluşturulması Sınıflandırmanın ilk aşaması olan dayanak kütüphanenin oluşturulması iki basamakta gerçekleşir. İlk basamak olan ilkleme

basamağında dayanak kütüphane, her sınıftan rasgele seçilen soluk seslerinden bulunan  $k+1$  sayıda LPC vektörü ile ilklenir. Dayanak kütüphanenin oluşturulmasında soluk evreleri de dikkate alındığından,  $M$  sayıda sınıf için ilklenmiş kütüphanedeki vektör sayısının  $2M(k+1)$  olacağı görülebilir.

Dayanak kütüphanenin oluşturulmasındaki ikinci basamak sınıflandırıcının eğitilmesidir.  $k$ -EYK yöntemi, bilinmeyen test işareti ile dayanak kütüphanedeki her bir vektör arasındaki uzaklığın hesaplanmasını gerektirdiğinden, işlem sayısını ve bellek gereksinimini azaltmak için, sınıflandırıcının tüm örneklerle eğitilmesi yerine, Hart tarafından [24] önerilen sıkıştırılmış  $k$ -EYK kuralı kullanılmış ve sınıflandırıcı tüm eğitim setinin özelliklerini kapsayan bir alt eğitim seti ile eğitilmiştir. Sıkıştırılmış  $k$ -EYK kuralı şu temele dayanır : sınıflandırıcı işe ilklenmiş dayanak kütüphane ile başlar ve hangi sınıfa ait olduğu bilinen eğitim örneği (çerçeve) eğer bu kütüphane ile doğru olarak sınıflandırılıyorsa, bu çerçeveden bulunan LPC vektörü geçici olarak gözardı edilir, eğer çerçeve doğru olarak sınıflandırılmıyorsa, LPC vektörü dayanak kütüphaneye eklenir. Bu işlem, gözardı edilmiş vektörler için, tüm vektörlerin kütüphaneye katılmasına, ya da artık hiçbir vektörün kütüphaneye katılamamasına dek tekrarlanır. Dayanak kütüphanenin sıkıştırılması işlemi Şekil 4-3'de verilen akış çizeneğinde gösterilmiştir.

Dayanak kütüphanenin oluşturulmasında kullanılan veri tabanı "çift-bağlı liste" veri yapısında olup bir veri hücresi, çerçevenin ait olduğu soluk evresi, denegin sınıfı, çerçeveden bulunan LPC vektörü ve bir önceki ile bir sonraki hücrenin konumu bilgilerini içermektedir.



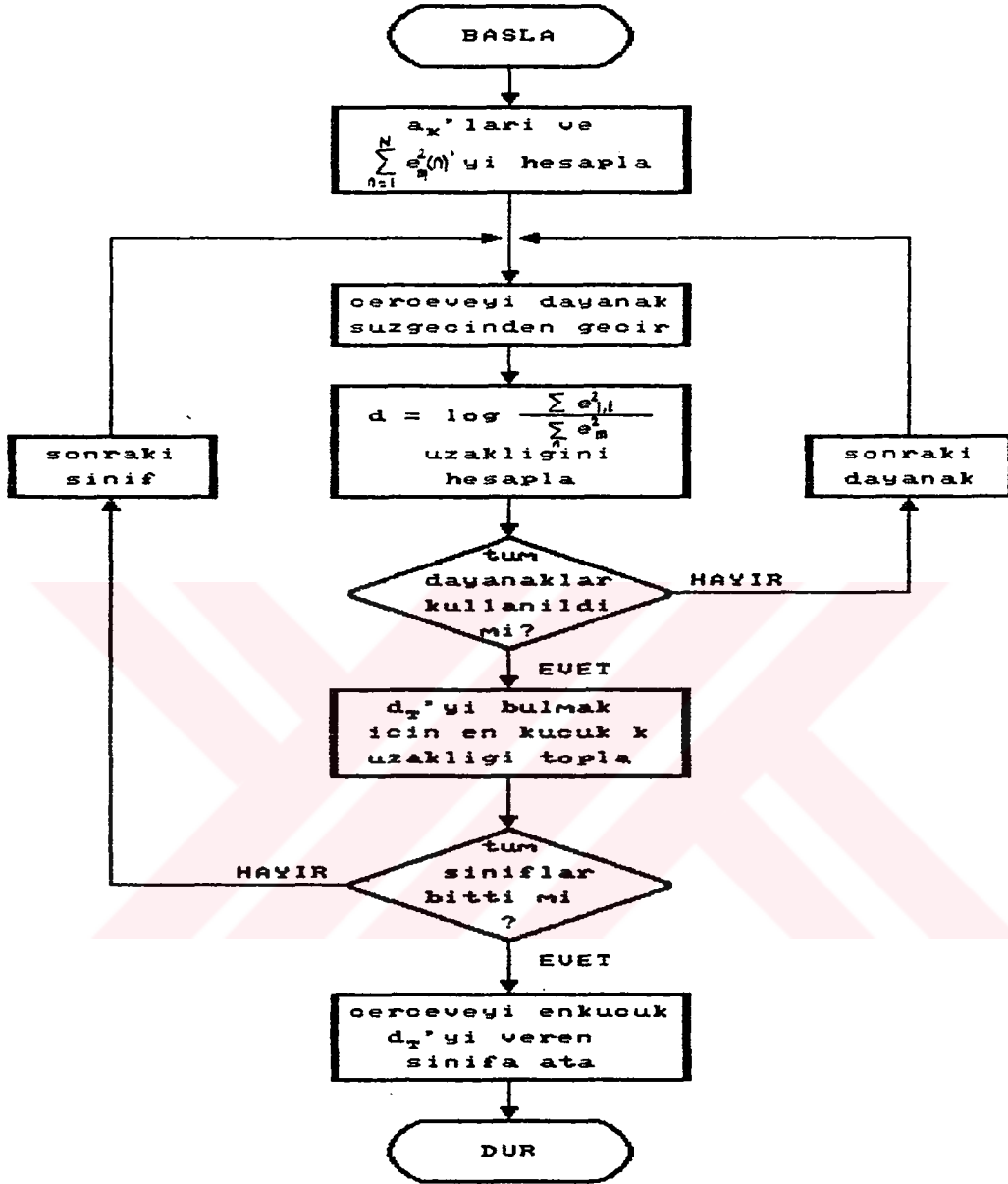
Şekil 4-3. Dayanak kütüphanenin sıkıştırılması.

Sınıflandırma Sınıflandırma aşamasının ilk basamağı soluk evrelerinin akış işareti yardımıyla ayrılmasıdır. Daha sonra herbir soluk evresindeki işaret, % 25 örtüşen, 256 örneklilik çerçevelere bölünmüş

ve bu çerçevelerin dayanak kütüphanedeki karşı gelen soluk evrelerine ait vektörlere olan Itakura uzaklığı hesaplanmıştır. Çerçevenin bir sınıfa olan uzaklığı, çerçevenin bu sınıfa ait vektörlere olan uzaklıklarının en küçük k tanesinin toplamı olarak alınmıştır. Sonuçta çerçeve, M tane sınıf içinden en küçük uzaklığı veren sınıfa atanmıştır. Bir çerçevenin sınıflandırılmasındaki basamaklar Şekil 4-4'de verilen akış çizeneğinde gösterilmiştir. Deneyin hangi sınıfa ait olduğuna ise, tüm çerçeveler için verilen kararların yüzde olarak karşılaştırılması yoluyla karar verilmiştir. Yani, iki sınıflı durum için, bir deneyin içsoluma işaretlerinden bulunan P tane çerçeveden Q tanesi A sınıfına,  $(P-Q)$  tanesi B sınıfına atanmışsa, deneyin, içsolumada,  $100 \times Q/P$  olasılıkla A sınıfına,  $100 \times (P-Q)/P$  olasılıkla B sınıfına ait olduğuna karar verilmiştir. Benzer işlemler dışsoluma için de geçerlidir.

#### 4.3. Deneysel Sonuçlar

Analiz ve sınıflandırmada kullanılan solunum sesleri kayıtları Cerrahpaşa hastahanesi, solunum laboratuvarında ve Çapa hastahanesi, çocuk hastalıkları bölümünde gerçekleştirilmiştir. Olanaklar el verdiği ölçüde farklı hastalıklardan örnekler alınmasına çalışılmış, ancak kayıtların yapıldığı zamanın hastalıkların çoğaldığı zamana denk gelmemesi yüzünden sınıflandırma işlemleri, diğerlerine göre daha çok sayıda kayıt alınabilen sağlıklı ve zatürreeli denekler üzerinde yürütülmüştür. Deneklerin yaşları 7 ile 67 arasında değişmektedir. Kayıtlar bulunabilinen en sessiz odada gerçekleştirilmiştir. Kayıt bölgeleri, soluk borusu, sağ taban ve sol taban olarak belirlenmiş, bu bölgelerden 12.8 saniye boyunca kayıt alınmıştır. Bu süre yaklaşık

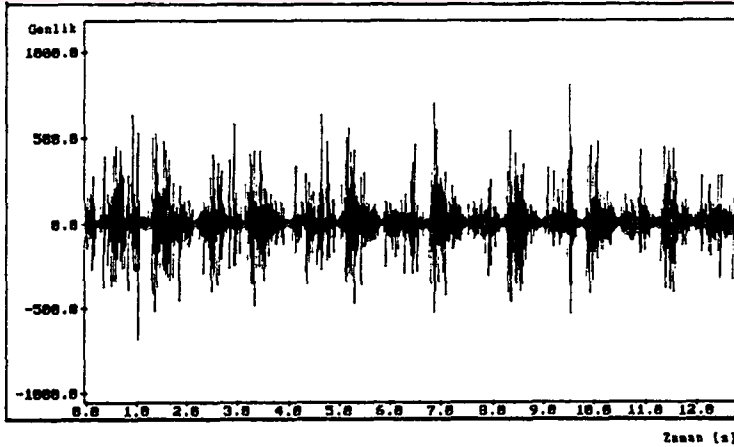


Şekil 4-4. Bir çerçevenin sınıflandırılmasında izlenen yol.

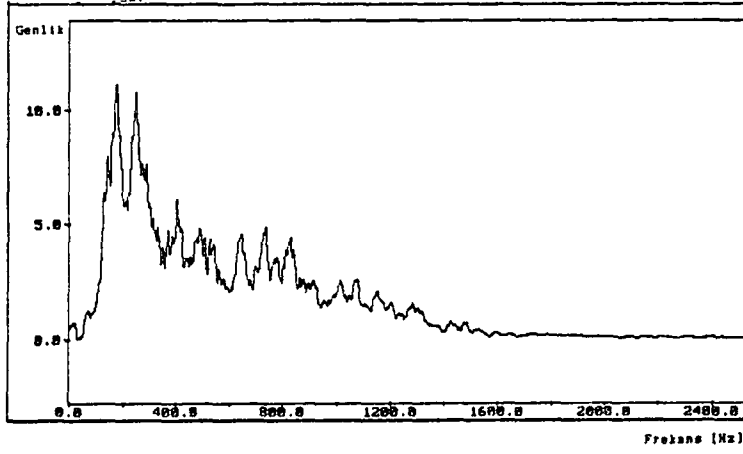
5-6 soluk çevrimine karşı düşmektedir. Kayıt sırasında deneklerden belirli bir akış hızında solumaları

istenmemiş, ancak solunum seslerinin duyulabileceği akış eşiğini aşmalarına dikkat edilmiştir. Kayıtlar, hem kayıt sırasında kulaklıkla, hem de kayıt sonrasında, geliştirilen yazılım sayesinde dinlenmiş ve bilgisayar ekranına çizilen dalga biçimlerine bakılarak hem görsel, hem de işitsel olarak irdelenmiştir.

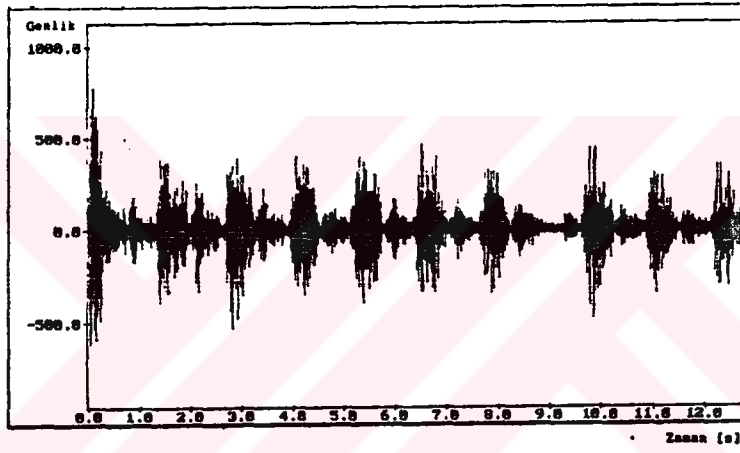
Zatürreeli bir hastanın sol tabanından alınan solunum sesi dalga biçimi ve ortalamaıyla yumuşatılmış içsoluma FFT spektrumu Şekil 4-5a ve 4-5b'de, sağlıklı bir denegın sol tabanından alınan solunum sesi dalga biçimi ile içsoluma FFT spektrumu 4-5c ve 4-5d'de verilmiştir. Zatürreenin belirtilerinden biri olan çıtırtı sesleri zaman dalga biçiminde açıkça görölmektedir. Ayrıca sağlıklı sesin spektrumu 0-500 Hz arasında sınırlı kalırken, zatürreeli denegın spektrumu yaklaşık 1600 Hz'e kadar yüksek frekans bileşenleri içermektedir. Bunun nedeni, akciğer yangılanmasının göğüs kafesinin aktarganlık işlevini değıştirmesi olarak gösterilebilir.



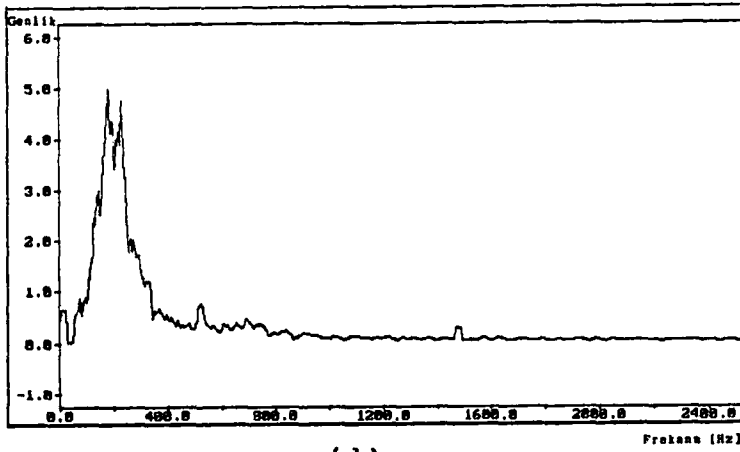
(a)



(b)



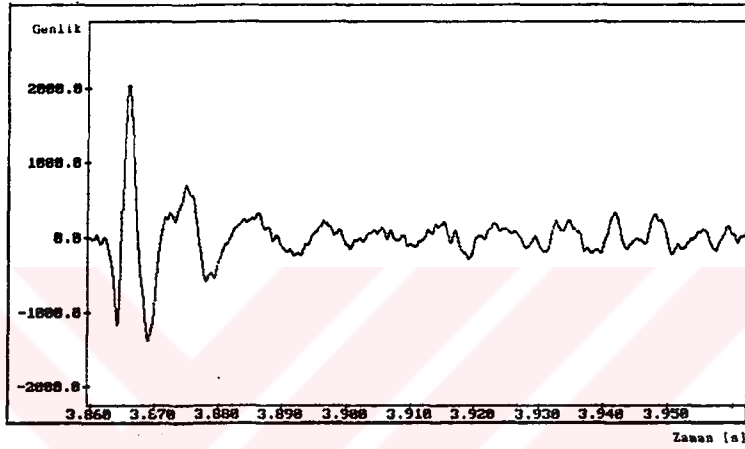
(c)



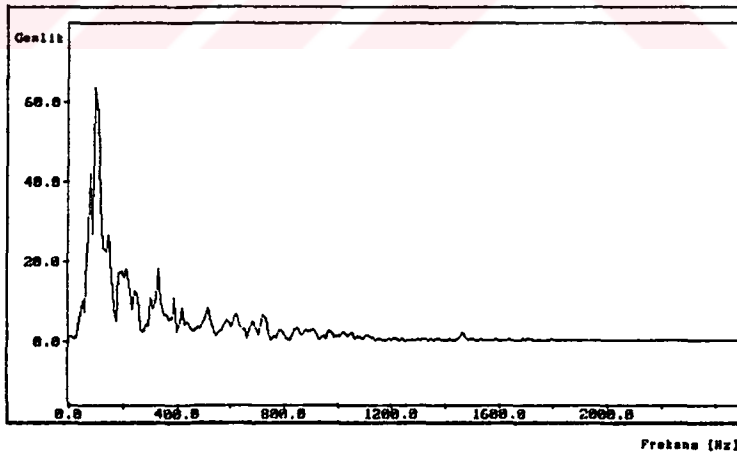
(d)

Şekil 4-5. (a) Zatürreeli denegin sol taban kaydı (b) zatürreeli denegin FFT spektrumu (c) sağlıklı denegin sol taban kaydı (d) sağlıklı denegin FFT spektrumu.

Şekil 4-6a ve 4-6b'de bronkopnömonili bir hastanın kaydından elde edilen çıtırtı sesi dalga biçimi ile FFT spektrumu görülmektedir. Gerek zaman dalga biçiminden, gerekse frekans spektrumundan, çıtırtı sesinin, üssel olarak sönümlenen sinüsoidallerin toplamından oluştuğu anlaşılmaktadır.



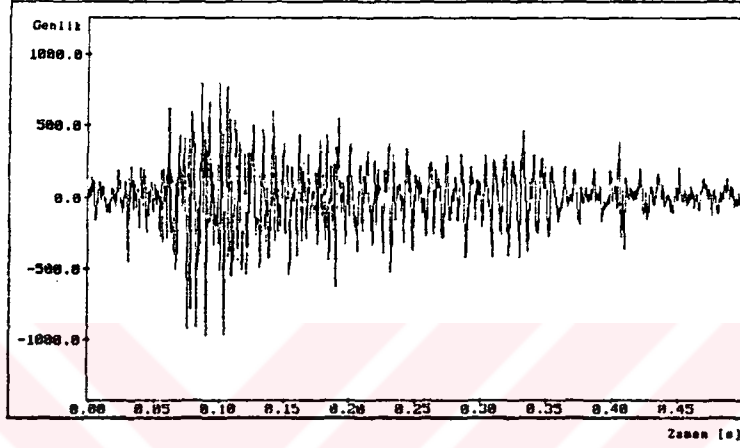
(a)



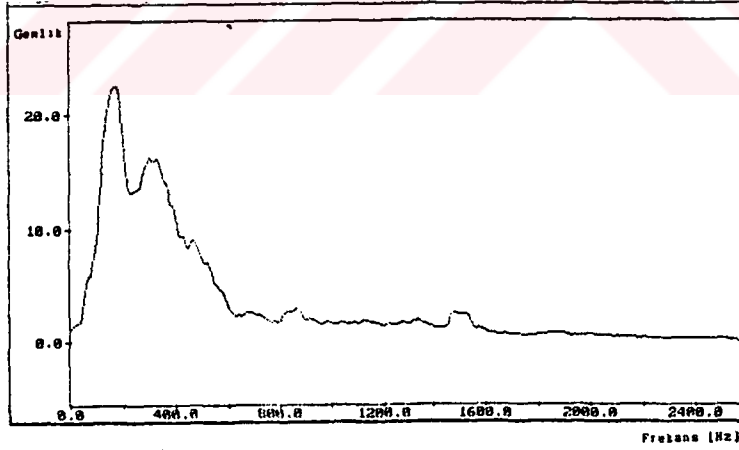
(b)

Şekil 4-6 Çıtırtı (a) dalga şekli ve (b) FFT spektrumu.

Şekil 4-7'de astımlı bir hastanın bir yarım soluk çevrimi ile buna karşı gelen yumuşatılmış FFT spektrumu görülmektedir. Spektrumda 300 Hz civarında görünen tepe, bu frekanslarda bir ısıklık sesi oluştuğunu göstermektedir.



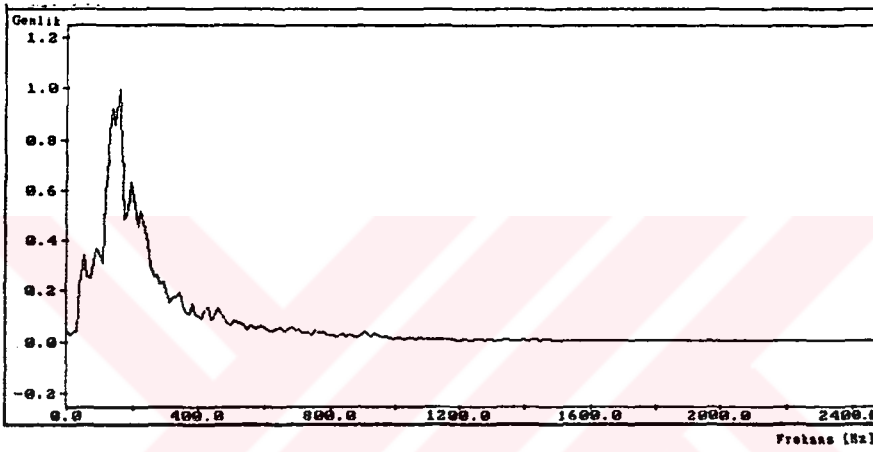
(a)



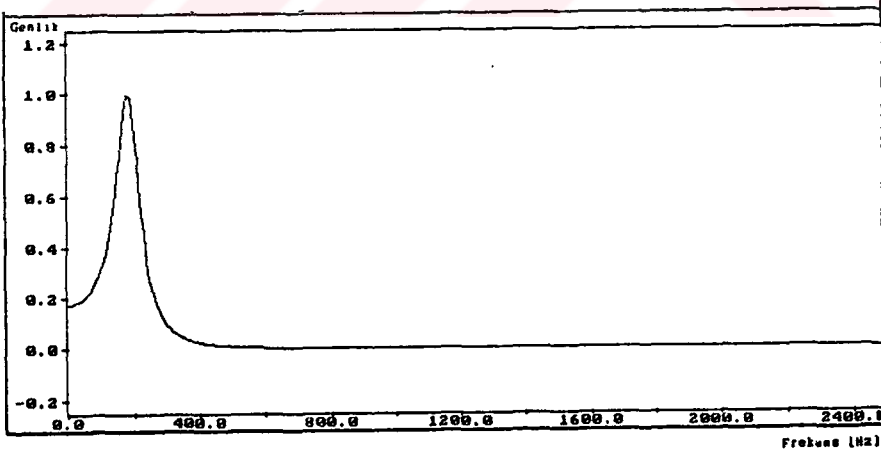
(b)

Şekil 4-7 Astımlı bir hastanın (a) bir yarım çevrim dalga biçimi ve (b) FFT spektrumu.

Şekil 4-8'de sağlıklı bir denneğin ortalama frekans spektrumu ile 8. dereceden bir AR modeli yardımıyla kestirilen spektrum verilmiştir. Her iki spektrum da en büyük değerlerine düzgelenmişlerdir. Bulunan AR spektrum kestirimi aynı zamanda göğüs kafesinin aktarganlık işlevinin de bir kestirimi olduğundan iletim yolunu etkileyen akciğer hastalıklarının yol açtığı değişiklikleri gösterebilir.



(a)

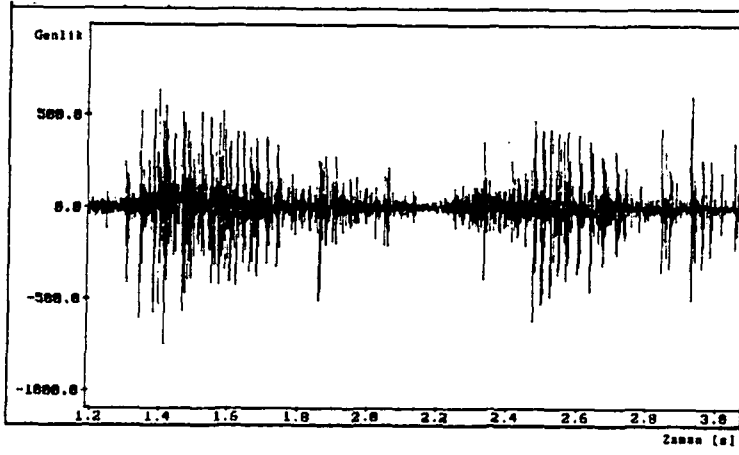


(b)

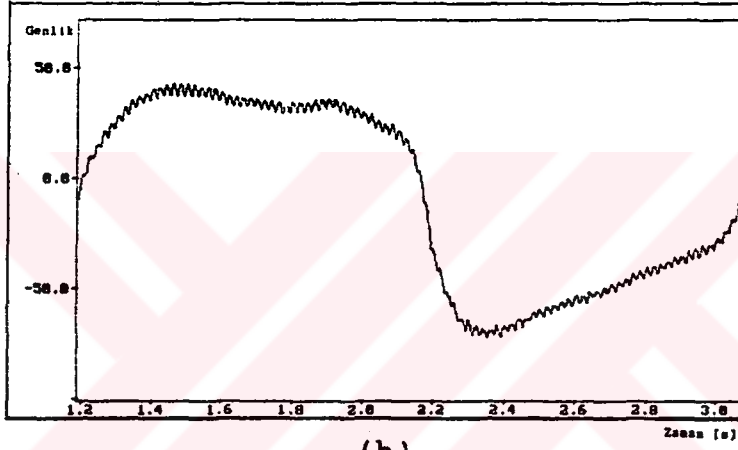
Şekil 4-8. Sağlıklı bir denneğin (a) FFT spektrumu (b) AR analizinden elde edilen kestirilmiş spektrum.

Solunum seslerinin iç ve dışsolumada farklılıklar göstermesi ve evre değişim noktalarında duragansızlıklar meydana gelmesi, hem analizi hem de sınıflandırmayı soluk çevrimine bağlı kılmaktadır. Şekil 4-9'da zatürreeli bir denekten seçilen soluk çevriminden hesaplanan güç spektrumu kestiriminin zaman içindeki değişimi görülmektedir. Spektrum değişimi, Şekil 4-10'da verilen sağlıklı denegin spektrum değişimi ile karşılaştırıldığında, yüksek frekanslarda bileşenler içerdiği görülmektedir.

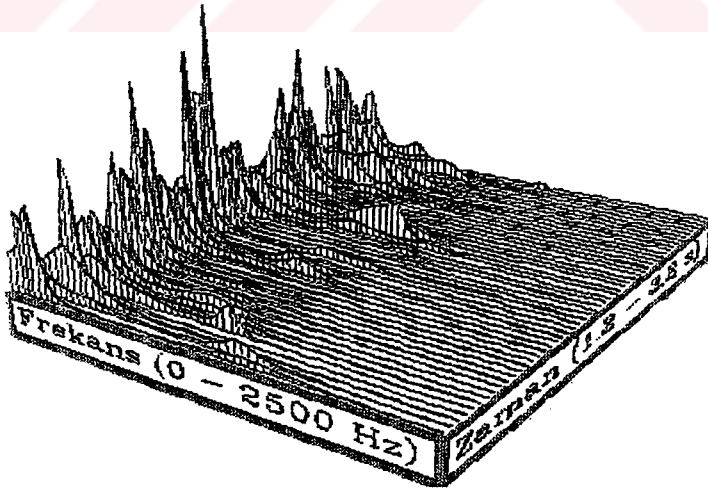
Solunum seslerinin k-EYK yöntemi ve Itakura uzaklık ölçüsünden yararlanılarak sınıflandırılmasından elde edilen sonuçlar Tablo 4-1'de verilmiştir. Eğitime ve sınıflandırma, 12 tane sağlıklı, 12 tane de zatürreeli soluk çevrimi üzerinde gerçekleştirilmiştir. Soluk çevrimleri sol taban ve sağ tabana ait olup, soluk borusundan alınan sesler, yüksek frekans bileşenleri içermeleri ve zatürreeye ilişkin herhangi bir belirti taşımamaları nedeniyle kullanılmamıştır. Sınıflandırıcı gelen işareti mutlaka iki sınıftan birine atamış, yani "bilinmeyen" olarak nitelendirmemiştir. Sınıflandırma içsoluma ve dışsolumaya göre ayrı ayrı yapılmış, yani her soluk evresi, dayanak kütüphanedeki kendine karşı gelen soluk evresindeki örneklerden yararlanarak Bölüm 4.2'de anlatılan "oylama yöntemi" ile sınıflandırılmıştır. Tüm soluk çevriminin sınıflandırılmasında ise, bu soluk çevrimini oluşturan soluk evrelerinin sınıflandırılmasından elde edilen sonuçlar kullanılmıştır. Yani içsolumada % m çerçeve, dışsolumada da % n çerçeve sağlıklı olarak sınıflandırılmışsa, bu soluk çevrimi  $\% (m+n)/2$  olasılıkla sağlıklı olarak kabul edilmiştir. k'nın 1, 3 ve 5 değerleri için tasarlanan sınıflandırıcının başarımının ölçülmesinde birini-dışarıda-bırak (leave-one-out) [26] yöntemi kullanılmıştır.



(a)

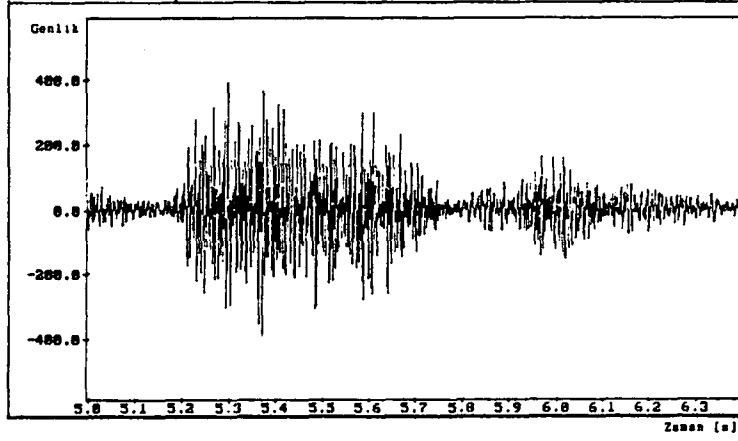


(b)

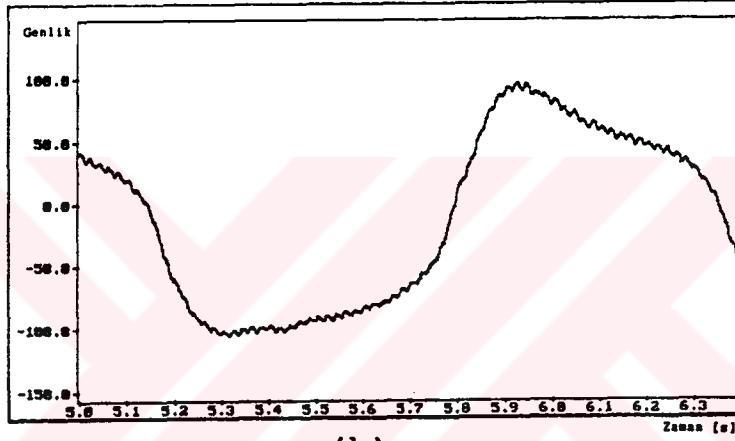


(c)

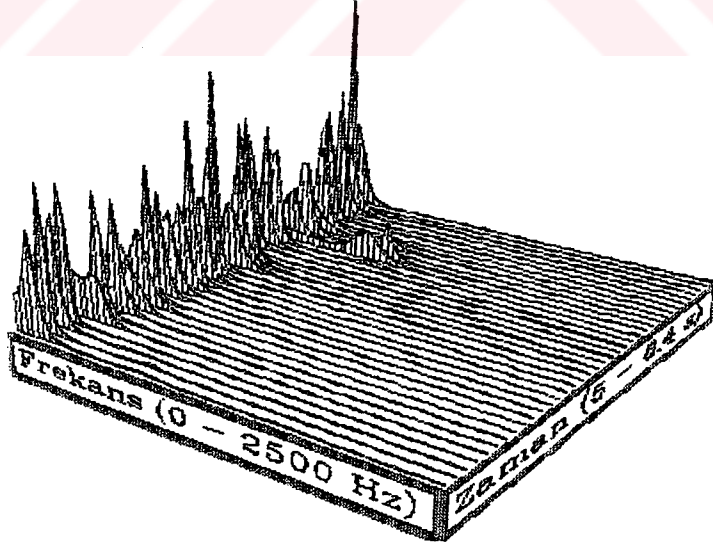
Şekil 4-9. Zatürreeli bir denegin (a) soluk çevrimi (b) karşı düşen akış işareti (c) kestirilen spektrumun soluk çevrimi boyunca değişimi.



(a)



(b)



(c)

Şekil 4-10. Sağlıklı bir denneğin (a) soluk çevrimi (b) karşı düşen akış işareti (c) kestirilen spektrumun soluk çevrimi boyunca değişimi.

Tablo 4-2. Sınıflandırıcının başarımlı testi sonuçları

|       | k = 1  |       |     | k = 3  |       |     | k = 5  |       |     |
|-------|--------|-------|-----|--------|-------|-----|--------|-------|-----|
|       | Dış S. | İç S. | Tüm | Dış S. | İç S. | Tüm | Dış S. | İç S. | Tüm |
| % DS  | 77     | 62    | 85  | 77     | 69    | 77  | 77     | 62    | 77  |
| % YP  | 29     | 29    | 14  | 29     | 14    | 29  | 29     | 29    | 29  |
| % YN  | 17     | 50    | 17  | 17     | 50    | 17  | 17     | 50    | 17  |
| % DU  | 83     | 50    | 83  | 83     | 50    | 83  | 83     | 50    | 83  |
| % AY  | 71     | 71    | 86  | 71     | 86    | 71  | 71     | 71    | 71  |
| % PKD | 71     | 60    | 83  | 71     | 75    | 71  | 71     | 60    | 71  |
| % NKD | 83     | 63    | 86  | 83     | 67    | 83  | 83     | 63    | 83  |

DS : Doğru sınıflandırma

YP : Yanlış pozitifler

YN : Yanlış negatifler

DU : Duyarlılık

PKD: Pozitif kestirilen değer

NKD: Negatif kestirilen değer

Dış S.: Dışsoluma

İç S. : İçsoluma

Tüm : Tüm soluk çevrimi

Birini-dışarıda-bırak yönteminde sınıflandırıcı, örneklerin biri hariç hepsiyle eğitilmekte, daha sonra dışarıda bırakılan örnek, eğitilen sınıflandırıcı ile sınıflandırılmaktadır. Bu işlem her bir örnek dışarıda bırakılarak tekrarlanmaktadır. Bu yöntem örnek sayısının az olduğu durumlarda kullanılmakta olup sınıflandırıcının hata olasılığını gerçeğe yakın olarak vermektedir. Başarımlı ölçülmesi doğru sınıflandırmalar (% DS), yanlış pozitifler (% YP), yanlış negatifler (% YN) yüzdelerinin hesaplanması ile gerçekleştirilmiştir. Bu büyüklükler

$$\% DS = \frac{(DP + DN)}{N} \times 100 \quad (4.5)$$

$$\% YP = \frac{YP}{(DN + YP)} \times 100 \quad (4.6)$$

$$\% YN = \frac{YN}{(DP + YN)} \times 100 \quad (4.7)$$

denklemleri ile hesaplanabilir. Denklemlerde DP, doğru pozitiflerin, yani hasta iken hasta olarak sınıflandırılanların sayısını, DN, doğru negatiflerin, yani sağlıklı iken sağlıklı olarak sınıflandırılanların sayısını, YP yanlış pozitiflerin, yani sağlıklı iken hasta olarak sınıflandırılanların sayısını, YN ise yanlış negatiflerin, yani hasta iken sağlıklı olarak sınıflandırılanların sayısını vermektedir. Sınıflandırıcının başarımlarında duyarlılık (DU), ayırganlık (AY), pozitif kestirilen değer (PKD) ve negatif kestirilen değer (NKD) büyüklükleri de kullanılmıştır. Bu büyüklükler

$$\% DU = \frac{DP}{(DP + YN)} \times 100 \quad (4.8)$$

$$\% AY = \frac{DN}{(DN + YP)} \times 100 \quad (4.9)$$

$$\% \text{ PKD} = \frac{\text{DP}}{(\text{DP} + \text{YP})} \times 100 \quad (4.10)$$

$$\% \text{ NKD} = \frac{\text{DN}}{(\text{DN} + \text{YN})} \times 100 \quad (4.11)$$

denklemleri ile hesaplanabilir.

Tablo 4-2'den de anlaşılabacağı gibi sınıflandırıcı en iyi başarımı  $k=1$  için göstermiştir. 1-EYK'da tüm soluk çevrimi için doğru sınıflandırma oranı yüzde 85 iken, 3-EYK ve 5-EYK için doğru sınıflandırma oranı yüzde 77'de kalmıştır. Sağlıklı iken hasta olarak sınıflandırılanların oranı  $k=1$  için yüzde 14 iken,  $k=3$  ve  $k=5$  için yüzde 29'a yükselmiştir. Hasta iken sağlıklı olarak sınıflandırılanların oranı bütün  $k$ 'lar için yüzde 17 olarak bulunmuştur. Tüm sınıflandırıcıların duyarlılıklarının aynı olmasına karşın 1-EYK sınıflandırıcısı yüzde 83'lük bir ayırganlık ile diğerlerinden daha çok ayırt edici gözükmektedir. Ayrıca 1-EYK için işlem sayısının daha az, dolayısıyla karar verme süresinin daha kısa olması da olumlu bir gelişmedir.  $k=1$ ,  $k=5$  ve kısmen  $k=3$  için içsolumaları sınıflandırmadaki başarımın, dışsolumaları sınıflandırmadaki başarımdan daha düşük olduğu gözlenmektedir. Bunun nedeni solumun sesinin soluk çevrimine göre değişen doğası, hastalıkların farklı soluk çevrimlerinde farklı belirtiler göstermesi olabilir. Ayrıca kayıt sırasında ağızdan gelen yüksek frekanslı soluk seslerinin bazı deneklerde sol ve sağ taban kayıtlarına karışması, sınıflandırıcının iç ve dışsolumalardaki başarımını etkileyebilir.

## SONUÇLAR VE ÖNERİLER

Bu çalışmada solunum seslerinin bilgisayar yardımıyla analizi ve sınıflandırılmasına ilişkin bir yöntem önerilmiştir. Yapılan deneylerde sınıflandırıcının karakteristik özellikleri belirgin olan zatürreeli soluk çevrimlerini yüzde 85'lik doğruluk oranı ile ayırtadedebildiği görülmüştür.

Sınıflandırıcının başarımını etkileyen faktörlerin başında solunum seslerinin akış hızı, kayıt yerleri, solunum şekli, denegin pozisyonu gibi bir çok parametreye bağlı olması gelmektedir. Kayıt şartları, örneğin kayıt ortamının gürültülü olması da, dolaylı olarak hem analizi, hem de sınıflandırmayı etkileyen bir etmendir. Ayrıca sağlıklı bir örüntü tanıma ve sınıflandırma için dayanak veri tabanının oldukça büyük olması gerekmektedir. Daha uygun şartlarda, daha sağlıklı kayıtlar alınması, büyük bir veri tabanı kullanılması ve deneklerin yeteri kadar eğitilebilmesi halinde sınıflandırıcının başarımının daha da artması beklenmektedir.

Sınıflandırıcıda k-en yakın komşu yöntemi ve Itakura uzaklık ölçüsü kullanılmıştır. Yapılabilecek başka bir çalışmada, Mahalonobis uzaklık ölçüsü veya sınıflandırıcının gürültüye olan bağışıklığını arttırmak için frekans ağırlıklandırmalı Itakura ölçüsü gibi diğer ölçüler ve sınıflandırma yöntemleri denenerek bir karşılaştırma yapılabilir. Ya da, soluk seslerinin Gaussian olmadığı düşünülerek yüksek dereceli spektrum

yoluyla AR/ARMA parametreleri kestirilerek daha iyi bir sistem belirleme yoluna gidilebilir. Bu yolla astım hastalıklarında görülen çoksesli ıslıklar arasındaki evre bağlaşımı sezilebilir.

Yapılan deneyler çitirtı seslerinin frekans spektrumunda çok belirgin etkileri olmadığını göstermiştir. Test edilen sınıflandırıcı, büyük bir olasılıkla, zatürreeli hastalarda, göğüs kafesinin aktarganlık işlevinin değişmesini esas alarak sınıflandırma yapmaktadır. Çitirtılar üzerinde yapılan incelemeler, çitirtıların Denk. (2.1)'deki gibi üssel olarak sönümlenen sinosoidlerin toplamı olarak modellenebileceğini göstermiştir. Çitirtıların doğrusal olmayan bir süzgeçle uyarlamalı bir yöntemle sezilip soluk sesinden arındırılmasından sonra, bu model kullanılarak çitirtı analizi ve sınıflandırılması yapılabilir. Ayrıca bu parametrelerle eğitilen bir sınıflandırıcı kullanılarak çitirtıların sezilmesi gerçekleştirilebilir.

Uzman dizgeler son yıllarda biyomedikal alanında önemli uygulama alanları bulmuşlardır. Solunum seslerinden elde edilen parametreler uzmanların öznel yargılarına dönüştürülebilirse, işaret işlemenin ve uzman dizgelerin etkileşimli kullanıldığı akciğer ve solunum hastalıkları tanı sistemleri gerçekleştirilebilir.

## KAYNAKLAR

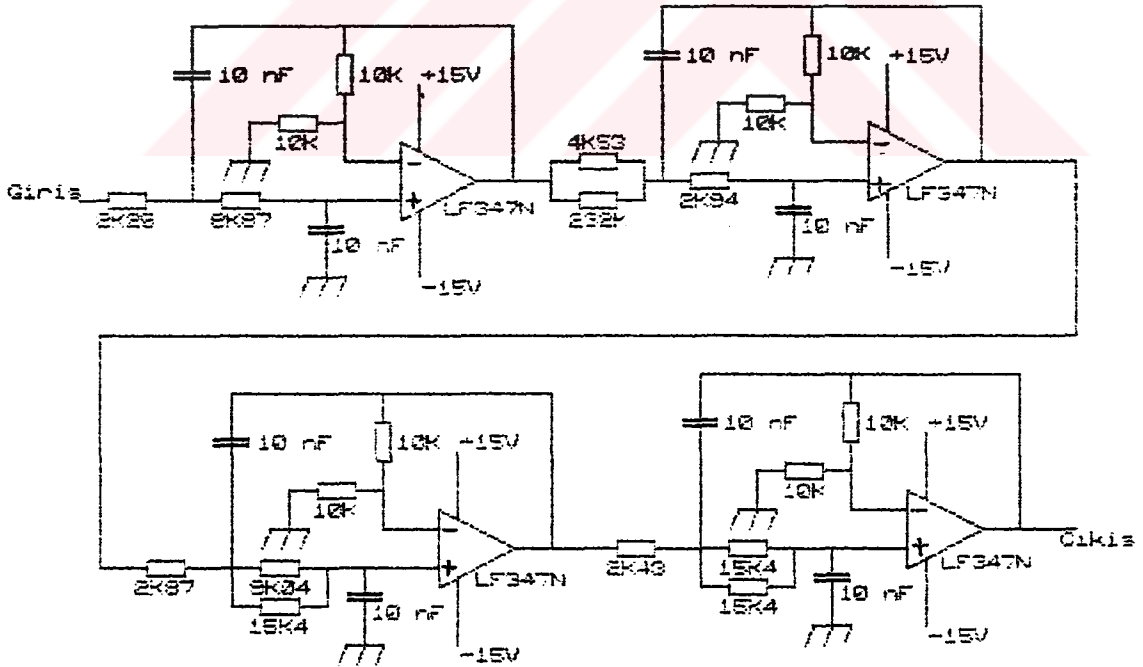
- [1] COHEN, A., Signal Processing Methods for Upper Airway and Pulmonary Dysfunction Diagnosis, IEEE Eng. Med. Biol. Magazine, 72-75, (March 1990).
- [2] SMYLLIE, H.C., BLENDIS, L.M., ARMITAGE, P., Observer Disagreement in Physical Signs of the Respiratory System, The Lancet, 412-413, (Aug. 1965).
- [3] FORGACS, P., Lung Sounds, London, England : Baillere Tindall, (1978).
- [4] COHEN, A., LANDSBERG, D., Analysis and Automatic Classification of Breath Sounds, IEEE Trans. on Biomed. Eng., BME-31, 585-590, (1986).
- [5] KAHYA, Y., A Computerized Pulmonary Diagnosis System, Ph.D. Thesis, Boğaziçi University, (1987).
- [6] DRUZGALSKI, C.K., DONNENBERG, R.L., CAMPBELL, R.M., Techniques of Recording Respiratory Sounds, Journal of Clinical Engineering, 321-329, (Dec. 1980).
- [7] BASSEVILLE, M., Distance Measures For Signal Processing and Pattern Recognition, Signal Processing, Vol.18, No.4, 349-369, (Dec. 1989).
- [8] SANKUR, B., KAHYA, Y., ENGİN, T., GÜLER, Ç., Analysis and Classification Of Respiratory Sounds, Proc. of IEEE/ProRISC Symposium on Circuits, Systems and Signal Processing, Veldhoven, 1-9, (1991).
- [9] KRAMAN, S.S., Lung Sounds : Relative Sites of Origin and Comparative Amplitudes in Normal Subjects, Lung, Vol. 161, 57-64, (1983).

- [10] IYER, V.K., RAMAMOORTHY, P.A., PLOYSONGSANG, Y.,  
Autoregressive Modelling of Lung Sounds:  
Characterization of Source and Trans-  
mission, IEEE Trans. on Biomed. Eng.,  
Vol. 36. No. 11, 1133-1137, (Nov. 1989).
- [11] FENTON, T.R., PASTERKAMP, H., TAL, A., CHERNICK,  
V., Automated Spectral Characterization  
of Wheezing in Asthmatic Children, IEEE  
Trans. on Biomed. Eng., Vol. BME-32, No. 1,  
50-55, (Jan. 1985).
- [12] PASTERKAMP, H., FENTON, R., LEAHY, F., CHERNICK,  
H., Spectral Analysis Of Breath Sounds  
In Normal Newborn Infants, Medical  
Instrumentation, Vol. 17, No. 5, 355-357,  
(Sept.- Oct. 1983).
- [13] SHYKOFF, B.E., PLOYSONGSANG, Y., CHANG, H.K.,  
Airflow and Normal Lung Sounds, Am. Rev.  
Respir. Dis. Vol. 137, 872-876 (1988).
- [14] COHEN, A., LEIBERMAN, A., Analysis and Classifica-  
tion of Snoring Signals, Proc. of The  
Int. Conf. on Acoust. Speech and Sig.  
Proc. (ICASSP), Tokyo, 693-696, (1986).
- [15] CHARBONNEAU, G., RACINEUX, J.L., SUBRAUD, M.,  
TUCHAIS, E., An Accurate Recording System  
and Its Use in Breath Sounds Spectral  
Analysis, J. Appl. Physiol., Vol. 55, No.  
4, 1120-1127, (1983).
- [16] IYER, V.K., RAMAMOORTHY, P.A., FAN, H. PLOYSONGSANG  
Y., Reduction of Heart Sounds from Lung  
Sounds by Adaptive Filtering, IEEE Trans.  
on Biomed. Eng., Vol. BME-33, No. 12,  
1141-1148, (Dec. 1986).
- [17] ONO, M., ARAKAWA, K., MORI, M., SUGIMOTO, T.,  
HARASHIMA, H., Seperation Of Fine  
Crackles From Vesicular Sounds By A  
Nonlinear Digital Filter, IEEE Trans. on  
Biomed. Eng., Vol. BME-36, No. 2,  
286-291, (Feb. 1989).
- [18] KÖYMEN, H., ALTAY, B.K., IDER, Y.Z., A Study Of  
Prosthetic Heart Valve Sounds, IEEE  
Trans. on Biomed. Eng., Vol. BME-34, No.  
11, 853-863, (Nov. 1987).

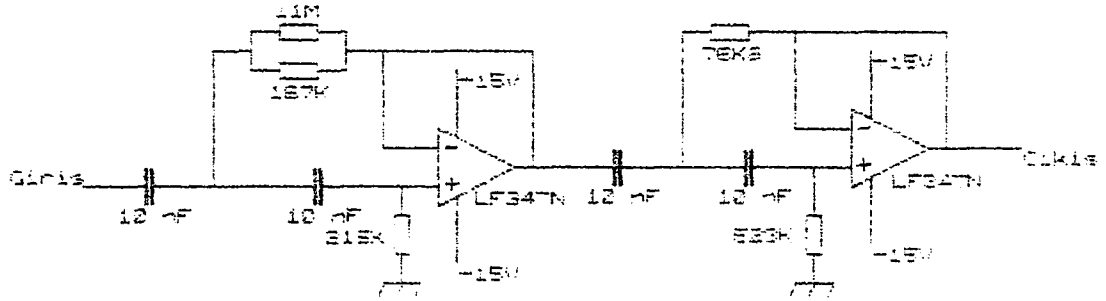
- [19] SCHREUR, H.J.W., STERK, P.J., The Comparison Of Airflow-Standardized Lung Sound Intensity Between Normal Subjects And Patients With Emphysema, University Hospital Leiden Report.
- [20] COHEN, A., Biomedical Signal Processing, CRC Press Inc., Florida, (1986).
- [21] HAYKIN, S., Adaptive Filter Theory, Prentice-Hall, Englewood Cliffs, New Jersey, (1986).
- [22] RABINER, L.R., SHAFFER, R.W., Digital Processing of Speech Signals, Prentice-Hall, Englewood Cliffs, New Jersey, (1979).
- [23] DUDA, R.O., HART, P.E., Pattern Classification And Scene Analysis, John Wiley & Sons, Inc., (1973).
- [24] HART, P.E., The Condensed Nearest Neighbor Rule, IEEE Trans. Inform. Theory, Vol. IT-14, 515-516, (May 1968).
- [25] SOONG, F.K., SONDHI, M.M., A Frequency Weighted Itakura Distorsion Measure and Its Applications to Speech Recognition in Noise, IEEE Trans. on ASSP, Vol. 36, No. 1, 41-48, (Jan. 1988).
- [26] DURAND, L., BLANCHARD, M., CLOUTIER, G., SABBAH H.N., STEIN, P., Comparison of Pattern Recognition Methods for Computer-Assisted Classification of Spectra of Heart Sounds in Patients with a Porcine Bioprosthetic Valve Implanted in the Mitral Position, IEEE Trans. on Biomed. Eng., Vol. BME-37, No. 12, 1121-1129, (Dec. 1990).

## EK A. DONANIM

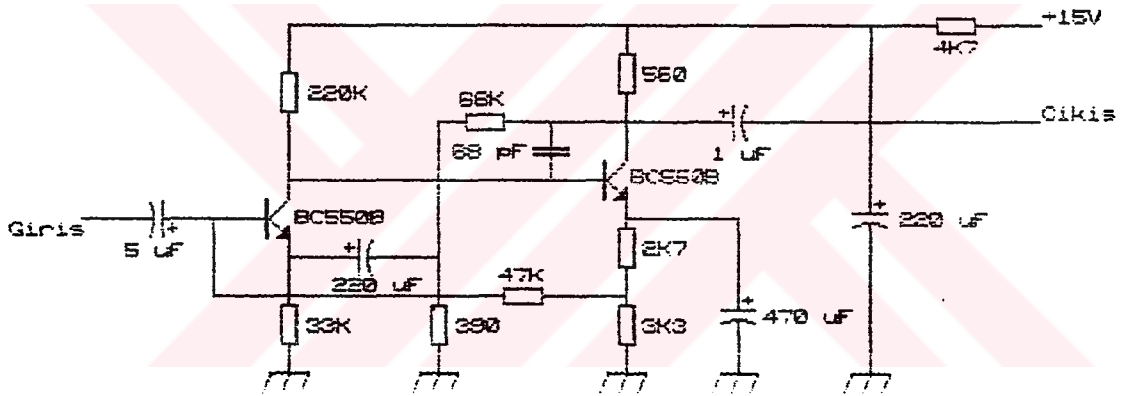
Solunum seslerinin kayıt aşamasında kullanılan alçakgeçiren süzgeç, yüksekgeçiren süzgeç ve önkuvvetlendiricinin devre şemaları sırasıyla Şekil A-1, A-2 ve A-3'te verilmiştir. Alçakgeçiren süzgeç 8. dereceden bir Bessel süzgeci olup 2000 Hz kesim frekansına ve 24 dB'lik bir kazanca sahiptir. Yüksekgeçiren süzgeç ise 4. dereceden bir Butterworth süzgeci olup 80 Hz'lik kesim frekansına sahiptir. Her iki süzgeçte geçirme bantlarında doğrusal evreye sahiptirler. Önkuvvetlendirici ise 36 dB'lik bir kazanca sahip olup frekans yanıtı 5-100000 Hz arasında düzdür.



Şekil A-1. 8 kutuplu Bessel alçakgeçiren süzgeç.

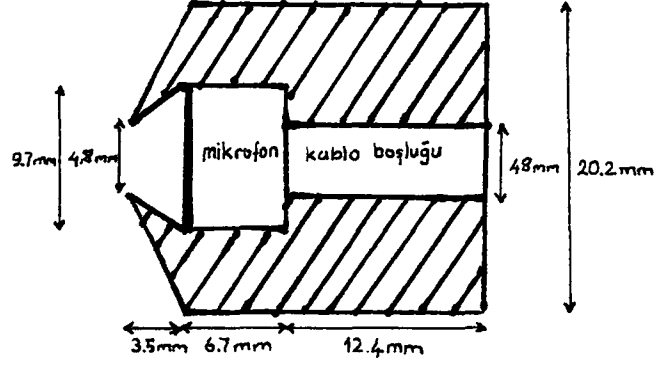


Şekil A-2. 4 kutuplu Butterworth yüksekgeçiren süzgeç.



Şekil A-3. 36 dB kazançlı önkuvvetlendirici.

Kayıt yaparken kullanılan mikrofonların içine konduğu plastik kılıfın iki kat büyütülmüş şeması Şekil A-4' de verilmiştir.



Şekil A-4. Mikrofon kılıfının iki kat büyültülmüş şeması.

## EK B. BİLGİSAYAR PROGRAMLARI

Bu bölümde, önceki bölümlerde anlatılan analiz ve sınıflandırma yöntemlerinden solunum seslerinin AR modellerle modellenmesine ve k-EYK kuralı ve Itakura ölçüsünden yararlanılarak sınıflandırılmasına ilişkin bilgisayar programları verilmiştir. Programlar Light-Speed-C derleyicisinde ve MacIntosh bilgisayarı üzerinde yazılmışlardır.



```

/*****
    Bu program zatureeli ve saglikli soluk cevrimlerini
    K-EYK kuralindan ve Itakura olcusunden yararlanarak
    siniflandirir.
*****/

#include <stdio.h>
#include <storage.h>
#include <math.h>
#include <unix.h>
#include "cadold.h"
#include "DataManipulation.h"

/* global degiskenler */
int frlen      = 256;
int K          = 1;
int numofCase  = 2; /*saglikli,zaturee*/
int overlap    = 64;
int order      = 8;
int libcheck;
coef_set *start;
coef_set *last;

/*-----*/
void Initialize()
{
    int Case, disker;
    char *fname;
    int *samparr, *flowarr;
    long len;

    fname = (char*)NewPtr(200*sizeof(char));
    RemoveLib();
    for (Case=1; Case<=numofCase; Case++) {
        printf("%d. siniftan bir dosya adi girin ->", Case);
        gets(fname);
        filelen("YPK:", CtoPstr(fname), len, &disker);
        samparr = (int*)NewPtr(len);
        bload("YPK:", CtoPstr(fname), samparr, len, &disker);
        printf("ayni denegin akis kayit dosyasinin adini girin ->");
        gets(fname);
    }
}

```

```

        filelen("YFK:", CtoPstr(fname), len, &diskr);
        flowarr = (int*)NewPtr(len);
        bload("YFK:", CtoPstr(fname), flowarr, len, &diskr);
        if (diskr != 0) {
            printf("Dosya okumada hata !!!");
            ExitToShell();
        }
        InitRefLib(samparr, flowarr, len/sizeof(int), Case);
        DisposPtr(flowarr);
        DisposPtr(samparr);
        DisposPtr(fname);
    }
    MainMenu();
}

/*-----*/
void Condense()
{
    int Case, diskcr, numofadds;
    unsigned Hf, Hs;
    char *fname;
    int *samparr, *flowarr;
    long len;

    fname = (char*)NewPtr(200*sizeof(char));
    do {
        numofadds = 0;
        for (Case=1; Case<=numofCase; Case++) {
            printf("%d. siniftan bir dosya adi girin ->", Case);
            gets(fname);
            filelen("YFK:", CtoPstr(fname), len, &diskcr);
            samparr = (int*)NewPtr(len);
            bload("YFK:", CtoPstr(fname), samparr, len, &diskcr);
            printf("ayni denegin akis kayit dosyasinin adini girin ->");
            gets(fname);
            filelen("YFK:", CtoPstr(fname), len, &diskcr);
            flowarr = (int*)NewPtr(len);
            bload("YFK:", CtoPstr(fname), flowarr, len, &diskcr);
            if (diskcr != 0) {
                printf("Dosya okumada hata !!!");
                ExitToShell();
            }
        }
        Hf = (unsigned)(len/sizeof(int));
    }

```

```

        numofadds += CondenseRefLib(samparr, flowarr, Hf, Case);
        DisposPtr(flowarr);
        DisposPtr(samparr);
    }

    } while (numofadds != 0); /*dayanak kutuphaneye eklenen varsa
        tekrar basa don */

    DisposPtr(fname);
    SaveLib();
    libcheck = TRUE;
    MainMenu();
}

/*-----*/
void ClassifySubject()
{
    int i, j, decision, phase;
    int healthcnt, pneucnt, diskcr;
    int *samparr, *flowarr, *ifr;
    unsigned firstpt, lastpt;
    unsigned Hf, frbegin;
    unsigned phaserec[200];
    long len;
    char *fname;
    float *cor, *ffr, *par, *LPC, error;

    cor = (float*)NewPtr((order+1)*sizeof(float));
    ifr = (int*)NewPtr(frlen*sizeof(int));
    ffr = (float*)NewPtr(frlen*sizeof(float));
    par = (float*)NewPtr((order+1)*sizeof(float));
    LPC = (float*)NewPtr((order+2)*sizeof(float));
    fname = (char*)NewPtr(200*sizeof(char));

    if (!cor&&!par&&!LPC&&!ifr&&!ffr&&!fname) {
        printf("Bellek kalmadi!!!");
        ExitToShell();
    }

    if (libcheck != TRUE) {
        eraseplot();
        printf("Dikkat! Dayanak kutuphane yok!!!");
        getch();
        MainMenu();
    }
}

```

```

eraseplot();
printf("%d. dosya adi girin ->");
gets(fname);
filelen("YPK:", CtoPstr(fname), len, &disker);
samparr = (int*)NewPtr(len);
bload("YPK:", CtoPstr(fname), samparr, len, &disker);
printf("ayni denegin akis kayit dosyasinin adini girin ->");
gets(fname);
filelen("YPK:", CtoPstr(fname), len, &disker);
flowarr = (int*)NewPtr(len);
bload("YPK:", CtoPstr(fname), flowarr, len, &disker);
if (disker != 0) {
printf("Dosya okumada hata !!!");
ExitToShell();
}
Nf = (unsigned)(len/sizeof(int));
InspExp(flowarr, Nf, phaserec, 200); /*ic-dissoluma ayrimi*/
eraseplot();
printf("Denek siniflandiriliyor, lutfen bekleyin. ....");
for (phase = EXP; phase <= INSP; phase++) {
    if (phase == EXP) printf("Dissoluma Evresi");
    else printf("Icsoluma Evresi");
healthcnt=pneucnt=0;
j=0;
do {
    /*dissolumanin baslangic ve bitis noktaları bulunuyor*/
    if (phase==EXP) {
        while(phaserec[j] && phaserec[j]!=EXP_BEGIN) j+=2;
        firstpt=phaserec[j+1];
        while(phaserec[j] && phaserec[j]!=EXP_END) j+=2;
        lastpt=phaserec[j+1];
    }
    /*icsolumanin baslangic ve bitis noktaları bulunuyor*/
    else if (phase==INSP) {
        while(phaserec[j] && phaserec[j]!=INSP_BEGIN) j+=2;
        firstpt=phaserec[j+1];
        while(phaserec[j] && phaserec[j]!=INSP_END) j+=2;
        lastpt=phaserec[j+1];
    }
    for (i=0; i++) {
frbegin = (firstpt+overlap)+i*(frlen-overlap);
        if (frbegin+frlen < lastpt) {
            GetFrame(ifr, samparr, frbegin, frlen);
            Scale(ifr, ffr, frlen);

```

```

    for (i=0;;i++) {
        frbegin = (firstpt+overlap)+i*(frlen-overlap);
        if (frbegin+frlen < lastpt) {
            GetFrame(ifr, samparr, frbegin, frlen);
            Scale(ifr, ffr, frlen);
            Hamming(ffr, frlen);
            decision = ClassifyFrame(ffr, LPC, numofCase, phase);
            if (decision==HEALTHY)
                ++healthcnt;
            else if (decision==PNEUMONIA)
                ++pneucnt;
        } else
            break;
    }

    while(phaserec[j++2] != 0); /*bir sonraki ic veya dissoluma
                                evresine konumlan*/
    printf("SAGLIKLI   : yuzde %.2f", healthcnt*100./(healthcnt+pneucnt));
    printf("ZATURREK   : yuzde %.2f", pneucnt*100./(healthcnt+pneucnt));
}
DisposPtr(cor); DisposPtr(par);
DisposPtr(ffr); DisposPtr(ifr);
DisposPtr(LPC); DisposPtr(fname);
printf("siniflandirma sona erdi.");
getch();
MainMenu();
}

/*-----*/
void MainMenu()
{
    eraseplot();
    printf("1- Dayanak Kutuphaneyi Ilkle");
    printf("2- Dayanak Kutuphaneyi Sikistir");
    printf("3- Siniflandir");
    printf("4- Bitti");
    switch(getch()) {
        case '1': Initialize();break;
        case '2': Condense();break;
        case '3': ClassifySubject();break;
        case '4': ExitToShell();
        default : MainMenu();
    }
}
}

```

```
/*-----*/  
main()  
{  
    start=last=NULL; /*listenin uzunlugunu sifir yap*/  
    libcheck = LoadLib();  
    MainMemu();  
}
```



```
/*****< ISARET ISLEME ISLEVLERI >*****/
```

Asagidaki islevler solunum seslerinin analizi sırasında kullanılan islevlerdir.

```
*****/
```

```
#include <storage.h>
#include <math.h>
#include "cadold.h"
```

```
/*****-----*/
```

AutoCor : Bu islev 'N' orneklik veri dizisi 's[]'nin 'ord' sayidaki duzgelemis ozilinti katsayilarini hesaplar ve 'cor[]' dizisine atar

```
*****/
```

```
void AutoCor(s, N, ord, cor)
float *s, *cor;
int N, ord;
{
    register int i, j;
    double eng = 0.0;
    for (i=0; i<N; i++) eng += s[i]*s[i];
    for (i=1; i<ord; i++) {
        cor[i] = 0.0;
        for (j=0; j<N-i-1; j++)
            cor[i] += s[j]*s[i+j];
        cor[i] /= eng;
    }
    cor[0] = 1.0;
}
```

```
/*****-----*/
```

Lpc : Bu islev 'cor[]' ozilinti dizisi verilmiş bir veri için 'ord' dereceden lpc katsayilarini hesaplar ve 'lpc[]' dizisinde geri gönderir. Islevde ayrıca kisimsal ilinti katsayilari 'par[]', karesel ongoru hatasi da 'err' degiskenleri ile cagiran programa gönderilir.

```
*****/
```

```
void Lpc(ord, cor, lpc, par, err)
int ord;
float *cor, *lpc, *par, *err;
```

```

{
    int i, j;
    float *war, sum;

    war = (float*)NewPtr(sizeof(float)*(ord+1));

    *err = cor[0];

    for (i=1; i<=ord; i++) {
        sum = 0.0;
        for (j=1; j<=i-1; j++)
            sum += lpc[j]*cor[i-j];
        par[i] = -(cor[i]+sum)/(*err);
        if (fabs(par[i]) > 1) {
            printf("Ongorucu Kararsiz!!!");
            getch();
            return;
        }
        lpc[i] = par[i];
        for (j=1; j<=i-1; j++)
            war[j] = lpc[j] + par[i]*lpc[i-j];
        for (j=1; j<=i-1; j++)
            lpc[j] = war[j];
        *err *= (1.0-par[i]*par[i]);
    }
}

```

```

/*-----*/

```

InspExp : Bu islev akis isaretinden yararlanarak akis evrelerini ayirir. 'flowarr[]' akis isareti dizisi, 'Hf' akis isaretinin uzunlugu, 'phaserec[]' evrelerin baslangic ve bitis noktalarini gosteren evre kayit dizisi, 'Np' de bu dizinin uzunlugudur.

```

/*-----*/

```

```

void InspExp(flowarr, Hf, phaserec, Np)
int *flowarr;
unsigned *phaserec, Hf, Np;

```

```

{

    int phase, max1, max2, maxavg;
    int threshold, guardband=1000, locmax;
    double DC=0.0;

```

```

unsigned n, j, k;
unsigned *temprec;

temprec=(unsigned*)NewPtr(sizeof(unsigned)*Np);

/* DC degeri hesapla */
for (n=0;n<Nf;n++)
    DC += (double)flowarr[n];
DC /= (double)Nf;

/* akis isaretinden DC degerini cikar */
for (n=0;n<Nf;n++)
    flowarr[n] -= (int)DC;

/*esik degerini bul*/
findmax(flowarr, (Nf/2), &max1, &locmax);
findmax(flowarr+(Nf/2), (Nf/2), &max2, &locmax);
maxavg = (max1+max2)/2;
threshold = maxavg*12/100;
/*akis isaretindeki gurutleri yoketmek icin
isareti yumusat*/
movavg(flowarr, Nf, 10);

/*isareti esikten gecir*/
for (n=0;n<Nf;n++) {
    if (abs(flowarr[n]) < threshold)
        flowarr[n]=0;
}

k=0;

/*sifira dusme ve sifirdan yukselme noktalarini bul*/
for (n=2;n<Nf-1;n++) {
    if (flowarr[n] == 0) {
        if (flowarr[n+1] == 0 && flowarr[n-1] == 0)
            continue;
        else if (flowarr[n+1] > 0 && flowarr[n-1] == 0) {
            temprec[k] = EXP_BEGIN; /*dissoluma baslangici*/
            temprec[k+1] = n;
        }
        else if (flowarr[n+1] < 0 && flowarr[n-1] == 0) {
            temprec[k] = INSP_BEGIN; /*icsoluma baslangici*/
            temprec[k+1] = n;
        }
    }
}

```

```

else if (flowarr[n+1] == 0 && flowarr[n-1] > 0) {
    temprec[k] = EXP_END; /*dissoluma sonu*/
    temprec[k+1] = n;
}
else if (flowarr[n+1] == 0 && flowarr[n-1] < 0) {
    temprec[k] = INSP_END; /*icsoluma sonu*/
    temprec[k+1] = n;
}

k+=2;

}
}

/*yalanci evreleri kontrol et ve ele*/
j=0;
for (n=0;n<k;n+=2) {
    if ((temprec[n]==EXP_BEGIN && temprec[n+2]==EXP_END) ||
        (temprec[n]==INSP_BEGIN && temprec[n+2]==INSP_END))
        if ((temprec[n+3]-temprec[n+1]) > guardband) {
            phaserec[j]=temprec[n];
            phaserec[j+1]=temprec[n+1];
            phaserec[j+2]=temprec[n+2];
            phaserec[j+3]=temprec[n+3];
            j+=4;
        }
}
for (n=j;n<Np;n++) phaserec[n] = 0; /*idizinin gerisini sifirila doldur*/
DisposPtr(temprec);
}

```

```
/*****< SINIFLANDIRMA ISLEVLERI >*****/
```

Asagidaki islevler dayanak kutuphanenin sikistirilmasinda, ilklenmesinde ve bir cercevenin siniflandirilmasinda kullanan islevlerdir.

```
*****/
```

```
#include <stdio.h>
#include <storage.h>
#include "cadold.h"
```

```
extern int frlen, order;
extern int numofCase, K, overlap;
extern coef_set *start;
```

```
/*-----
InitRefLib : Bu islev dayanak kutuphaneyi her
siniftan K+1 tane LPC vektoru ile ilkler. İlk-
leme sirasinda akis evreleri dikkate alinmak-
tadir. 'samparr[]', 'Ns' uzunlugunda solunun
sesi dizisi, 'flowarr[]', 'Nf' uzunlugunda
akis isareti dizisi, 'Case' denegin durumudur.
Islevde tanimlanmayan dizi ve degiskenler ana
programda global olarak tanimlanmislardir.
-----*/
```

```
void InitRefLib(samparr, flowarr, Nf, Case)
int *samparr, *flowarr, Case;
unsigned Nf;
```

```
{
    int i, j, *ifr, phase;
    unsigned frbegin, firstpt, lastpt;
    coef_set *data;
    unsigned phaserec[200];
    float *cor, *ffr, *par, *LPC, error;

    cor = (float*)NewPtr((order+1)*sizeof(float));
    ifr = (int*)NewPtr(frlen*sizeof(int));
    ffr = (float*)NewPtr(frlen*sizeof(float));
    par = (float*)NewPtr((order+1)*sizeof(float));
    LPC = (float*)NewPtr((order+2)*sizeof(float));

    if (!cor||!par||!LPC||!ifr||!ffr) {
```

```

    printf("Bellek kalmadi!!!");
    ExitToShell();
}
RemoveLib(); /*eski day. kutuphaneyi sil*/
InspExp(flowarr, Nf, phaserec, 200); /*akis evresi ayrimi*/
for (phase=EXP; phase<=INSP; phase++) {
    j=0;
    if (phase==EXP) {
        while(phaserec[j] && phaserec[j]!=EXP_BEGIN) j+=2;
        firstpt=phaserec[j+1]; /*baslangic noktasi*/
        while(phaserec[j] && phaserec[j]!=EXP_END) j+=2;
        lastpt=phaserec[j+1]; /*bitis noktasi*/
    }
    else if (phase==INSP) {
        while(phaserec[j] && phaserec[j]!=INSP_BEGIN) j+=2;
        firstpt=phaserec[j+1]; /*baslangic noktasi*/
        while(phaserec[j] && phaserec[j]!=INSP_END) j+=2;
        lastpt=phaserec[j+1]; /*bitis noktasi*/
    }

    for (j=0; j<(K+1); j++) {
        frbegin = firstpt + j*(lastpt-firstpt)/(K+1);
        GetFrame(ifr, samparr, frbegin, frlen);
        Scale(ifr, ffr, frlen);
        Hamming(ffr, frlen);
        AutoCor(ffr, frlen, order, cor);
        Lpc(order, cor, LPC-1, par, &error);
        data = (coef_set)NewPtr(sizeof(coef_set));
        if (!data) {
            printf("Bellek kalmadi!!!");
            ExitToShell();
        }
        data->subjCase = Case;
        data->flowphase = phase;
        for (i=0; i<order; i++)
            data->lpc[i] = LPC[i];
        start = AddLib(data, start);
    }
}
SaveLib();
DisposPtr(cor); DisposPtr(par);
DisposPtr(ffr); DisposPtr(ifr);
DisposPtr(LPC);
libcheck = TRUE;
}

```

```

/*-----*/
ClassifyFrame : Bu islev hangi akis evresinden
oldugu bilinen bir cerceveyi K-EYK kurali ve
Itakura uzaklik olcusunden yararlanarak sinif-
landirmaktadir. Islev ayni zamanda kendisini
cagiran programa cercevenin lpc katsayilarini
da gondermektedir.
/*-----*/

int ClassifyFrame(frame, LPC, numofCase, phase)
float *frame, *LPC;
int numofCase, phase;
{
    int i, vecnum, Case, num;
    float *buflpc, product;
    float *distance, *err, sqerr;

    distance = (float*)NewPtr((numofCase+2)*sizeof(float));
    buflpc = (float*)NewPtr((order+1)*sizeof(float));

    if (!distance || !buflpc) {
        printf("Bellek kalmadi!!!");
        ExitToShell();
    }
    AutoCor(frame, frlen, order, cor);
    Lpc(order, cor, LPC-1, par, &sqerr);
    for (Case=1; Case<=numofCase; Case++) {
        /*dayanak kutuphanedeki 'Case' durumun da
        olanlari 'phase' evrelerine ait
        lpc vektor sayisini bul */
        num = numof(Case, phase);
        err = (float*)NewPtr((num+1)*sizeof(float));
        /*cerceveyi dayanak kutuphanedeki lpc vektorleri ile suz
        ve her vektor icin ongoru hatasini hesapla*/
        for (vecnum=1; vecnum<=num; vecnum++) {
            /*lpc vektorunu buflpc[]'ye kopyala*/
            ReadCoef(Case, phase, vecnum, buflpc);
            err[vecnum] = FindError(buflpc, frame);
        }

        /* ilk K minimum hatanin carpimini bul */
        product = 1.0;
        Sort(num, err);
        for (i=1; i<=K; i++) product *= err[i];
    }
}

```

```

        /* cercevenin 'Case' sinifina olan uzakligini hesapla */
        distance[Case] = log10(product)-K*log10(sqerr);
        DisposPtr(err);
    }
    DisposPtr(bufipc);
    /*en kucuk uzakligi veren sinifin indisini dondur */
    return(MinOf(distance));
}

/*-----*/
CondenseRefLib : Bu islev InitRefLib islevi
ile ilklenen dayanak kutuphaneyi, sikistiril-
mis K-EYZ kuralina gore sikistirmektedir.
/*-----*/
int CondenseRefLib (samparr, flowarr, Nf, Case)
int *samparr, *flowarr, Case;
unsigned Nf;

{
    int i, j, *ifr, phase, decision;
    int addnum, discardnum, k;
    unsigned frbegin, firstpt, lastpt;
    coef_set *data;
    unsigned phaserec[200];
    float *cor, *ffr, *par, *LPC, error;

    cor = (float*)NewPtr((order+1)*sizeof(float));
    ifr = (int*)NewPtr(frlen*sizeof(int));
    ffr = (float*)NewPtr(frlen*sizeof(float));
    par = (float*)NewPtr((order+1)*sizeof(float));
    LPC = (float*)NewPtr((order+2)*sizeof(float));

    if (!cor||!par||!LPC||!ifr||!ffr) {
        printf("Bellek kalmadi!!!");
        ExitToShell();
    }
    InspExp(flowarr, Nf, phaserec, 200);
    addnum=discardnum=0;
    for (phase = EXP; phase <= INSP; phase++) {
        j=0;
        do {
            if (phase==EXP) {
                while(phaserec[j] && phaserec[j]!=EXP_BEGIN) j+=2;

```

```

        firstpt=phaserec[j+1];
        while(phaserec[j] && phaserec[j]!=EXP_END) j+=2;
        lastpt=phaserec[j+1];
    }
    else if (phase==INSP) {
        while(phaserec[j] && phaserec[j]!=INSP_BEGIN) j+=2;
        firstpt=phaserec[j+1];
        while(phaserec[j] && phaserec[j]!=INSP_END) j+=2;
        lastpt=phaserec[j+1];
    }

    for (i=0;;i++) {
        data = (coef_set*)NewPtr(sizeof(coef_set));
        if (!data) {
            printf("Bellek kalmadi!!!");
            ExitToShell();
        }
        frbegin = (firstpt+overlap)+i*(frlen-overlap);
        if (frbegin+frlen < lastpt) {
            GetFrame(ifr,samparr,frbegin,frlen);
            Scale(ifr,ffr,frlen);
            Hamming(ffr,frlen);
            decision = ClassifyFrame(ffr,LPC,numofCase,phase);
            if (decision != Case) {
                ++addnum;
                data->subjCase = Case;
                data->flowphase = phase;
                for (k=0;k<order;k++)
                    data->lpc[k] = LPC[k];
                start = AddLib(data,start);
            }
            else if (decision == Case){
                ++discardnum;
            }

        }
        else
            break;
    }
    }while(phaserec[j+2] != 0);
}
return(addnum); /* kutuphaneye eklenenlerin sayisi */
}

```

```
/*****< VERI TABANI ISLEVLERI >*****/
```

Asagida verilen islevler cift bagli liste yapısına sahip bir veri tabani olusturulmasında kullanılmaktadır. Dayanak kutuphanesi bu veri tabanından yararlanarak olusturulmaktadır.

```
*****/
```

```
#include <stdio.h>
#include <storage.h>
#include "cadold.h"
```

```
extern coef_set *start;
extern coef_set *last;
```

```
/*-----*/
```

AddLib : Bu islev 'coeff set' turundeki 'data' veri hucresini dayanak kutuphaneye ekler ve dayanak kutuphanenin en basindaki veri hucresinin adresini dondurur.

```
/*-----*/
```

```
coef_set *AddLib(data, top)
coef_set *data, *top;
```

```
{
```

```
    coef_set *old, *tp;
```

```
    tp = top;
```

```
    if (last==NULL) { /* listedeki ilk eleman */
```

```
        data->next = NULL;
        data->prior= NULL;
        last=data;
        return data;
    }
```

```
    old = NULL;
```

```
    while (tp) {
```

```
        if (!(tp->subjCase == data->subjCase && tp->flowphase == data->flowphase)) {
```

```
            old = tp;
            tp = tp->next;
```

```
        }
```

```
    else {
```

```

        if (tp->prior) { /* orta eleman */
            tp->prior->next=data;
            data->next=tp;
            data->prior=tp->prior;
            tp->prior=data;
            return top;
        }
        data->next = tp; /* yeni ilk eleman */
        data->prior= NULL;
        tp->prior = data;
        return data;
    }
}
old->next = data; /* sona koy */
data->next = NULL;
data->prior= old;
last = data;
return top;
}

/*-----*/
SaveLib : Bu islev tum dayanak kutuphaneyi
          "dayanak kutuphane" adli dosyaya kaydeder.
/*-----*/
void SaveLib()

{
    FILE *fp;
    coef_set *data;

    if ((fp=fopen("dayanak kutuphane", "wb"))==NULL) {
        printf("Dayanak dosya acilamiyor");
        ExitToShell();
    }
    printf("Dayanak kutuphane kaydediliyor....");

    data=start;
    while(data) {
        fwrite(data, sizeof(coef_set), 1, fp);
        data=data->next;
    }
    fclose(fp);
}

```

```

/*-----*/
LoadLib : Bu islev dayanak kutuphaneyi "daya-
nak kutuphane" adli dosyadan okur.
/*-----*/
int LoadLib()

{
    FILE *fp;
    coef_set *data, *temp=NULL;

    start = (coef_set*)NewPtr(sizeof(coef_set));
    if (!start) {
        printf("Bellek yetmiyor!!");
        ExitToShell(0);
    }
    if ((fp=fopen("dayanak kutuphane", "rb")) != NULL) {
        /* katsayi vektorlerini oku */
        data = start;
        while (!feof(fp)) {
            if(!fread(data, sizeof(coef_set), 1, fp)) break;
            /*bir sonraki icin bellek ayir*/
            data->next=(coef_set*)NewPtr(sizeof(coef_set));
            if (!data->next) {
                printf("Bellek yetmiyor!!");
                ExitToShell();
            }
            data->prior=temp;
            temp = data;
            data = data->next;
        }
        temp->next=NULL;
        last=temp;
        start->prior=NULL;
        fclose(fp);
        return TRUE;
    }
    else return FALSE;
}

```

```

/*****< YARDIMCI ISLEVLER >*****/

```

```

#include <math.h>

```

```

#include "cadold.h"

```

```

extern coef_set *start;

```

```

extern coef_set *last;

```

```

/*-----*/

```

```

void GetFrame(ifr, samparr, location, N)

```

```

int *ifr, *samparr, N;

```

```

unsigned location;

```

```

/* bu islev ornek dizisinin belirli bir yerin
   den bir cercevelik veri alır */

```

```

{
    unsigned i;
    for (i=location; i<(location+N); i++)
        ifr[i-location] = samparr[i];
}

```

```

/*-----*/

```

```

void Hamming(ffr, N)

```

```

float *ffr;

```

```

int N;

```

```

/* bu islev cerceveyi Hamming penceresi ile
   ağırlıklandırır. */

```

```

{
    int i;

    for (i=0; i<N; i++)
        ffr[i] = (0.54-0.46*cos((2*PI*i)/(N-1)));
}

```

```

/*-----*/

```

```

void Scale(arrin, arrout, N)

```

```

int *arrin, N;

```

```

float *arrout;

```

```

/* bu islev 2048, -2048 arasında olan tamsayı
   veriyi -1,1 arasına ölçekler. */

```

```

{
    int i;

    for (i=0; i<N; i++)
        arrout[i] = arrin[i]/2048.0;
}

/*-----*/
float FindError(buf1pc, ffr)
float *buf1pc, *ffr;

/* bu islev ffr[] cercevesi buf1pc[] katsayi-
   lari ile suzuldugunde ortaya cikan ongoru
   hatasini hesaplar. */

{
    register int i, j;
    float toterr=0.0, sum;
    float *bufr;

    bufr = (float*)NewPtr((order+frlen)*sizeof(float));

    for (i=0; i<order; i++) bufr[i] = 0; /* ilk sartlar sifir */

    for (i=order; i<frlen; i++) {
        sum = 0.0;
        bufr[i] = ffr[i-order];
        for (j=i; j<=order; j++)
            sum += buf1pc[j-i]*bufr[i-j];
        toterr += (bufr[i]+sum)*(bufr[i]+sum);
    }

    DisposPtr(bufr);
    return(toterr);
}

/*-----*/
int WinOf(distance)
float *distance;

/* bu islev 'distance[]' dizisinin en kucuk
   elemaninin indisini bulur. Bu indis
   ClassifyFrame() islevinin cerceveyi atadigi

```

```

sinifin indisine karsi gelir. */

{
    float min;
    int i, index=-1;
    min = distance[i];
    for (i=2; i<=numofCase; i++) {
        if (distance[i]<min) {
            min = distance[i];
            index = i;
        }
    }
    return(index);
}

/*-----*/
void ReadCoef(Case, phase, vecnum, buflpc)
int Case, phase, vecnum;
float *buflpc;

/* bu islev dayanak kutuphanedeki bir lpc vekt-
runu buflpc[]'ye kopyalar. */

{
    coef_set *data;
    int i;

    data = start;
    while (!(data->subjCase == Case && data->flowphase == phase))
        data = data->next;
    while (--vecnum != 0)
        data = data->next;
    for (i=0; i<order; i++)
        buflpc[i] = data->lpc[i];
}

/*-----*/
int NumOf(Case, phase)
int Case, phase;

/*bu islev belirlenen evre ve siniftaki lpc
vektorlerinin sayisini hesaplar */

```

```

{
    int num=0;
    coef_set *tp;

    tp = start;

    while (!(tp->subjCase == Case && tp->flowphase == phase) && tp)
        tp = tp->next;

    while (tp->subjCase == Case && tp->flowphase == phase && tp) {
        ++num;
        tp = tp->next;
    }
    return (num);
}

/*-----*/
void RemoveLib()

/* bu islev dayanak kutuphaneyi bellekten ve
diskten siler */

{
    coef_set *old;

    if (libcheck==TRUE) {
        eraseplot();
        printf("Eski dayanak kutuphane silinsin mi? (E/H) ");
        if (toupper(getch())=='E') {
            /* ayrilan bellegi bosalt */
            while (start) {
                old = start;
                start = old->next;
                if (start) start->prior = NULL;
                else last = NULL;
                DisposPtr(old);
            }
            remove("dayanak kutuphane");
            start = (coef_set*)NewPtr(sizeof(coef_set));
        }
        libcheck = FALSE;
    }
}

```

```

/*-----*/
void MovAvg(inarr, N, M)
int *inarr, N;
unsigned M;

/* bu islev giris dizisinin yuruyen
ortalamasini hesaplar */

{
    unsigned i, j;
    int *temp;
    int sum;

    temp = (int*)NewPtr(sizeof(int)*(N+M));
    if (!temp) {
        printf("Bellek kalmadi!!!");
        ExitToShell();
    }
    for (i=0; i<M/2; i++)
        temp[i]=temp[i+M+M/2]=0;
    for (i=M/2; i<M/2+M; i++)
        temp[i]=inarr[i-M/2];
    for (i=0; i<N; i++) {
        sum=0;
        for (j=i; j<i+M; j++)
            sum += temp[j];
        inarr[i] = sum/M;
    }
    DisposPtr(temp);
}

/*-----*/
void FindMax(arr, N, ymax, locmax)
int *arr, N, *ymax, *locmax;

/* bu islev tamsayi bir dizinin maks. elemanini
ve bu elemanın yerini hesaplar */
{
    int i;
    int max=0;

    for (i=0; i<N; i++) {
        if (arr[i] >= max) {
            max = abs(arr[i]);

```

```
        ilocmax=i;  
    }  
    }  
    iymax=imax;  
}
```



```
/******< CADOLD.H >******/
```

```
#define EXP      1
#define INSP     2
#define TRUE     1
#define FALSE    0
#define HEALTHY  1
#define PNEUMONIA 2
#define INSP_BEGIN 1
#define INSP_END 2
#define EXP_BEGIN 3
#define EXP_END 4
#define PI 3.1415926
```

```
struct cell {
    int flowphase; /*akis evresi*/
    int subjCase; /*denegin durumu(saglikli,zaturree)*/
    float lpc[20]; /*LPC Katsayilari*/
    struct data *next; /*bir sonraki hucre*/
    struct data *prior; /*bir onceki hucre*/
};
```

```
typedef struct cell coef_set;
```

```
/* islev prototipleri */
void MainMenu(void);
void Condense(void);
void Initialize(void);
void ClassifySubject(void);
void SaveLib(void);
void AutoCor(float*, int, int, float*);
void Lpc(int, float*, float*, float*, float*);
void InspExp(int*, unsigned, unsigned*, unsigned);
void InitRefLib(int*, int*, unsigned, int);
void GetFrame(int*, int*, unsigned, int);
void Hamming(float*, int);
void Scale(int*, float*, int);
void MovAvg(int*, unsigned, int);
void FindMax(int*, int, int*, int*);
void ReadCoef(int, int, int, float*);
int ClassifyFrame(float*, float*, int, int);
int CondenseRefLib(int*, int*, unsigned, int);
```

```
int    MinOf(float*);  
int    NumOf(int, int);  
int    LoadLib(void);  
float  FindError(float*, float*);  
coef_set *AddLib(coef_set*, coef_set*);
```

## ÖZGEÇMİŞ

1967'de Ankara'da doğdu. İlköğrenimini Selimiye İlkokulu'nda, ortaokul ve lise öğrenimini Kadıköy Anadolu Lisesi'nde tamamladı. 1985 yılında İ.T.Ü. Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği bölümüne girdi. 1989 yılında lisans eğitimi tamamlayarak, aynı bölümde yüksek lisans eğitimine başladı. 1990'dan beri İ.T.Ü. Elektrik-Elektronik Fakültesi, Haberleşme Ana Bilim Dalında Araştırma Görevlisi olarak çalışmaktadır.