

T.C  
FIRAT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

TÜRKÇEDEN SEÇİLMİŞ BİR GRUP KELİMENİN  
ROBOT KOLU KONTROLÜNDE  
KULLANILMASI

Fikret Ata

DOKTORA TEZİ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ  
ANABİLİM DALI

Bu Tez , ..... Tarihinde, Aşağıda Belirtilen Jüri Tarafından  
Oybirliği / Oyçokluğu ile Başarılı / Başarısız  
Olarak Değerlendirilmiştir.

(İmza)

(İmza)

(İmza)

Danışman

Üye

Üye

ÖZET

Doktora Tezi

TÜRKÇEDEN SEÇİLMİŞ BİR GRUP KELİMENİN  
ROBOT KOLU KONTROLÜNDE  
KULLANILMASI

Fikret Ata

Fırat Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektrik-Elektronik Mühendisliği Anabilim Dalı

1996, Sayfa:114

Bu tezde, Türkçeden seçilmiş bir grup kelimenin kullanılması ile robot kolunun kontrolü yapılmıştır.

Seçilen belirli kelimelerin, hazırlanan yazılımlarla, konuşma örneklerinin doğrusal öngörümüne dayalı ve konuşmacıya bağımlı bir ayırık sözcük tanıma algoritması ile bilgisayar desteğinde tanınması gerçekleştirilmiştir.

Tanınan bu komutlar vasıtası ile, robot kolunun hareket denklemlerini de kullanan bir başka yazılım desteğinde, step motorlarının adımları, dolayısı ile eklemlerin açılıarı ayarlanarak, robot kolunun istenilen koordinatlara gitmesi ve belirlenen yerde belirli işi yapması sağlanmıştır.

**ANAHTAR KELİMELER:** Ayrık sözcük tanıma, özilişki, doğrusal öngörü katsayıları, dinamik zaman bükümü, en yakın komşu kuralı, robot, robot kolu.



SUMMARY

Ph.D.Thesis

USE OF A GROUP WORD CHOSEN FROM TURKISH  
IN ROBOT ARM CONTROL

Fikret Ata

Firat University  
Graduate School of Naturel and Applied Sciences  
Department of Electrical and Electronics Engineering

1996, Page:114

In this thesis, the robot arm control has been preformed using a group word chosen from Turkish.

Chosen certain words, by the written softwares, speech samples base on the linear predictions and the computer aided recognition with an isolated word recognition algorithm depended on the speaker has been actualized.

By means of these recognized commands and with the help of another software which uses the motion equations of the robot arm, by adjusting the steps of the step motors and hence the joints angles, the position of the robot arm to the desired coordinates and in a certain place preforming a certain task have been ensured.

**KEY WORD:** Isolated word recognition, autocorrelation, linear predictive coding, dynamic time warping, nearest-neighbor decisionrule, robot, robot manipulator.



## **TEŐEKKÜR**

Bu alıőmanın ortaya ıkmasında yönetici ve yönlendirici olarak her zaman desteklerini saėlayan tez yöneticim Sayın Yrd.Do.Dr. Mustafa Poyraz'a teőekkür ederim.

Ayrıca alıőmalarım süresince gerek yardım eden gerekse manevi yönden destek olan bölümdeki arkadaşlarıma teőekkür ederim.

Son olarak, sabırla bana katlanan ve beni destekleyen aileme de minnet ve őükranlarımı sunarım.

## İÇİNDEKİLER

<b>ÖZET</b> .....	III
<b>SUMMARY</b> .....	V
<b>TEŞEKKÜR</b> .....	VII
<b>İÇİNDEKİLER</b> .....	VIII
<b>ŞEKİLLER LİSTESİ</b> .....	XII
<b>TABLolar LİSTESİ</b> .....	XIV
<b>SİMGELER LİSTESİ</b> .....	XV
<b>1. GİRİŞ</b> .....	1
<b>2. GENEL ANLAMDA İŞARET TANIMA</b> .....	3
2.1. İşaret Tanıma İşlemi.....	3
2.2. Deterministik sınıflandırma.....	5
2.3. İstatistiksel Sınıflandırma.....	6
<b>3. SÖZ TANIMA</b> .....	9
3.1. Giriş.....	9
3.2. Yalıtılmış Sözcük Tanımada Genel Model.....	10
3.3. Nitelik Ölçme.....	11
3.4. Doğrusal Öngörü Kodlaması.....	12
3.4.1. Genel Model.....	12

3.4.2. Otoregresif Model.....	13
3.5. Bütün Bir Nitelik Ölçücü.....	18
3.5.1. Blok Diyagram.....	18
3.5.2. Sayısal Örnek.....	20
3.6. Başlangıç ve Bitiş Noktalarını Belirleme.....	25
3.7. Ön dengeleme.....	28
3.8. Pencereleme.....	30
3.9. Özellik Çıkarma.....	35
3.10. Dinamik Zaman Bükümü.....	37
3.10.1 Doğrusal Bükme.....	37
3.10.2. Doğrusal Olmayan Bükme.....	38
<b>4. ROBOTLAR.....</b>	<b>43</b>
4.1. Giriş.....	43
4.2. Robotların Kullanım Alanları.....	44
4.3. Robotların Yapısı.....	44
4.4. Kontrol Sistemleri.....	45
4.5. Koordinat Sistemleri.....	45
4.6. Robotların Hareket Uzayı.....	46
4.7. Tahrik Sistemleri.....	47
4.8. Tahrik Kaynakları.....	47
<b>5. KURULAN SİSTEME GENEL BAKIŞ.....</b>	<b>48</b>
5.1. Mekanik Donanım.....	49
5.1.1. Kullanılan Robotun Yapısı.....	50
5.2. Elektronik Donanım.....	53
5.2.1. Alçak Geçiren Filtre ve Kuvvetlendirici.....	53
5.2.2. PCL-812 Arabirim Kartı.....	55
5.2.2.1. Giriş.....	55
5.2.2.2. PCL-812 Arabirim Kartı'nın Genel Özellikleri.....	55
5.2.2.3. Yazılım Desteği.....	56
5.2.2.4. Temel Adres Seçimi.....	57

5.2.2.5. Pin Bağlantıları.....	57
5.2.3. Terminal Bağlantı Kartı.....	58
5.2.4. Röle Kartı.....	58
5.2.5. Eklem Limitleri Arabirim Kartı.....	59
5.2.6. Step Motor Sürücü Kartı.....	61
5.2.7. Step Motor Arabirim Kartı.....	62
5.3. Step Motorlar.....	62
5.3.1. Sabit Miknatıslı Step Motorlar (PM).....	63
5.3.1.1. Step Motor Parametreleri.....	65
5.3.1.2. PM Step Motorlarda Kayma.....	67
5.3.2. Değişken Relüktanslı (VR) Step Motorlar.....	68
5.3.2.1. VR Step Motorun Çalışma Prensibi.....	69
5.3.3. Step Motorların Sürülmesi.....	70
5.3.3.1. Step Motor Uyarım Metotları.....	71
5.3.3.1.1. İki Fazlı Tek-Tek Uyarım.....	71
5.3.3.1.2. İki Fazlı İki Sargının Aynı Anda Uyarımı.....	73
5.3.3.1.3. Üç Fazlı Tek-Tek Uyarım.....	73
5.3.3.1.4. Üç Fazlı İki Sargının Aynı Anda Uyarımı.....	74
5.3.3.1.5. Dört Fazlı Tek-Tek Uyarım.....	75
5.3.3.1.6. Dört Fazlı İki Sargının Aynı Anda Uyarımı.....	75
5.3.3.2. Step Motorlarda Kaymayı Önleme Metotları.....	75
5.3.3.2.1. Mekanik Metot.....	76
5.3.3.2.2. Rezistif Metot.....	76
5.3.3.2.3. Kapasitif Metot.....	77
5.3.3.2.4. Transistörlü Step Motor Sürücü Devresi.....	77
5.4. Besleme Devresi.....	79
5.5. Koordinat Sisteminde Bir Noktanın Tanımlanması.....	80
<b>6. YAZILIM.....</b>	<b>82</b>
6.1. Öğrenme Kısmı Yazılımı.....	83
6.2. Test Kısmı Yazılımı.....	84
6.3. Kumanda Kısmı Yazılımı.....	84

<b>7. SONUÇLAR.....</b>	<b>86</b>
<b>KAYNAKLAR .....</b>	<b>88</b>
<b>EK - 1. SAA1027 Step Motor Sürücü Entegresinin Blok Diyagramı ve Bacak Bağlantıları.....</b>	<b>94</b>
<b>EK - 2. SAA1027'nin Giriş Çıkış İşaretleri ve Durum Diyagramı.....</b>	<b>95</b>
<b>EK - 3. Terminal Bağlantı Kartı I.....</b>	<b>96</b>
<b>EK - 4. Terminal Bağlantı Kartı II.....</b>	<b>97</b>
<b>EK - 5. Terminal Bağlantı Kartı Malzeme Yerleşim Planı.....</b>	<b>98</b>
<b>EK - 6. Step Motorun Teknik Resmi.....</b>	<b>99</b>
<b>EK - 7. Öğrenme Kısmı Yazılımı.....</b>	<b>100</b>
<b>EK - 8. Test Kısmı Yazılımı.....</b>	<b>101</b>
<b>EK - 9. Örnekalma Altprogramı.....</b>	<b>103</b>
<b>EK - 10. Başlangıç ve Bitiş Örneklerini Belirliyen Altprogram.....</b>	<b>105</b>
<b>EK - 11. Öndenge Altprogramı.....</b>	<b>108</b>
<b>EK - 12. Hampencere Altprogramı.....</b>	<b>108</b>
<b>EK - 13. Özilişki Altprogramı.....</b>	<b>108</b>
<b>EK - 14. DOK Altprogramı.....</b>	<b>110</b>
<b>EK - 15. Ozıldok Altprogramı.....</b>	<b>110</b>
<b>EK - 16. Hamming Window Katsayılarını Hesaplayıp Bir Kütüğe Yazan Program.....</b>	<b>111</b>
<b>EK - 17. Doğrusal Bükme Altprogramı.....</b>	<b>111</b>
<b>EK - 18. DZB Altprogramı.....</b>	<b>112</b>
<b>EK - 19. Farklılık Altprogramı.....</b>	<b>113</b>
<b>EK - 20. Karar Altprogramı.....</b>	<b>113</b>
<b>EK - 21. Depolama Altprogramı.....</b>	<b>114</b>

## ŞEKİLLER

Şekil 2.1. Genel Bir İşaret Tanıma İşlemi.....	4
Şekil 2.2. Bir Sınıflandırıcı.....	5
Şekil 3.1. Yalıtılmış Sözcük Tanıyıcının Blok Diyagramı.....	11
Şekil 3.2. D.Ö.K. Analizi Kullanan Nitelik Ölçücü.....	18
Şekil 3.3. İşaretin Zaman Çerçevesine Ayrılması.....	19
Şekil 3.4. Sayısal Örneğe İlişkin Blok Diyagramı.....	21
Şekil 3.5. $x(n)$ İşareti.....	22
Şekil 3.6. $f(n)$ İşareti.....	22
Şekil 3.7. $x(n)$ İşaretinin Frekans Spektrumu.....	23
Şekil 3.8. $f(n)$ İşaretinin Frekans Spektrumu.....	23
Şekil 3.9. Kalan Hataya Ait İşaret.....	24
Şekil 3.10. Kalan Hatanın Frekans Spektrumu.....	24
Şekil 3.11. $f(n)$ 'nin Gerçek ve Kestirilmiş Spektrumu.....	25
Şekil 3.12. $H(z)$ 'nin Sıfırı ve Kutbu.....	29
Şekil 3.13. Sayısal Filtrenin Frekans Cevabı.....	30
Şekil 3.14. Dikdörtgen Pencere ve Hamming Pencere.....	32
Şekil 3.15. Dikdörtgen Pencerenin Güç Spektrumu.....	32
Şekil 3.16. Hamming Pencerenin Güç Spektrumu.....	33
Şekil 3.17. Üst üste Binme Olmadan Pencereleme.....	34
Şekil 3.18. Üst üste Binme Durumunda Pencereleme.....	35
Şekil3.19. Bükme Yöntemlerinin Birlikte Kullanımı.....	38
Şekil 3.20. Dinamik Zaman Bükümü.....	39
Şekil 3.21. Yerel Patika Sınırlamaları.....	40
Şekil 5.1. SCARA Robotun Temsili Gösterimi.....	50
Şekil 5.2. Tutucu Kısım ve Pünomatik Şeması.....	51
Şekil 5.3. SCARA Tipi Robotun Görünümü.....	51

<b>Şekil 5.4.</b> Kurulan Sistemin Yazılım ve Donanım Blok Şeması.....	52
<b>Şekil 5.5.</b> Alçak geçiren Filtre.....	53
<b>Şekil 5.6.</b> Filtre Karakteristiği.....	54
<b>Şekil 5.7.</b> Kuvvetlendirici Devre Şeması.....	54
<b>Şekil 5.8.</b> PCL-812 Kartının Blok Diyagramı.....	56
<b>Şekil 5.9.</b> Digital Giriş ve Çıkış İçin Konnektör Bağlantıları.....	58
<b>Şekil 5.10.</b> Röle Kartı Konnektör Bağlantı Şemaları.....	59
<b>Şekil 5.11.</b> Röle Kartı Malzeme Yerleşim Planı.....	60
<b>Şekil 5.12.</b> Eklem Limitleri Arabirim Kartı.....	60
<b>Şekil 5.13.</b> Step Motor Sürücü Devresi.....	61
<b>Şekil 5.14.</b> Step Motor Arabirim Kartı.....	62
<b>Şekil 5.15.</b> 2 Faz İki Kutuplu Step Motor.....	63
<b>Şekil 5.16.</b> 4 Faz İki Kutuplu Step Motor.....	64
<b>Şekil 5.17.</b> Step Motorun Çalışma Prensi Şeması.....	65
<b>Şekil 5.18.</b> Disk Tipi Rotora Sahip Step Motor.....	66
<b>Şekil 5.19.</b> PM Step Motorlarda Kayma.....	67
<b>Şekil 5.20.</b> VR Motorun Prensi Şeması.....	68
<b>Şekil 5.21.</b> Üç Faz (VR) Değişken Relüktanslı Step Motor.....	69
<b>Şekil 5.22.</b> Tork-Hız Grafikleri.....	70
<b>Şekil 5.23.</b> Step Motor Sürücü Devresi Blok Şeması.....	71
<b>Şekil 5.24.</b> İki Fazlı Tek-Tek Uyarım.....	72
<b>Şekil 5.25.</b> Üç Fazlı Step Motorun Prensi Şeması.....	73
<b>Şekil 5.26.</b> Dört Fazlı Uyarım.....	74
<b>Şekil 5.27.</b> Kaymanın Önlenmesinde Rezistif Metot.....	76
<b>Şekil 5.28.</b> Kaymanın önlenmesinde Kapasitif Metot.....	77
<b>Şekil 5.29.</b> Step Motor Sürücü Devre Şeması.....	78
<b>Şekil 5.30.</b> Besleme Devresi Şeması.....	79
<b>Şekil 5.31.</b> Koordinat Sisteminde Bir Noktanın Tanımlanması.....	80

**TABLolar**

<b>Tablo 5.1.</b> Temel Adres Seçimi.....	57
<b>Tablo 5.2.</b> İki Fazlı Tek ve Aynı Anda Uyartım.....	72
<b>Tablo 5.3.</b> Üç Fazlı Tek-Tek ve Aynı Anda Uyartım.....	74
<b>Tablo 5.4.</b> Dört Fazlı Tek-Tek ve Aynı Anda Uyartım.....	75



**SİMGELER**

Çalışmada kullanılan simgeler her bölümün kendi içinde verildiğinden burada tekrar verilmesine gerek duyulmamıştır.



## 1.GİRİŞ

İşaret tanıma işlemi ve onun özel konularından biri olan söz tanıma, günlük yaşamda geniş uygulama alanları bulabilmektedir. Özellikle mikroişlemcilerin gelişmesi, bellek miktarları ve çalışma hızları açısından iyileşmeleri, söz tanıma problemi için de önemli gelişmeler sağlamaktadır. Zira söz tanıyan sistemler yeteneğine göre farklılık gösterse de geniş bellek alanı ve hızlı işlem yapmaya ihtiyaç duymaktadırlar ( Myers, C., Rabiner.L.R., ve Rosenberg, A.E., 1980).

Söz tanıma, en genel olarak, bir sözü pek çok aday arasından belirleyebilmektir. Sözün niteliği problemin karakterini de ortaya çıkarır. Örneğin, belli sayıda ve belirli sözcükler arasından hangisinin konuşulduğunu algılama problemi yalıtılmış sözcük tanıma olarak adlandırılır (Rabiner, L.R., ve Levinson,S.E., 1981). Böyle bir sistem aralarına yapay beklèmeler konulmuş bir dizi sözcüğü tanıyabilmelidir. Eğer bu yapay beklèmeleri kaldırır, ama konuşulan sözü harf harf seslendirirsek, bir adım daha ileri gidip bağlanmış sözcük tanıma gerçekleştirilmelidir. Buna belirgin bir örnek telefon numaralarını tanıyabilen sistemdir. Kısaca ifade edilen bu problemlerin yanında daha zor olan sürekli söz tanıma ve söz anlama diğer iki ana sorundur. Eğer bir sözcüğün dizisini harflerine ayırmak suretiyle seslendirmek yerine doğal yapısında söylemek istersek, problem sürekli söz tanıma olmaktadır ( Jelinek, F., 1976). Bununda üzerinde eğer söylenen cümlenin içerdiği anlamın belirlenmesi isteniyor ise, sistem söz-anlama işlemi de yapabilmelidir. Örneğin, “ Pamukkale “ denildiğinde ilk akla gelen bir yörenin adı olduğudur. Oysa bu bilgiye sahip olmayan bir söz-anlayıcı bu sözcüğü pamuktan yapılmış bir kale olarak anlamlandırabilir. Görülüyor ki sözün anlaşılması çok daha doğal bir işlemdir ve bu nedenle de en zor problemdir (Snyers, D., 1986).

Genel problemin zorluğu bir yana bırakılırsa, en temel sorun konuşulan bir sözün tanınabilmesidir. Bu temel problemle uğraşmak diğerleri için de fikir verici

olmaktadır. Bu nedenle tezin ikinci bölümlerinde söz tanımaya temel teşkil etmesi bakımından, deterministik ve istatistiksel yönden sınıflandırma işlemi ele alınmıştır.

Üçüncü bölümde, daha sonra konuşmacıya bağımlı sözcük tanımada kullanılacak olan, doğrusal öngörü kodlaması verilerek, sayısal bir örnekle desteklenmiştir (Makhoul, J., 1975).

Dördüncü bölümde, genel anlamda, robotlara ilişkin, kullanım alanları, kontrol ve koordinat sistemleri konusunda açıklamalar yapılmıştır (Schilling, J., 1990). Beşinci bölümde ise, robot kontrolünde sıklıkla kullanılan step motorlarla ilgili detaylı bilgiler verilmiş ve gerçekleştirilen sistemin ayrıntılarına girilmiştir. Altıncı bölümde, gerek söz tanıma işleminde ve gerekse robotun kontrolünde kullanılmak üzere hazırlanan yazılımlar konusunda açıklamalarda bulunulmuştur. Yedinci bölümde ise, bu çalışmadan elde edilen sonuçlar verilmiştir.



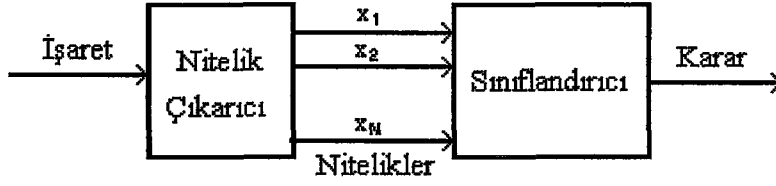
## 2. GENEL ANLAMDA İŞARET TANIMA.

### 2.1. İşaret Tanıma İşlemi.

İşaret tanıma işlemi, bir grup nesnenin, öznel bir bakış açısı altında, belirli sayıdaki kümeler içinde toplanması veya kısaca sınıflandırılması olarak anlatılabılır. Her bir sınıftaki nesnelere arasında kimi ortak öznel özellikler vardır. Başka bir deyişle, bu ortak özellikler değişik yorumlandıkça, nesnelere değişik biçimlerde sınıflandırmak olasıdır. Sözelimi; Türk alfabesindeki harfleri tanıyan bir tanıma işleminde yirmi dokuz sınıfa gereksinim duyulur. Buna karşılık, Türkçe harfleri Arapça'dakilerden ayırt etmek istersek yalnızca iki sınıf yeterlidir. Görüldüğü gibi önce ayrı sınıflandırılan harfler şimdi tek bir sınıf içindedir.

İşaret tanıma işlemi pek çok alanda kullanılabilir: parmak izlerinin ve yüzlerin kimliğinin belirlenmesi, yazılı metinlerin okunması, konuşmaların tanınması, elektrokardiogram işaretlerinin sınıflandırılması, genel hastalık tanısı, sismik işaretlerin sınıflandırılması, hedef saptama, hava kestirimi v.d.

En basit işaret tanıma yaklaşımlarından biri asal işaret ile karşılaştırmadır. Bunun için tanıyıcıdaki her bir sınıfa bir asal-ışaret belirlenip bellekte saklanır. Bu asal işaretlerin belirlenmesine, çoğu zaman bir öğrenme işlemi olarak bakılabilir. Bundan sonra, bir bilinmeyen işareti, bütün asal işaretler ile karşılaştırılır. Bu karşılaştırmada belirli bir benzerlik ölçütü kullanılır. Böylece bilinmeyen işaretin en yakın olduğu yada en çok benzediği asal-ışaretin ait olduğu sınıfta karar verilir. Bu temel yöntemin birtakım sakıncaları vardır: her işaret sınıfına iyi bir asal-ışaret bulunmayabilir ve uygun bir benzerlik ölçütü belirlenemeyebilir. Eğer bir sınıfa ait işaretler arasında büyük değişme yada bozulmalar olabiliyorsa, bu sakınca daha da önem kazanır. Sözelimi; el yazılarının tanınmasında benzer zorluklar ile karşılaşılır.



Şekil 2.1. Genel Bir İşaret Tanıma İşlemi.

Böyle bir işaretle, basitlik amacı ile asal işaretler veri biçiminde saklanır. El yazısının çeşitliliği nedeniyle karşılaşılan işaretlerde oldukça fazla değişiklik gösterebilir.

Bu soruna daha ileri bir yaklaşım “nitelik ölçmeleri” yapmaktır. Aynı sınıfa ait işaretlerden elde edilen ölçümleri yada kısaca nitelikler arasında ayırım olmadığı yada çok az olduğu düşünülür. Daha açık bir anlatım ile nitelikler aynı sınıfta bulunan işaretlerdeki değişimlere karşı daha az duyarlıdır. Bu durumda işaret tanıma işlemi iki alt soruna ayrılabilir. Bunlardan birisi, giriş işaretlerinden hangi ölçmelerin yapılacağıdır. Ölçüm seçimi, kavramsal bir yaklaşımın yanında pek çok pratik öğeye de bağlıdır. Sözgelimi ölçmelerin gerçekleştirilebilirliği, kullanılacak ölçmeler için gerekli bellek miktarı ve tanıma zamanı gibi. İkinci sorun, elde edilen niteliklerden tanıma işleminin gerçekleştirilmesidir. Bu düşünceler altında Şekil 2.1. de genel bir işaret tanıma işlemi gösterilmiştir.

Burada bir  $x = (x_1, \dots, x_n)$  vektörü giriş işaretini simgeliyor ise, bu vektörün bileşenleri de nitelik ölçümleridir. Sınıflandırıcı, nitelikleri giriş olarak alan bir karar kuralı yardımı ile çalışmaktadır.

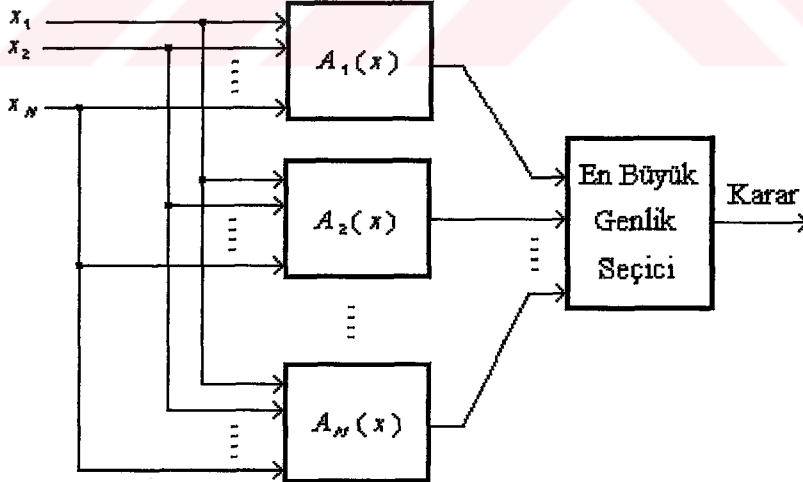
Eğer asal-ışaretler nitelikler cinsinden bellekte saklanacak olursa ve özel bir sınıflandırma ölçütü kullanılacak olursa, asal işaret ile karşılaştırma yaklaşımı nitelik-çıkarma yaklaşımının bir özel durumu gibi düşünülebilir.

## 2.2 Deterministik Sınıflandırma

İşaretlerin sınıflandırılması kavramı, nitelik uzayının bölgelere ayrılması yada nitelik uzayından karar uzayına bir fonksiyon olarak düşünülebilir. Her bir giriş işaretinden  $N$  sayıda nitelik ölçümü yapılmış olsun. Her  $N$  nitelik bir vektöründe toplansın. Böylece bir deterministik vektörü,  $N$  boyutlu bir  $U$  nitelik uzayında bir nokta olur.

$$x = (x_1, \dots, x_n) \quad x \in U \quad (2.1.)$$

Sınıflandırma sorunu her  $x$  vektörünü olası  $m_i, i = 1, \dots, M$  sınıftan birine koymaktır. Böylece nitelik uzayı hepsi birbirini dışında olan  $M$  sayıda bölgeye ayrılmış olmaktadır. Matematiksel olarak sınıflandırma işlemi ayrılmış olmaktadır. Matematiksel olarak sınıflandırma işlemi ayırıcı fonksiyonlar yardımı ile ifade



Şekil 2.2. Bir Sınıflandırıcı.

edilebilir. Bu durumda  $A_j(x)$  ayırıcı fonksiyonu her  $j = 1, \dots, M$  için tasarlanmalıdır. Bu tasarı için şu ölçüt kullanılabilir.  $A(x)$  öyle olmalıdır ki,  $M_j$  sınıfındaki her  $x$  için aşağıdaki koşul sağlansın;

$$A_j(x) \geq A_i(x) \quad i, j = 1, \dots, M \quad (2.2.)$$

Böylece,  $U$  uzayındaki bölgeler arasındaki sınır aşağıdaki gibi bulunur.

$$A_j(x) - A_i(x) = 0 \quad (2.3.)$$

Bu ifade karar sınırı adını alır ve  $m_j$  ile  $m_i$  sınıflarını simgeleyen iki bölge arasındaki sınırdır. Ayrıca fonksiyonları ile çalışan bir sınıflandırıcının genel yapısı Şekil 2.2. de gösterilmiştir.

Bu durumda bir sınıflandırıcının tasarımı ayırıcı fonksiyonların belirlenmesinden ibaret olmaktadır.

### 2.3. İstatistiksel Sınıflandırma.

Bundan önceki bölümde  $x_1, x_2, \dots, x_N$  ölçümleri deterministik olarak verildi, oysa pek çok durumda bir sınıftaki işaretler büyük değişiklikler gösterir ve ölçmelerin alınması sırasında gürültü etkisi vardır. O zaman  $x_i$ 'leri rastlantı değişkeni olarak düşünmek yerinde olur. Yani  $x_i$ ,  $i$ 'inci niteliğin gürültülü ölçümüdür. Bir  $x$  vektörünün bir  $m_i$  sınıfında ortaya çıkması ile ilgili olan  $p(x/m_i)$  olasılık yoğunluk fonksiyonu ve  $p(m_i)$  olasılığı her  $i = 1, \dots, M$  için bilinsin. Bu bilgiler altında tanıyıcı sınıflandırma işlemini, tanıyamama olasılığını en küçük yapmak üzere gerçekleştirsin. Bu durumda tanıma işlemi yada sınıflandırma istatistiksel karar verme sorununa dönüşür (M hipotezin sınanması) (Whalen, 1971).

Bir  $x$  nitelik vektörü  $m_i$  sınıfında ise bunu  $x \sim m_i$  ile gösterelim. Bu durumda bir  $K(\cdot)$  karar fonksiyonu şöyle tanımlanabilir.

$$K(x) = k_i \quad \text{ise} \quad x \sim m_i \quad (2.4)$$

Giriş işareti gerçekte  $M_g$  sınıfına ait iken, sınıflandırıcının onu bir  $M_i$  sınıfına koyması ile ortaya çıkan kaybı  $K(m_g, k_i)$  ile gösterelim. Koşullu risk şöyledir.

$$x \sim m_i \quad \text{için} \quad r(m_i, k) = \int_U K(m_i, k) \cdot p(x | m_i) dx. \quad (2.5)$$

Burada  $U$ , nitelik uzayıdır. Verilmiş bir  $P = (p(m_1), \dots, p(m_M))$  kümesi için ortalama kayıp yada risk aşağıdaki gibi hesaplanabilir.

$$R(P, k) = \sum_{i=1}^M p(m_i) r(m_i, k). \quad (2.6)$$

(2.5)'i (2.6)'da yerine koyarsak

$$R(P, k) = \sum_{i=1}^M p(m_i) \int_U K(m_i, k) p(x | m_i) dx \quad (2.7)$$

$$= \int_U \sum p(m_i) K(m_i, k) p(x | m_i) dx.$$

Burada

$$r_x(P, k) = \sum_{i=1}^M p(m_i) K(m_i, k) p(x | m_i) / p(x) \quad (2.8)$$

dersek (2.7) şöyle olur.

$$R(P, k) = \int_U p(x) r_x(P, k) dx. \quad (2.9)$$

$r_x(P, k)$  terimi,  $K$  kararının posteriori koşullu ortalama kaybı olarak tanımlanır. Sorun  $R(P, k)$ 'yi en küçük yapacak yada  $P$  kümesinin bilinmemesi halinde,  $r(m_i, k)$  koşullu kaybının en büyük değerini en küçük yapacak biçimde (Whalen, 1971) bir karar kuralı belirlemektir. Ortalama kaybı en küçük yapan karar kuralı Bayers kuralı olarak adlandırılır. (2.9)'dan her bir  $x_i$  'yi ayrı ayrı düşünmek ve  $r_x(P, k)$ 'yi en küçük yapma anlamında optimum ise

$$\forall k \quad \text{için} \quad r_x(P, k_0) \leq r_x(P, k) \quad (2.10)$$

dır. Bu ifade (2.8) ile yeniden yazılırsa;

$$\sum_{i=1}^M K(m_i, k_0) p(m_i) p(x|m_i) \leq \sum_{i=1}^M K(m_i, k) p(m_i) p(x|m_i) \quad (2.11)$$

olur. Eğer (0,1) kayıp fonksiyonu seçerek

$$K(m_i, k_j) = 0 \quad i = j \quad ; 1 \quad i \neq j \quad (2.12)$$

ortalama kayıp aynı zamanda tanıyamama olasılığı olur. Bu durumda Bayers karar kuralı şöyledir.

Eğer her  $j = 1, \dots, M$  için

$$p(m_i) p(x|m_i) \geq p(m_j) p(x|m_j) \quad \text{ise}$$

$$x \sim m_i \quad \text{olarak karar ver.} \quad (2.13)$$

Eğer  $m_i$  ve  $m_j$  sınıfları arasında bir olurluk oranı tanımlanırsa

$$\lambda = p(x|m_i) / p(x|m_j) \quad (2.14)$$

(2.13) karar kuralı yeniden yazılabilir.

$$\text{Eğer her } j = 1, \dots, M \text{ için } \lambda \geq p(m_j) / p(m_i) \quad \text{ise}$$

$$x \sim m_i \quad \text{dir.}$$

### 3. SÖZ TANIMA.

#### 3.1. Giriş.

Söz tanıma başlıca dört bölüme ayrılabilir. Yalıtılmış sözcük tanıma,bağlanmış sözcük tanıma,sürekli sözcük tanıma ve söz anlama.

Yalıtılmış sözcük teriminin iki anlamı olabilir. Birincisi, tanımak istenen, yalnız bir tek sözcüktür. İkincisi, bir dizi sözcük ard arda söylenir. Ancak bu süre içinde sözcükler arasında, konuşan kişi tarafından yapay beklentiler konmuştur. Örnek olarak ses komutlu telefon ağı verilebilir. Burada, aranmak istenilen abonenin numarası veya abonenin kayıtlı adı, rakamlarına veya harflerine ayırmak suretiyle, tanıyıcıya okunur. Tanıyıcının sözlüğü alfabenin harflerinden veya (0-9) arası sayılardan oluşmuştur. Her bir harfin veya rakamın okunmasından sonra bir uyarı sesi alınarak ikinci harf veya rakamın okunmasına geçilebilir. Numara veya adın tamamı verildikten sonra tanıyıcı tanıdığı sözcük dizisini çağıran aboneye tekrarlayarak sınamasını yapar. Hata olmaması durumunda karşı abone ile bağlantı kurulur. Yalıtılmış sözcük tanıma, genel söz tanıma sorununun kolay olan en uç noktasıdır.

Söz anlama ise en zor olan diğer uç noktasıdır. Bazen sürekli söz tanıma sorunu ile söz anlama arasındaki ayrımı kestirmek güçtür. Yine de söz anlama işlemi, söz tanımanın üstünde birtakım görevleri üstlenmiştir. Sözü anlayan dizge, konuşulan bir cümlenin ne olduğunu belirledikten sonra onun anlamını kestirmek zorundadır. Bunun için örneğin söze ilişkin akustik işaret alınıp, cümlenin hepsi fonemlerine (Snyers,1986) ayrıldıktan sonra cümle içindeki sözcüklerin eldeki fonem dizisinden belirlenmesi gerekmektedir. Elde edilen sözcüklerden oluşan cümlenin geçerliliği yada kullanılan bir dilin dilbilgisine uygunluğu da belirlenmelidir. Bundan sonra tespit edilmiş olan cümlenin içerdiği anlamı çıkarmak yada sözcüğü vermek istediği komutu belirlemek gerekir. Böyle bir işlem kendisine konuşularak yazı yazan bir yazıcıda veya bir robota sözle komut verilerek belirli işlemlerin yaptırılmasında kullanılabilir.

Yalıtılmış sözcük tanıma tekniklerinin en önemli uygulamalarından biri bağlanmış sözcük tanımadır. Burada konuşularak verilen giriş, belirli bir sözlüğe ilişkin bir dizi sözcükten oluşmuştur. Tanıma işlemi ise, sözcük dizisini, yalıtılmış sözcük modelleri yardımı ile, sözcüklerine ayırmaktır. Görüldüğü gibi sözcükler arası yapay beklemler yoktur. Bu nedenle giriş dizisinin sözcüklerine ayrılması sorunu doğmaktadır. Bağlanmış sözcük tanımaya iki tipik örnek verilebilir. Bağlanmış sayı dizilerinin tanınması ve bağlanmış harf dizilerinin tanınması. İlkinde tanıyıcının sözlüğü (0-9) arası sayılardır, ikincisi ise alfabenin A' dan Z'ye harfleri olmaktadır.

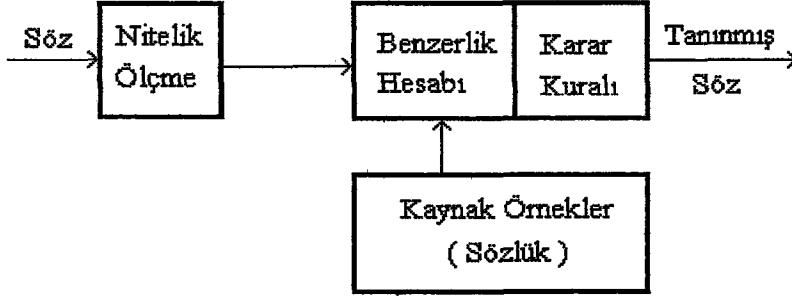
Sürekli söz tanıma sorunu ise yukarıdakilerden daha geneldir. Burada sözcükler arası yapay beklemler olmadığı gibi, aynı zamanda sözcük tanımadaki gibi onun harflerini ayrı ayrı söylemek gerekmez. Sürekli söz tanıma, genellikle fonem, hece gibi sözcüğe oranla daha küçük ses birimlerini temel tanıma birimi olarak seçer. (Bahl,1983 ve Jelinek,1976).

### 3.2. Yalıtılmış Sözcük Tanımadaki Genel Model.

Yalıtılmış sözcük tanımadaki kullanılan bir model blok diyagramlar halinde Şekil 3.1. de verilmiştir. Şekil 3.1. in başlıca üç ana bloktan oluştuğu görülmektedir.

- 1) Nitelik ölçümü
- 2) Benzerlik hesapları
- 3) Karar kuralı

Genel model, değişik sözlük büyüklüğü, nitelik ölçümleri, örnek benzerlik algoritmaları ve değişik karar kuralları altında değişmeden kalır. Dizge deterministik yada istatistiksel yöntemler ile çalışabilir. Bu yalnızca işaret benzerlik ölçümlerini ve karar kuralını etkiler, oysa genellikle nitelik ölçümleri ister deterministik ister istatistiksel yöntemler ile çalışsın, aynı olabilir.



**Şekil 3.1.** Yalıtılmış Sözcük Tanıyıcının Blok Diyagramı.

### 3.3. Nitelik Ölçme.

Nitelik ölçümü, genel olarak veri sayısında indirgeme yapmaktır. Bir akustik söze ilişkin dalga biçimi, uygun bir frekansta örneklenecek genellikle çok sayıda veri elde edilir. Buna karşılık, elde edilen bu veriler işlenerek, sayıları azaltılabilir. Elde edilen azaltılmış veriler nitelik ölçümleri adımı alır ve bunlar belirli anlamda iyi bir biçimde giriş akustik dalga biçiminin yerini tutmalıdır. Gerekli başarı ölçümü, yani bulunan indirgenmiş verinin gerçek işareti temsil etme yeteneği türlü yönlerden tanımlanabilir ve bu veri indirgeme tekniğine bağlıdır. Nitelik kümeleri, basitten karmaşığa, pek çok biçimde oluşturulabilir. Sözelimi, örneklenecek söz işaretinin belirli frekans aralıklarındaki enerjileri, ölçülen nitelikler olabilir (Rabiner,1981) yada giriş işaretinin belirli süreleri kapsayan aralıklardaki örnek değerlerine “doğrusal öngörü kodlaması” uygulanabilir.(Makhoul,1975 ve Rabiner,1981).

Hangi nitelik ölçümünün kullanılacağı karmaşık bir sorundur (Fu K.S,1968). Gerçekleşecek dizgenin maliyeti, cevap süresi, yapılan hesapların zorluğu gibi etkenlere bağlıdır.

Bunlar içinde en önemlileri şöyle sıralanabilir. Hesaplama süresi, dizge için gerekli bellek alanı, dizgenin gerçekleştirilebilirliği. Doğal olarak dizgenin sağladığı

genel verime bakılmalıdır. Bu da yukarıdakilerin pek çoğunu birlikte göz önünde tutmayı gerektirir.

### 3.4. Doğrusal Öngörü Kodlaması (DÖK).

#### 3.4.1. Genel Model.

İlk önce bir sözcüğe ilişkin akustik dalga biçimi  $s(t)$  uygun bir  $F_0$  frekansıyla örneklenecek bir  $S(n)$  ayrık zaman işareti elde edilir.

$$T_0 = 1 / F_0 \quad s(n) = s(n.T_0) \quad n = 0,1,2,\dots \quad (3.1)$$

Bu ayrık zaman işaretinin kısaca  $S_n = S(n)$  ile gösterelim. Kullanılan en güçlü modellerden birinde,  $S(n)$  işareti bir dizgenin  $U_n$  giriş uyarımına ilişkin çıkışı olarak düşünülür. Modeli sayısal bir dizge biçiminde tasarlırsak, zaman domeninde aşağıdaki giriş çıkış ilişkisi yazılabilir.

$$b_0 = 1 \quad S_n = - \sum_{k=1}^p a_k s_{n-k} + K \sum_{l=0}^q b_l u_{n-l} \quad (3.2)$$

Yukarıda  $a_k$ ,  $b_l$  ve  $K$  kazancı, dizgenin yada süzgecin parametreleridir. Buradan görülüyor ki  $s_n$  örnek değeri, çıkışın geçmiş değerleri ile girişin o andaki ve geçmiş değerleri ile öngörülmektedir ve eşitlik doğrusaldır. Doğrusal öngörü adı bu nedenle kullanılır.

(3.2)'deki süzgecin frekans domenindeki transfer fonksiyonu  $z$  dönüşümü alınarak elde edilebilir. Bu fonksiyon (3.1)'de her iki yanın  $z$ -dönüşümü alınarak bulunur.

$$H(z) = \frac{S(z)}{U(z)} = \frac{1 + \sum_{l=1}^q b_l z^{-l}}{1 + \sum_{k=1}^p a_k z^{-k}} K \quad (3.3)$$

Burada,

$$S(z) = \sum_{k=-\infty}^{\infty} s_k z^{-k} \quad (3.4)$$

$H(z)$  için iki özel durum ayırt edilebilir. Bunlardan biri,  $(a_k = 0, k = 1, \dots, q)$  olmak üzere hepsi-sıfır modeli, diğeri  $(b_k = 0, k = 1, \dots, q)$  olmak üzere hepsi-kutup modelidir. İstatistiksel dilde ilki değişken-ortalama ikincisi otoregresif model adını alır.

### 3.4.2. Otoregresif Model.

Yeni duruma göre  $S_n$  ve  $H(z)$  fonksiyonunu yeniden yazalım

$$S(n) = -\sum_{k=1}^p a_k s_{n-k} + K \cdot U_n \quad (3.5)$$

$$H(z) = K / (1 + \sum a_k z^{-k}) \quad (3.6)$$

Şimdi sorun  $a_k$  öngörü katsayılarını ve  $K$  kazancını belirtmektir. Bunun için en küçük kareler yöntemi kullanılabilir.  $U(n)$  bilinmemektedir. Bu nedenle  $S_n$  işaretini yalnızca onun geçmiş değerleriyle öngörmeye çalışalım.

$$\hat{S}_n = -\sum_{k=1}^p a_k s_{n-k} \quad (3.7)$$

Burada  $\hat{S}_n$ ,  $S_n$  'nin öngörölmüş değeridir. Böylece yapılan hata aşağıdaki gibi olur.

$$e_n = S_n - \hat{S}_n = S_n + \sum_{k=1}^p a_k s_{n-k} \quad (3.8)$$

Buradaki  $e_n$  özel olarak "kalan hata" adını alır. Doğrusal öngörü katsayıları hatanın ortalama karesel değerini en küçük yapmak üzere belirlenir. Bu nedenle  $e_n$ , yöntemin başarısı için aynı zamanda bir ölçü olur. Bu arada en küçük yapma işlemi,

$S_n$  işaretinin deterministik yada rastlantı işareti olmasına göre değişiklik gösterir. Öngörülen  $S_n$  'nin bir konuşma işareti olduğu düşünülürse onu bir rastlantı süreci olarak ele almak daha doğru olur. Bu durumda (3.8)' deki hata da bir rastlantı süreci olur ve hatanın karesinin beklenen değeri en küçük yapılmalıdır.

$$E(e_n^2) = E \left( \left( s_n + \sum_{k=1}^p a_k s_{n-k} \right)^2 \right) \quad (3.9)$$

Beklenen değeri en küçük yapmak için aşağıdaki p sayıdaki denklemden oluşan takım çözülmelidir.

$$i = 1, \dots, p \quad \frac{\partial E}{\partial a_i} = 0 \quad (3.10)$$

Böylece “normal eşitlikler” adını alan aşağıdaki denklem takımı elde edilmiş olmaktadır.

$$\sum_{k=1}^p a_k E(s_{n-k} s_{n-i}) = -E(s_n s_{n-i}) \quad (3.11)$$

(3.11) eşitliğini (3.9) ifadesinde kullanırsak, hatayı en küçük yapmak üzere seçilen  $a_k$  'lar nedeniyle, en küçük beklenen hata yada ortalama hata bulunabilir.

$$E_p = E(s_n^2) + \sum_{k=1}^p a_k E(s_n s_{n-k}) \quad (3.12)$$

Beklenen değerlerin hesaplanması  $s_n$  'in istatistiksel özelliklerine bağlıdır. Bu durumda  $s_n$  sürecini stasyonere varsayarak (Papoulis, 1965) beklenen değeri hesap edebiliriz.

$$E(s_{n-k} s_{n-i}) = R(i - k) \quad (3.13)$$

Burada  $R(i)$ ,  $s_n$  sürecinin öz ilişki fonksiyonudur ve ayrıca  $s_n$  gerçek olduğundan bu fonksiyon çifttir.

$$R(i) = R(-i) = \sum_{k=-\infty}^{\infty} s_n s_{n+i} \quad (3.14)$$

Uygulamada  $s_n$  işareti zamanda belirli ve sonlu bir aralıkta gözlenir. Bu nedenle bir zaman aralığını analiz ediyor olmanın etkisini azaltmak için pencere fonksiyonu kullanılmalıdır.

$$\bar{s}_n = s_n p_n \quad 0 \leq n \leq N-1 \quad ; 0 \text{ değilse} \quad (3.15)$$

Böylece öz ilişki fonksiyonu aşağıdaki biçime indirgenir.

$$R(i) = \sum_{n=0}^{N-1-i} \bar{s}_n \bar{s}_{n+i} \quad i \geq 0 \quad (3.16)$$

Sonuç olarak (3.13) ve (3.16) yardımıyla (3.11) eşitlikleri kurulabilir. Bu denklem takımının çözülmesi için pek çok yöntem vardır. Bunlardan biri Durbin'in rekürsif yöntemi aşağıda algoritma biçiminde verilmiştir.

$$E_0 = R(0) \quad (3.17,a)$$

$$k_i = -(R(i) + \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j)) / E_{i-1} \quad (3.17,b)$$

$$a_i^{(i)} = k_i$$

$$a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \quad 1 \leq j \leq i-1 \quad (3.17,c)$$

$$E_i = (1 - k_i^2) E_{i-1} \quad (3.17,d)$$

(3.17,b-d) eşitlikleri rekürsif biçimde  $i=1, \dots, p$  için çözülür ve en son olarak elde edilen  $a_j$  değerleri doğrusal öngörü katsayıları olur.

$$a_j = a_j^{(p)} \quad (3.17,e)$$

Ayrıca, rekürsiyonun her bir  $i$ 'inci adımı için şu özellik vardır.

$$0 \leq E_i \leq E_{i-1} \quad (3.17,f)$$

Bundan sonra (3.6) daki  $K$  kazancı bulunmalıdır. Bunun için (3.8)'i yeniden yazalım.

$$s_n = -\sum_{k=1}^p a_k s_{n-k} + e_n \quad (3.18)$$

Bu ifadeyi (3.5) ile karşılaştırırsak  $s_n$  işaretinin süzgeç çıkışından görülmesi için aşağıdaki ilişkinin geçerli olması gerektiği görülür.

$$e_n = K \cdot u_n \quad (3.19)$$

Yukarıdaki eşitliği sağlamayan başka bir  $u_n$  girişi, değişik bir  $s_n$  verecektir. Bununla birlikte  $u_n$  ne olursa olsun süzgeç çıkışındaki işaretin enerjisi, gerçek  $s_n$  işaretinin enerjisine eşit olsun dersek en azından giriş işaretinin enerjisini

belirleyebiliriz. Böylece  $K.u_n$  çarpımının toplam enerjisi, (3.17,e)'de hesaplanan  $\alpha_k$ 'lar ile, hatanın en küçük enerjisi olan  $E_p$  olur.  $u_n$  giriş işaretini beyaz gürültü olarak seçersek, yani

$$E(u_n) = 0 \quad E(u_n \cdot u_m) = 1 \quad n = m, 0 \quad n \neq m \quad (3.20)$$

ise girişteki toplam enerji  $K^2$  olur ve bu da hatanın enerjisine eşit olmalıdır. (3.12)'den  $e_n$ 'i en küçük yapan  $\alpha_k$ 'ları kullanırsak,  $E_p$  ve dolayısı ile kazanç hesaplanır.

$$K^2 = E_p = R(0) + \sum_{k=1}^p \alpha_k R(k) \quad (3.21)$$

Yukarıdaki biçimde gerçekleştirilen doğrusal öngörü analizindeki bazı özellikler aşağıda belirtilmiştir.

### 1) Süzgeç Kararlılığı.

(3.17, a-f) formülleri ile  $\alpha_k$  katsayıları bulunurken, aralarında (3.17, f) ilişkisi bulunan  $E_i$ ,  $i = 0, \dots, p$  enerjileri hesaplanmaktadır. Her  $E_i$  enerjisinin sıfırdan farklı ve pozitif olma koşulu,  $H(z)$  süzgecinin kararlılığı için gerekli ve yeterli koşuldur.

$$i = 1, \dots, p \quad E_i > 0 \quad (3.22)$$

Aynı zamanda bu koşul şöyle de yazılabilir.

$$i = 1, \dots, p \quad |k_i| < 1 \quad (3.23)$$

Böylece (3.17)'deki yöntem süzgeç kararlılığının kontrolünü de yapmaktadır.

### 2) Normalize Hata.

En küçük hatanın işarettaki enerjiye olan oranı olarak tanımlansın.

$$V_p = E_p / R(0) \quad (3.24)$$

a) Normalize hata  $p$  değeri arttıkça azalır.

b)  $V_p$ , işaret spektrumunun biçimine de bağlıdır. Spektrumda en yüksek ve en alçak genlik değerini sırasıyla  $Y$  ve  $A$  ile gösterirsek bunlar arasında bir oran tanımlanabilir.

$$d = Y / A \quad (3.25)$$

Eğer  $P(\omega)$ ,  $s_n$  işaretinin spektrumu ise

$$Y = \max P(\omega) \quad A = \min P(\omega) \quad (3.26)$$

olur. Gösterilebilir ki (Makhoul,1975), normalize hata  $V_p$  spektrumun  $d$  oranı arttıkça azalır.

### 3) Kritik Koşulluluk.

(3.11)'deki normal eşitlikler (3.13) ve (3.16) yardımıyla matris biçiminde olmak üzere yeniden yazılsın

$$\begin{bmatrix} R_0 & R_1 & R_2 & \dots & R_{p-1} \\ R_1 & R_0 & R_1 & \dots & R_{p-2} \\ R_2 & R_1 & R_0 & \dots & R_{p-3} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ R_{p-1} & R_{p-2} & \cdot & \dots & R_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = - \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \cdot \\ \cdot \\ R_p \end{bmatrix} \quad (3.27)$$

Burada  $R_k = R(k)$  olarak kısaltılmıştır. (3.27)'deki öz ilişki normal eşitliklerinin çözümünde, öz ilişki matrisinin değerleri önemlidir. Çünkü eğer (3.27) kritik koşullara uyan eşitlikler ise (Stanton,1981) öz ilişki değerlerindeki çok küçük değişimler bile  $(a_k, k=1, \dots, p)$  çözümünü çok değişik yerlere getirebilir. Bu durumda (3.27)'nin kritik koşullara uyan bir eşitlik olup olmadığı bilinmelidir. Kritik koşulluluğa uygun bir ölçü şudur.

$$d' = \lambda_{\max} / \lambda_{\min} \quad (3.28)$$

Burada  $\lambda_{\max}$  ve  $\lambda_{\min}$ , sırasıyla öz ilişki matrisinin en büyük öz değerlerinin  $\lambda \in [A, Y]$ ,  $i=1, \dots, p$  arasında olduğu gösterilmiştir. Burada  $A$  ve  $Y$  (3.26)'dan bulunur. Bu durumda  $d'$  oranı yaklaşık olarak  $d$  oranına eşit olur.

$$d' \cong d \quad (3.29)$$

$d'$  değeri ne kadar büyükse, matrisin kritik koşullu olması şansı o kadar çok olur.

Böylece  $d$  oranı kritik koşulluluk içinde iyi bir ölçüdür. Ayrıca en küçük normalize hata  $V_p, d$  ile orantılı olduğuna göre,  $V_p$ 'nin kendisi de ölçü olarak alınabilir.  $V_p$  azaldıkça kritik koşulluluk artar. Bu ikinci özellik ile bir karşılık oluşturmaktadır. Buna karşın  $s_n$  işaretinin spektrumundaki  $Y/A$  oranı

düştürülmelidir, yani  $s_n$  işareti önce bir düzleştirici süzgeçten geçirilmelidir. Böyle bir süzgecin giriş-çıkış ilişkisi aşağıdaki gibidir.

$$s_n = s_n + a \cdot s_{n-1} \quad (3.30)$$

Buradaki  $a$  katsayısı şöyle bulunabilir.

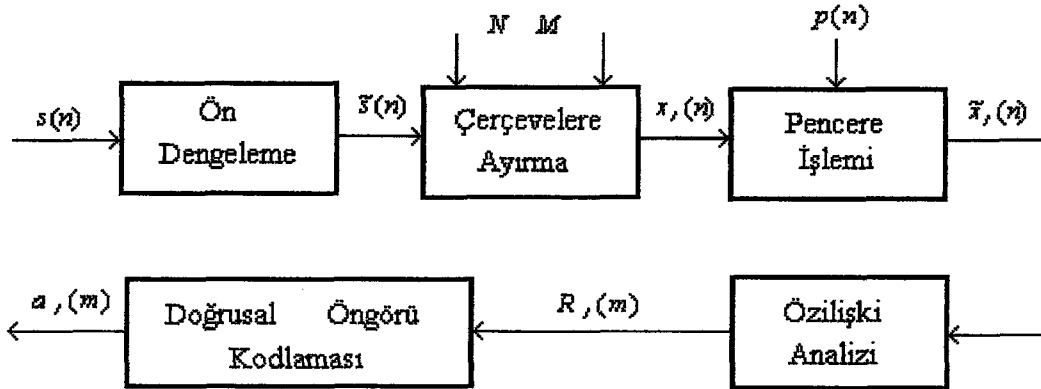
$$a = -R(1) / R(0) \quad (3.31)$$

$R(1)$  ve  $R(0)$ ,  $s_n$ ' e ilişkin öz ilişki fonksiyonundan elde edilebilir. (3.30)'daki işlem ön dengeleme adını alır.

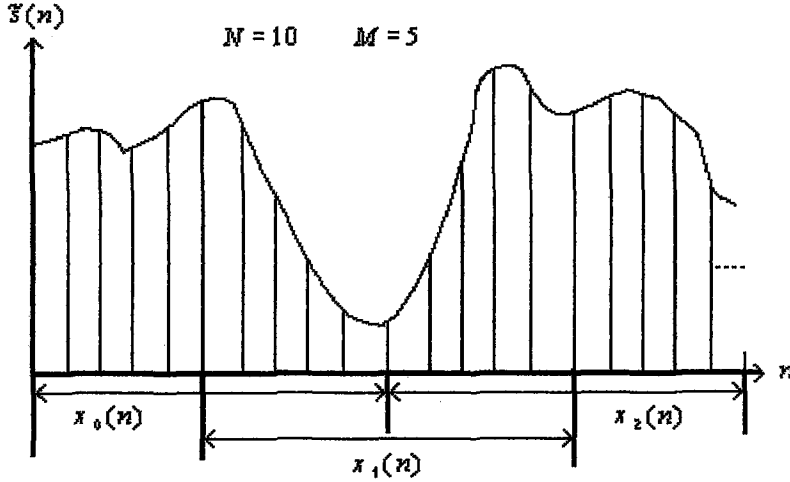
### 3.5. Bütün Bir Nitelik Ölçücü.

#### 3.5.1. Blok Diyagramı.

Yukarıdaki bilgiler altında, doğrusal öngörü kodlamasını kullanan bir nitelik ölçücü Şekil 3.2'de gösterilmiştir.



Şekil 3. 2. D.Ö.K. Analizi Kullanan Nitelik Ölçücü.



Şekil 3.3. İşaretin Zaman Çerçevesine Ayrılması.

$s(n)$  işareti ön dengelemeden geçtikten sonra çerçeve adı verilen  $N$  sayıda örneklerin toplandığı, zaman parçalarına ayrılmaktadır.

$$\begin{aligned} x_l(n) &= \tilde{s}(M.l + n) \\ n &= 0, 1, \dots, N-1 \\ l &= 0, 1, \dots, L-1 \end{aligned} \quad (3.32)$$

Görülüyor ki, mesela  $M = N/2$  yada  $N/3$  ise zaman çerçevesi arasında birbirini örtmeler olur.  $M = N$  ise hiçbir örtme yoktur. Bu nedenle  $M$  çerçevelerin başlangıçları arasındaki uzaklık olarak düşünülebilir. Bu durum Şekil 3.3'de gösterilmiştir. Bundan sonra her bir zaman çerçevesinde ilişkin  $x_l(n)$   $l = 0, \dots, L-1$  işareti pencere fonksiyonu ile çarpılacaktır.

$$\bar{x}_l(n) = x_l(n) \cdot p(n) \quad (3.33)$$

Doğrusal öngörü kodlaması analizinde kullanılan tipik bir fonksiyon Hamming penceresidir.

$$p(n) = 0.54 - 0.46 \cdot \cos(2 \cdot \pi \cdot n / (N-1)) \quad (3.34)$$

Sonraki adım öz ilişki analizidir.

$$m = 0, \dots, p \quad R_l(m) = \sum_{n=0}^{N-1-m} \bar{x}_l(n) \bar{x}_l(n+m) \quad (3.35)$$

Elde edilen öz ilişki değerleri kullanılarak, her bir  $l$ 'inci zaman çerçevesine ilişkin doğrusal öngörü katsayıları  $a_l(m)$ 'ler elde edilir. Bunun için

(3.17,a-e)'deki Durbin'in rekürsif yöntemi kullanılabilir. Elde edilen  $a_l(m)$  değerleri bir vektörde toplandığında ;

$$l = 0, \dots, L - 1 \quad \text{için} \quad a_l = (a_l(1), \dots, a_l(p)) \quad (3.36)$$

elde edilir. Yukarıdaki vektör, söz tanıma literatüründe "akustik vektör" adını alır. Genel olarak, bir nitelik ölçücünün girişi bir sözcüğe ilişkin ayrık zaman işareti, çıkışı ise bir dizi akustik vektör olmaktadır. Bu andan itibaren zaman eksenindeki bir sözcüğe ilişkin akustik dalga biçimi yerine, belirli sayıda akustik vektörden oluşan bir dizi kullanılacaktır.

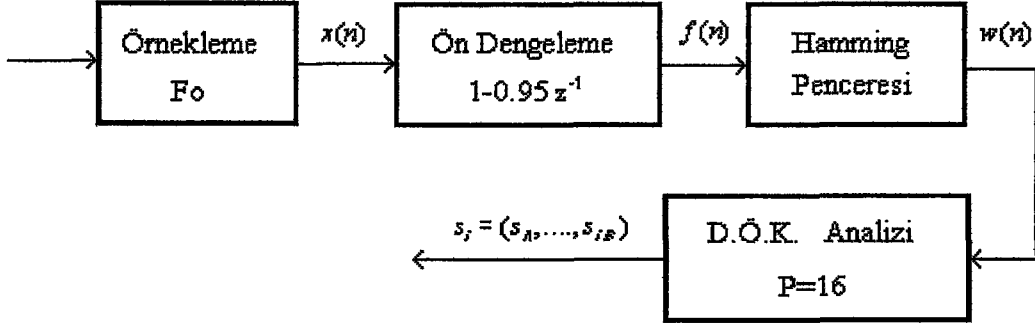
### 3.5.2. Sayısal Örnek.

Bir 200-örneklik  $x(n)$  Şekil 3.5. de gösterilmiştir. Bu işaret / i / sesli harfine aittir ve Şekil 3.3.'deki genel yapıda gösterilen  $x_l(n)$  zaman çerçevelerinden birine denk gelmektedir. Örnek genişliği  $N=200$  alınmıştır. Şekil 3.2.'deki genel yapıya uyması açısından, Şekil 3.4.'de verilecek örneğe ilişkin diyagram gösterilmiştir.

Burada aşağıdaki bağıntılar yazılabilir.

$f(n) = x(n) - a \cdot x(n-1)$	: süzölmüş işaret
$\omega(n) = f(n)p(n)$	: pencerelenmiş işaret
$a = 0.95$	: ön dengeleme sabiti
$p(n) = (0.54 - 0.46 \cdot \cos(2.n\pi / N - 1))$	: Hamming penceresi
$N = 200$	: Örnek sayısı
$F_0 = 10\text{Khz}$	: Örnekleme frekansı

Şekil 3.5 ve 3.6'da sırasıyla  $x(n), f(n)$  gösterilmiştir. Bu arada  $x(n)$  ve  $f(n)$ 'e ilişkin frekans spektrumları Şekil 3.7 ve 3.8'de verilmiştir. Söz konusu iki spektrumda en büyük ve en küçük genlik değerleri karşılaştırılırsa, (3.25)'deki  $Y/A$  oranının oldukça düştüğü görülebilir. Frekans spektrumları 1024 noktalı hızlı Fourier dönüşümü olarak elde edilmişlerdir.



Şekil 3.4. Sayısal Örneğe İlişkin Blok Diyagramı

Doğrusal öngörü analizinde, daha önce kalan hata adını verdiğimiz (3.8) formülü ile hesaplanan hata, Şekil 3.9'da çizilmiştir. (3.8) formülünde  $s_n$  yerine süzgeç çıkışındaki  $f(n)$  kullanılmıştır. Hata ile  $f(n)$  işareti arasındaki fark görülmektedir. Diğer yandan Şekil 3.10'da hatanın frekans spektrumu verilmiştir. Görüldüğü gibi spektrumun zarfı düz bir biçime sahiptir.(Beyaz gürültü)

Hata analizinde kullanılan diğer bir ölçü,  $s_n$  işaretine ilişkin frekans spektrumunun kestirilmesi ile ilgilidir. Gösterilebilir ki, kestirilmiş spektrumun gerçeğine ne kadar yakınsa, doğrusal öngörü analizinde yapılan hata o kadar düşüktür.  $P(\omega)$ ;  $s_n$ 'nin frekans spektrumu olsun.

$$P(\omega) = |S(e^{j\omega})|^2 \quad (3.37)$$

Öte yandan (3.8) ifadesinin iki yanında z-dönüşümü alırsak

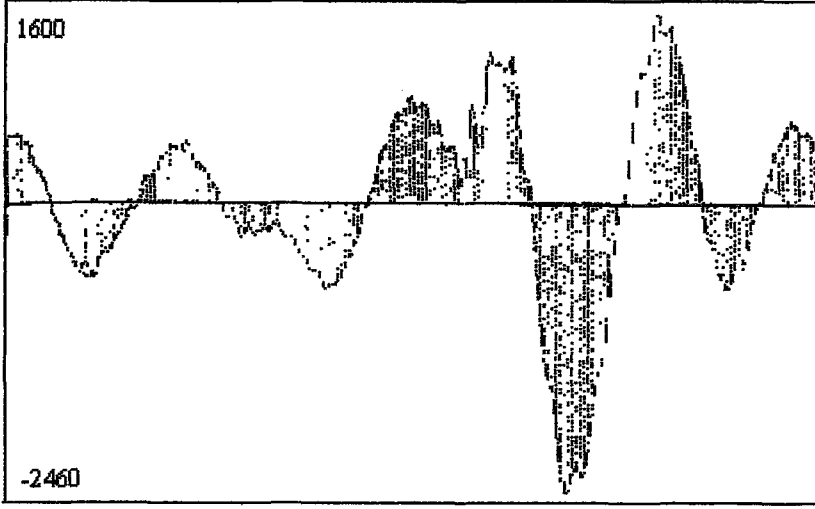
$$E(z) = S(z) \cdot \left(1 + \sum_{k=1}^p a_k z^{-k}\right) = S(z)A(z) \quad (3.38)$$

bulunur. Burada  $E(z)$  ve  $S(z)$  sırasıyla hatanın ve işaretin z-dönüşümleridir.  $A(z)$  ise (3.6)'daki  $H(z)$ 'nin ters süzgeç transfer fonksiyonudur. Böylece şu yazılabilir,

$$P(\omega) = |S(e^{j\omega})|^2 = |E(e^{j\omega})|^2 / |A(e^{j\omega})|^2 \quad (3.39)$$

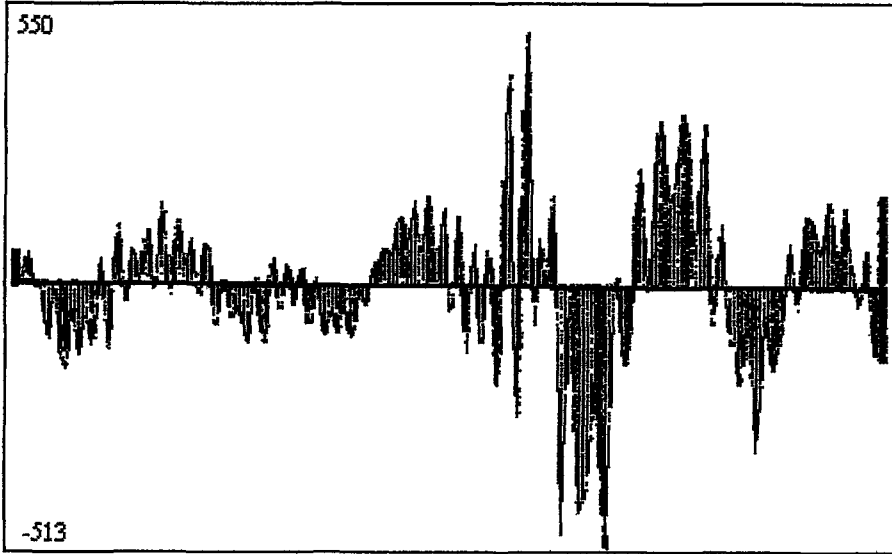
Daha önce, (3.20)'de girişi beyaz gürültü olarak almıştık ve hata (3.19) ifadesi ile verilmişti. Bu durumda  $P(\omega)$  için aşağıdaki kestirimi yapabiliriz.

$$\hat{P}(\omega) = K^2 / |A(e^{j\omega})|^2 \quad (3.40)$$

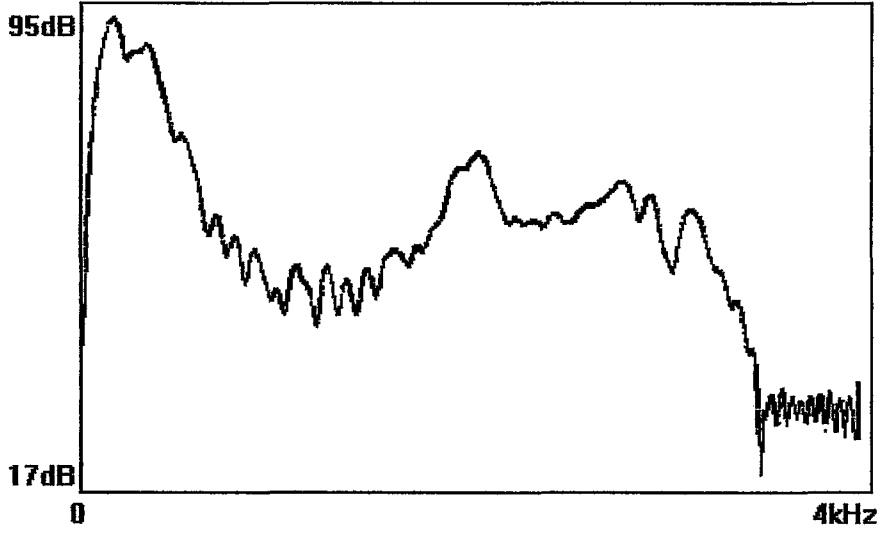


Şekil 3.5.  $x(n)$  İşareti.

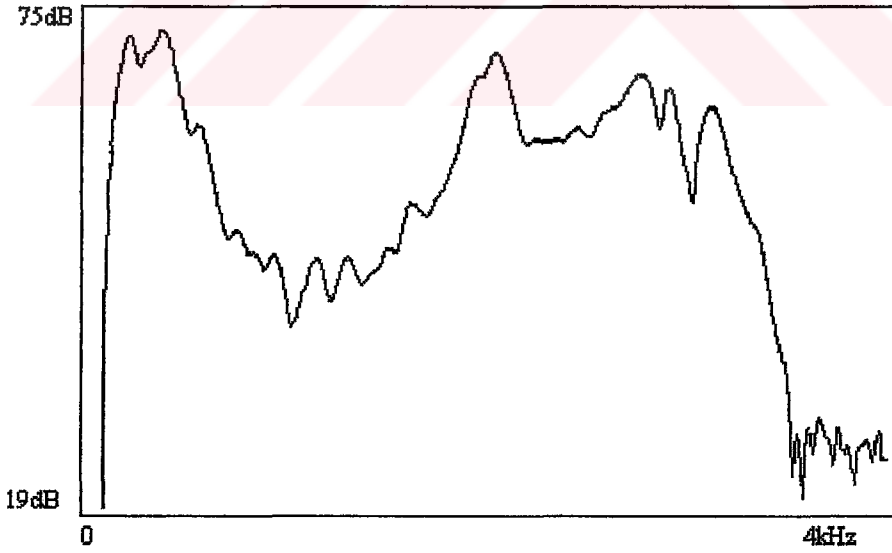
Sayısal örnekte hesaplanan  $(\alpha_k, k=1, \dots, p)$  ve  $K$  değerleriyle  $\hat{P}(\omega)$  spektrumu elde edilmiştir. Bu spektrumu  $f(n)$ 'nin gerçek spektrumu ile karşılaştırmak için Şekil 3.11'de iki çizim üst üste gösterilmiştir. Burada daha düz olan kestirilmiş spektrumdur. Çizimlerin birbirine uygunluğu açıkça görülmektedir.



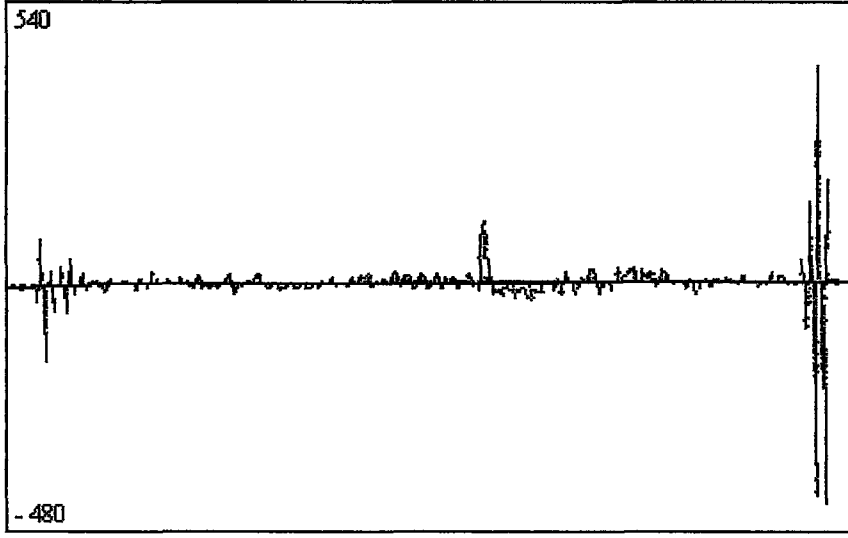
Şekil 3.6.  $f(n)$  İşareti.



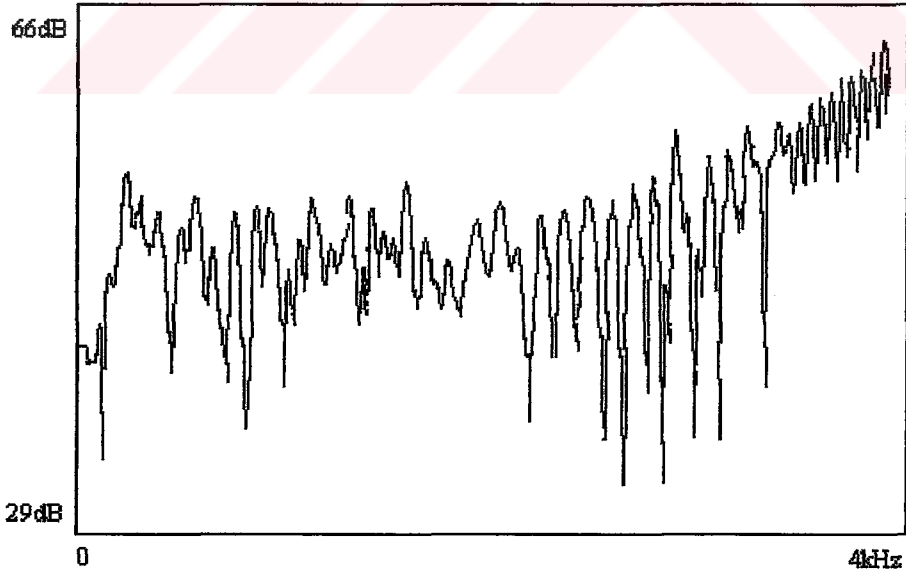
Şekil 3.7.  $x(n)$  İşaretinin Frekans Spektrumu.



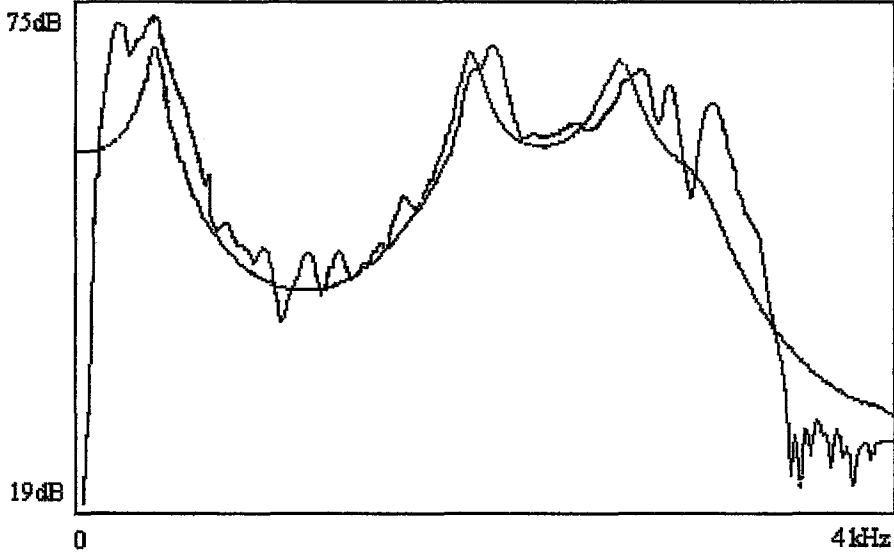
Şekil 3.8.  $f(n)$  İşaretinin Frekans Spektrumu.



Şekil 3.9. Kalan Hataya Ait İşaret.



Şekil 3.10. Kalan Hatanın Frekans Spektrumu.



Şekil 3.11.  $f(n)$ 'nin Gerçek ve Kestirilmiş Spektrumu.

### 3.6. Başlangıç ve Bitiş Noktalarını Belirleme.

Konusulan bir sözcüğün başlangıç ve bitiş noktalarını belirleme işlemi son nokta belirleme problemi olarak bilinir. İyi bir tanıma için, başlangıç ve bitiş noktalarına ait bu sınırların doğru bir şekilde belirlenmesi, bütün sözcük tanıma sistemleri için gereklidir.

Sinyalin gürültüye oranı ( $SNR = e_s / e_n$ ) büyük olan işaretlerde, en küçük seviyedeki konuşma işaretinin enerjisi bile, ortamın gürültüsünün enerji seviyesini aştığından, başlangıç ve bitiş noktalarını belirleme işlemi kolaylaşır. Ancak doğal ortamlarda genellikle istenilmeyen gürültü kaynakları bulunduğu için, sözcüğe ait işaretleri gürültüden ayırmak özel işlemler gerektirir. Bu çalışmada, enerji ( $E$ ) ve sıfır kesme oranına ( $S.K.O.$ ) dayalı başlangıç ve bitiş noktalarını belirleyen algoritma kullanılmıştır. Bahsedilen her iki özellik de zaman domeni özelliğidir (Rabiner ve Sambur, 1975 ; Lamel vd., 1981).

Öncelikle yaklaşık olarak, başlangıç ve bitiş noktalarını belirlemek için enerji ve daha sonrada sıfır kesme oranı kullanılarak daha kesin sonuçlar elde edilir.

Enerji,

$$E = \sum_{n=-\infty}^{\infty} x^2(n) \quad (3.41)$$

denklemini şeklinde genel olarak tarif edilir. Ancak konuşma işaretlerinde olduğu gibi sabit olmayan işaretleri aşağıdaki şekilde ele almak daha uygun olacaktır.

$$E(n) = \sum_{j=0}^{N-1} [\omega(j)x(n-j)]^2 \quad (3.42)$$

Burada  $\omega(j)$ ,  $x(n)$ 'nin belirli bir kısmını pencereleyen bir ağırlık fonksiyonudur.  $N$  ise penceredeki örnek sayısıdır. Seçilen aralıktaki örneklere eşit miktarda ağırlık uygulamak için dikdörtgen pencere kullanılır. Bu durumda  $\omega(j) = 1$  olur. Buna göre  $E(n)$ ,  $x(n)$ 'nin en son  $N$  tane değerinin karelerinin toplamı olur.  $E(n)$ , sözcüğe ait işaretin zamanla değişen genlik özelliklerini gösterir. Uygun bir enerji seviyesi için  $N$  dikkatle seçilmelidir. (3.42) ifadesinde kare alma işlemi sebebi ile, büyük sinyal seviyelerine hassas tepki verir, sonuçta büyük değerlerin ortaya çıkmasına sebep olur. Bu istenmeyen durumu önlemek için

$$E(n) = \sum_{j=0}^{N-1} |\omega(j)x(n-j)| \quad (3.43)$$

eşitliğini kullanmak gerekir. Burada kare alma yerine mutlak değer alma işlemi yapılmıştır. Sıfır kesme oranı (S.K.O), işaretin bölümlere ayrılmış kısımlarının, boşluğa mı yoksa sözcüğe mi ait olduğunu belirlemek için kullanılır ve aşağıdaki gibi tanımlanır (Ito ve Donaldson, 1971).

$$S.K.O.(n) = \sum_{j=0}^{N-1} [Sgn[x(n-j+1)] - Sgn[x(n-j)]] \omega(j)$$

Burada

$$Sgn[x(n)] = 1 \quad x(n) \geq 0$$

$$Sgn[x(n)] = -1 \quad x(n) \leq 0$$

ve

$$\omega(j) = 0.5 \quad 0 \leq j \leq N-1 \quad \omega(j) = 0 \quad \text{diğer durumlarda.}$$

Başlangıç ve bitiş noktalarını belirleyen BBN isimli alt programın algoritması aşağıda verilmiştir (Rabiner ve Schafer,1978).

1- Örneklenmiş sözcük, SOZCUK isimli diziye atılmış durumdadır. Bu durumda, alınmış 15000 adet örnek 100'erli guruplara ayrılır. (100'er örnek değeri bulunan 150 adet bölüm).

2- Her bölümün enerjisi ve sıfır kesme oranı hesaplanarak, maksimum enerji ve maksimum sıfır kesme oranı belirlenir.

3- Bu kısım için ortalama S.K.O'nı, S.K.O.'nın standart sapması ve ortalama enerjisi, son yirmi bölümün, sözcüğe değil de boşluğa karşılık geldiği düşünülerek hesaplanır.

4- Sıfır kesme oranı eşik değeri  $SKOED$ , aşağıdaki eşitlikten belirlenir.

$$SKOED = \min(15, BOSKO + 3 * SKOSS)$$

Burada  $BOSKO$ , boşluğun ortalama sıfır kesme oranı,  $SKOSS$  ise sıfır kesme oranının standart sapmasıdır.

5- Alt enerji eşik değeri  $AEED$ , aşağıdaki gibi belirlenir.

$$AEED = \min[0.03 * (BME - BOE) + BOE, 4 * BOE]$$

Burada  $BME$  boşluğun maksimum enerjisi,  $BOE$  ise boşluğun ortalama enerjisidir.

6- Üst enerji eşik değeri  $UEED$ , aşağıdaki gibi hesaplanır.

$$UEED = 5 * AEED$$

7- Yaklaşık başlangıç ve bitiş bölümleri olan  $YBA$  ve  $YBI$  değerleri aşağıdaki gibi belirlenir.

$YBA$ 'nın belirlenmesi: Baştan itibaren her bölümün enerji değeri kontrol edilir. Enerjisi  $AAED$  değerinden büyük olan ilk bölüm belirlenir. Eğer bu bölümden sonraki bölümlerde enerji değeri  $AAED$ 'den aşağı düşmeden önce  $UEED$  değerini de aşıyor ise,  $AAED$  değerini ilk aşan bölüm başlangıç bölümüdür.  $AAED$  değeri aşıldıktan sonraki bölümlerde,  $UEED$  değeri aşılmadan enerji tekrar  $AAED$ 'nin altına düşüyor ise bu bölümler atlanır. Bundan sonraki ilk  $AAED$ 'yi aşan bölüm için aynı işlemler tekrarlanır.

$YBI$ 'nin belirlenmesi: Yukarıda bahsedilen işlemler, sözcüğün sonundan başına doğru gidilerek tekrarlanır ve böylece  $YBI$  belirlenmiş olur.

8- Yaklaşık olarak belirlenen başlangıç (*YBA*) ve bitiş (*YBI*) bölümlerini daha doğru bir şekilde bulmak için sıfır kesme oranı bilgileri aşağıdaki şekilde kullanılır:

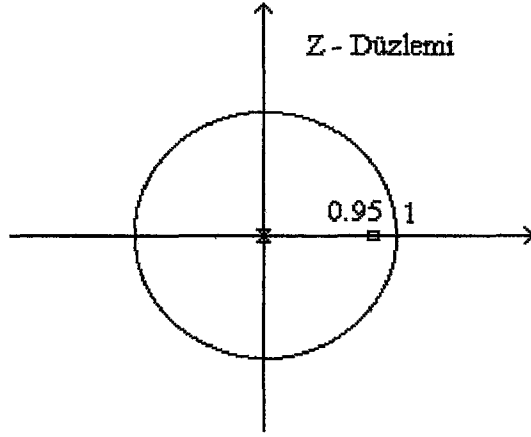
(*YBA*) değerinden başlayarak geriye doğru (sözcüğün başlangıcına doğru) ardışık on bölüm için araştırma yapılır. Eğer arka arkaya en az üç bölümün *S.K.O.*, *SKOED* değerinden yukarıda kalıyor ise, başlangıç bölümü olarak bu ardışık değerlerin ilki alınır. bu bölüm *BA* olsun. Eğer arka arkaya üçten az bölüm *SKOED* değerini aşıyor ise, bu bölümler atlanır ve geriye doğru bu işlemler tekrarlanır. Yukarıdaki işlemler, önceden belirlenen (*YBA*) bölümünden başlayarak, sözcüğün başlangıcına doğru (geriye doğru) sadece on bölüm için yapılır.

*BA*'nın bulunması için anlatılmış olan yöntem aynı şekilde *BI*'nin bulunması içinde kullanılır. Ancak yaklaşık bitiş değeri olarak belirlenen *YBI* bölümünden başlayarak, ileriye doğru (sözcüğün sonuna doğru) ardışık on bölüm için yukarıdaki işlemler yapılır. Sonuç olarak belirlenen *BA* başlangıç bölümü (aralığı), *BI* ise bitiş bölümüdür. Bu bölümlerin ilk örnek değerleri sırasıyla başlangıç ve bitiş noktalarıdır.

### 3.7. Ön Dengeleme.

Ön dengeleme, spektral eğimi 6dB/octav'dan aşağıda kalan ses işaretlerini kompanze etmek için kullanılan bir yöntemdir (Rabiner ve Schafer,1978;Makhoul ve Wolf,1972). Ön dengeleme yardımı ile frekans spektrumunda görülen düşük frekanslı bileşenlerden kurtulmak mümkündür. Ön dengeleme, bahsedilen düşük frekanslı bileşenlerden kurtulmanın yanı sıra, yüksek frekanslı bileşenleri güçlendirerek, tanıma oranının artmasını sağlar (Rabiner vd.,1977; Tribolet vd.,1979). Ön dengeleme işlemi, formant pozisyonlarında kaydırma yaparak distorsiyona sebep olur. Buna rağmen söz tanıma sistemlerinde tanıma oranını arttırdığı için, bu olumsuz duruma hoşgörü ile bakılabilir. Ön dengeleme zaman domeninde aşağıdaki şekilde ifade edilebilir.

$$S'(n) = S(n) - \alpha S(n-1) \quad (3.44)$$



Şekil 3.12.  $H(z)$  'nin Sıfırı ve Kutbu.

Burada  $S(n)$  ve  $S(n-1)$  ardışık konuşma örnekleridir. Sözcük analizi için  $a$  katsayısını 0.9 ile 1 arasında seçmek uygundur (Rabiner vd.,1979). Bu çalışmada  $a = 0.95$  alınmıştır. (3.44) ifadesinin z-dönüşümü alınır

$$S'(z) = (1 - 0.95z^{-1})S(z) = H(z)S(z) \quad (3.45)$$

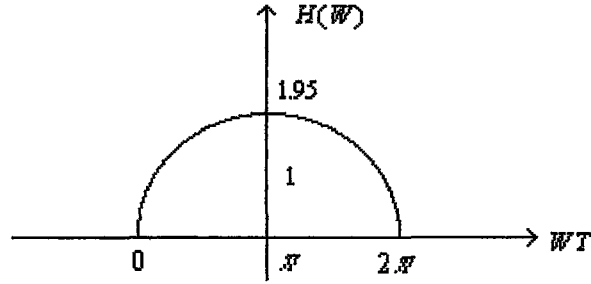
olur ve burada  $H(z) = 1 - 0.95z^{-1}$ , zaman domenindeki fark işleminin frekans domenindeki karşılığı olan birinci dereceden sayısal filtrenin transfer fonksiyonudur. Bu filtre, sisteme  $z = 0.95$ 'de bir sıfır ilave eder. Şekil 3.12'de  $H(z)$ 'nin sıfırı ve kutbu'nun yerleri gösterilmiştir. Birim çember üzerinde sıfırın yerine bakıldığında, düşük frekanslar bölgesine daha yakın olduğu görülecektir. Bu nedenle işaretin düşük frekanslı bileşenlerini zayıflatır. Orijindeki kutup, çember üzerindeki bütün noktalara eşit uzaklıkta olduğundan, transfer fonksiyonu  $H(z)$ 'nin büyüklüğünü etkilemez.  $S'(n)$ 'nin güç spektrumu,

$$P'(\omega) = |S'(\omega)|^2 = |H(\omega)|^2 (P\omega) \quad (3.46)$$

olur. Burada  $P(\omega)$  orijinal sözcüğe ait işaretin güç spektrumudur.  $z = e^{j\omega T}$  eşitliği,  $H(z)$ 'de kullanılarak güç spektrumu yeniden yazılırsa,

$$P'(\omega) = |1 - 0.95e^{-j\omega T}|^2 P(\omega) \quad (3.47)$$

eşitliği elde edilir. Burada  $T$  örnekleme periyodudur. Euler formülünü kullanarak



**Şekil 3.13.** Sayısal Filtrenin Frekans Cevabı.

$$P'(\omega) = \left| 0.0005 + 3.8 \sin^2(\omega T/2) \right| P(\omega) \quad (3.48)$$

yazılabilir. Filtrenin frekans cevabı yaklaşık olarak aşağıdaki gibi olur.

$$|H(\omega)| = \left| 0.0005 + 3.8 \sin^2(\omega T/2) \right|^{1/2} = 1.95 \sin(\omega T/2) \quad (3.49)$$

Böylece ön dengelemenin etkisi, orijinal işaretin güç spektrumunu,  $z = 0.95$  sıfırının spektral cevabı  $0.0005 + 3.8 \sin^2(\omega T/2)$  ile çarpma olarak ortaya çıkar. Şekil 3.13'de  $z = 0.95$ 'de sıfırı olan sayısal filtrenin frekans cevabı görülmektedir.

### 3.8. Pencereleme.

Uygun bir zamandan bağımsız filtre modeli elde etmek için, işarete ait bazı özelliklerinin, değişiminin önemli miktarda olmadığı uygun bir aralıkta, öz ilişki katsayılarının hesaplanması gerekir. Sözcüklere ait işaretleri analiz ederken, katsayı hesabı için sadece  $N$  örneği kullanmak demek, sözcüğe ait işaret örneklerini, yüksekliği 1 ve genişliği  $N$  olan bir dikdörtgen pencere ile çarpma demektir. Bir pencere, sonlu bir dizidir ve ana dizide karşılık gelen her elemanın pencere elemanları ile çarpılması işlemine pencereleme denir (Harris, 1978).

Dikdörtgen pencerelemede, bir aralıktaki, ilk  $P$  tane sözcük örneği, pencerelenen aralığın dışındaki sıfır örnek değerleri kullanılarak öngörülürler. Bu ise ilk  $P$  tane örneğin kestiriminde (öngörülmesinde) büyük oranda hataların oluşmasına sebep olur. Ortaya çıkan bu büyük orandaki hata miktarı, dikdörtgen olmayan, fakat giderek işareti azaltan ve uçlarda sıfır yapan bir pencere kullanılarak azaltılabilir. Zaman bölgesinde bir işareti pencere fonksiyonu ile çarpmak, frekans bölgesinde, işaretin frekans spektrumu ve pencere fonksiyonunun, frekans spektrumunun katlanması (konvolüsyon) demektir.

$$F_w(w) = F(w) * W(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\alpha) W(w - \alpha) d\alpha \quad (3.50)$$

Burada  $F(w)$  konuşma işaretinin Fourier dönüşümü,  $W(w)$ 'de pencere fonksiyonunun Fourier dönüşümüdür. Pencereleme işleminin, sözcüğe ait işaret üzerindeki etkisini görmek için iki farklı pencereleme fonksiyonu (Dikdörtgen ve Hamming Pencere) aşağıda incelenecektir.

Dikdörtgen pencerede,  $(n - N + 1)$  ile  $n$  arasındaki bütün örneklere eşit ağırlık uygulansın.

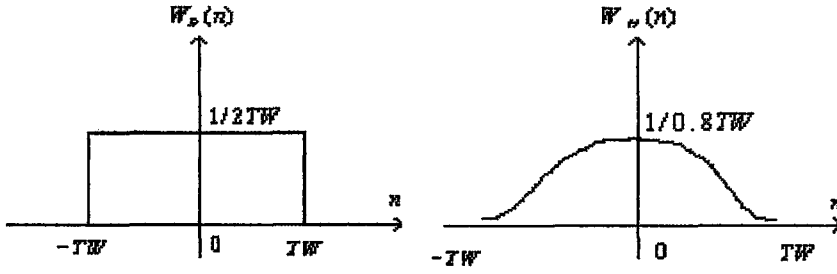
$$\begin{aligned} W_R(n) &= 1 & 0 \leq n \leq N - 1 \\ W_R(n) &= 0 & \text{diğer durumlarda} \end{aligned}$$

Hamming pencere ise, pencerenin sınırlarına gidildikçe, işareti sıfıra doğru zayıflatan ve uçlarda sıfır yapan bir pencere fonksiyonudur. aşağıdaki gibi ifade edilebilir.

$$\begin{aligned} W_H(n) &= 0.54 - 0.46 \cos(2\pi n / (N - 1)) & 0 \leq n \leq N - 1 \\ W_H(n) &= 0 & \text{diğer durumlarda.} \end{aligned}$$

Şekil 3.14'de dikdörtgen pencere ve hamming pencerenin zaman domenindeki normalize edilmiş temsilleri gösterilmiştir. Burada  $2Tw$ , pencere genişliğidir. dikdörtgen ve hamming pencereye karşılık gelen güç spektrumları Şekil 3.15 ve Şekil 3.16'da verilmiştir.

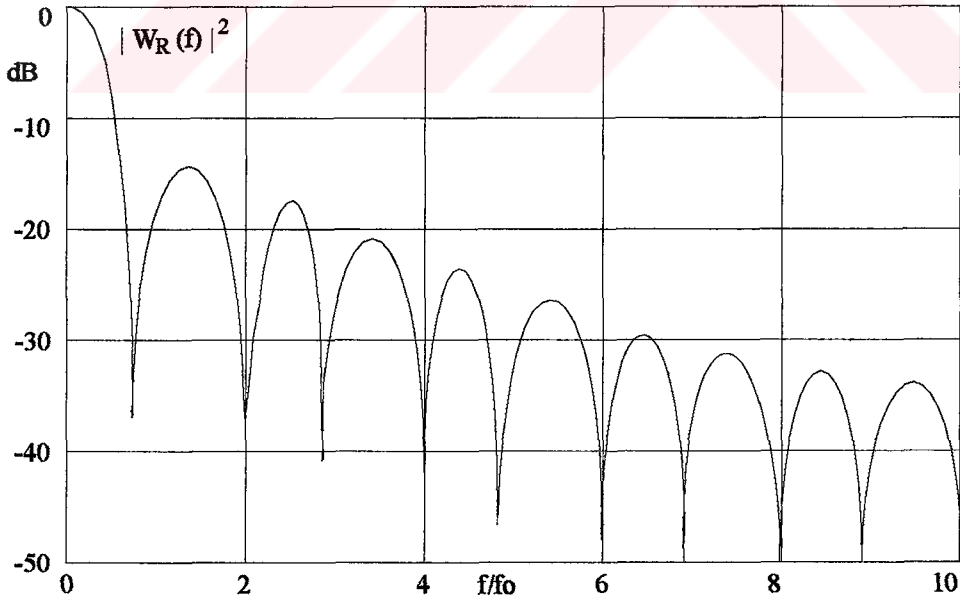
Şekil 3.15 ve 3.16'dan görüldüğü gibi, sinyalin enerjisinin büyük bir kısmını kapsayan ana kulaklar ve enerjinin diğer kısmını kapsayan yan kulaklar, güç



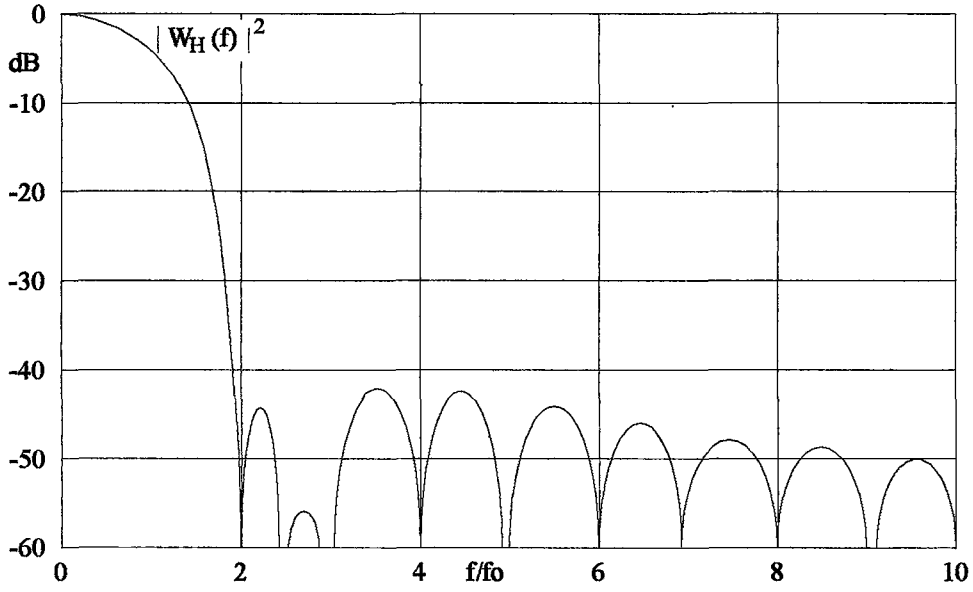
Şekil 3.14. Dikdörtgen Pencere ve Hamming Pencere.

spektrumlarını oluştururlar. (3.50) ifadesinde verilen konvolüsyon işlemi sözcüğe ait spektrumu düzeltmektedir. Pencerenin belirlenmesinde en önemli faktör,  $W(0)$ 'a göre  $W(f)$ 'in relatif büyüklüğüdür. Yan kulaklardaki enerjilerin sıfır olmamaları sebebi ile, çalışılan frekanslardan uzak frekanslarda, güç spektrumunun bileşenlerinde bazı hatalar ortaya çıkmaktadır. Bu hatalar sızıntı olarak isimlendirilir ve minimize edilmeleri gerekir. Bu sebep ile bir pencere fonksiyonu aşağıdaki şartı sağlamalıdır.

$$\left| \frac{W(f)}{W(0)} \right| < \epsilon \quad |f| \geq F_0 \quad \text{ve} \quad \epsilon \ll 1 \quad (3.51)$$



Şekil 3.15. Dikdörtgen Pencerenin Güç spektrumu.



Şekil 3.16. Hamming Pencerenin Güç Spektrumu.

Burada  $F_0 = 1/\tau$  sözcüğe ait işaretin ilk formantıdır.  $\tau$  ise ilgili aralığın perde(pitch) periyodudur.  $\epsilon$  genellikle, uygun sonuçlar için  $\epsilon \leq 0.002$  olarak seçilir (Makhoul ve Wolf, 1972). Bu ise  $W(f)$ 'in  $f \geq F_0$  için,  $W(0)$ 'dan 34dB aşağıda olması demektir.

Hamming pencerenin güç spektrumu şeklinden görüleceği gibi,  $f_0$  frekansının en az iki katı frekanslar ( $f(0) = 1/\tau'$ , burada  $\tau' = 2T_w$ 'dir) için  $|W_H(f)|$ 'in genliği  $W_H(0)$ 'ın en az 40dB altındadır ( $\epsilon = 0.01$  için). Böylece (3.51) ifadesi şu sınırlamaları sağlamış olur.

$$2f_0 < F_0 \quad \text{burada } f_0 = 1/\tau' \quad (3.52)$$

$$F_0 = 1/\tau \quad (3.53)$$

Pencerenin genişliğini belirlemek için, (3.52) ve (3.53) kullanılarak

$$\tau' = 2\tau \quad (3.54)$$

ifadesi türetilebilir. Örneğin, dikdörtgen pencere için sözcüğe ait sinyalin perde (pitch) periyodu  $\tau = 10$  msn ise, pencere genişliği en az 20msn olmalıdır. Bu

durumda  $f/f_0 \geq 10$  için,  $\epsilon = 0.03$  elde edilir. Ortaya çıkan durum iki sebepten dolayı arzu edilmez:

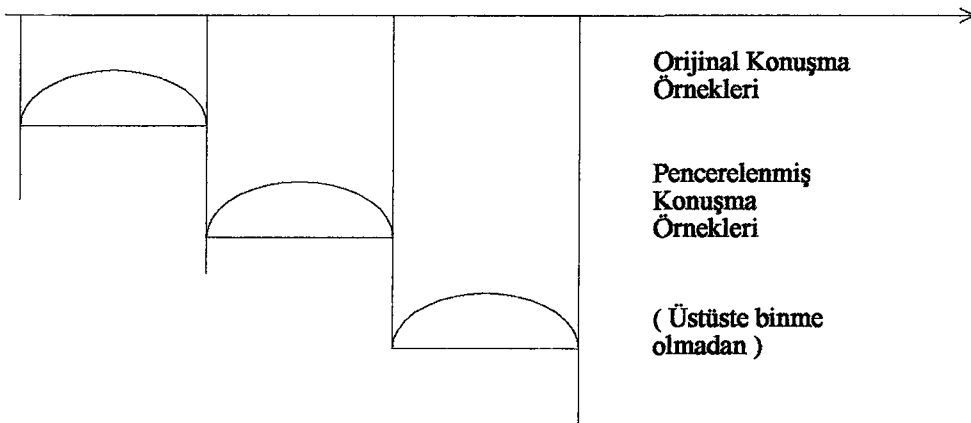
- 1)  $\epsilon$ , uygun bir pencere işlemi için büyük değere sahiptir.
- 2)  $f/f_0 \geq 10$  ifadesinin anlamı  $\tau' > 10\tau$  dur. Bir diğer şekilde ifade ile pencere genişliği perde periyodunun 10 katıdır. Bu değer ise, modelin izin verdiği aralık genişliğinden çok daha büyüktür. Bu durumda en iyi seçim,  $f/f_0 \geq 4$  için  $\epsilon = 0.1$  (20dB) alınmasıdır.

Yukarıdaki açıklamalardan çıkartılabilecek sonuç, dikdörtgen pencerede daha dar olan ana kulağın, daha iyi bir frekans çözünürlüğü sağlamasına rağmen, sızıntının çok olması sebebi ile Hamming pencereye yakın pencerelerin tercih edilmesidir (Rabiner vd, 1977).

Hamming pencere, işareti pencere sınırlarına gidildikçe sıfıra doğru zayıflatığı için, efektif pencere genişliği şöyle hesaplanabilir.

$$\tau'_e = 0.68 \tau' \quad (3.55)$$

Sözcüğe ait işaretlerin aralıklarında, üst üste binme olmaksızın, öz ilişki analizinde Hamming pencere kullanılacak olursa, pencerenin özelliğinden dolayı (uç noktalara doğru gidildikçe işaretin sıfıra yaklaşması), sözcüğe ait örneklerin önemli bir kısmı ihmal edilmiş olacaktır. Bu problem, aralıkların üst üste binmesi sağlanarak çözülebilir (Harris, 1978). bu çalışmada, aralık genişliği 300 örnek olarak alınmış ve



Şekil 3.17.Üst üste Binme Olmadan Pencereleme.

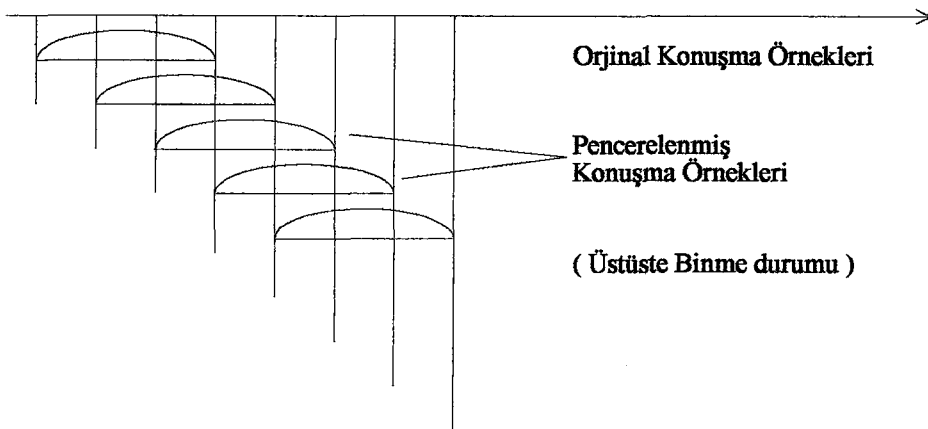
200 örnek üst üste bindirilmiştir. Üst üste binen ve binmeyen aralıkların durumu Şekil 3.17 ve Şekil 3.18'de gösterilmiştir.

### 3.9. Özellik Çıkarma.

İşaret tanıma sistemlerinde yapılması gereken önemli işlerden birisi de, önceki bölümlerde de kısmen deyinildiği gibi özellik çıkarma işlemidir. Bu çalışmada (Itakura,1975) tarafından önerilen, uzaklık ölçümüne dayalı, bir özellik çıkarma yöntemi kullanılmıştır. Bu yöntemin matematiksel ifadesi

$$d' (T(n), R_a(m)) = \log [T(n).R_a(m)] \quad (3.56)$$

şeklindedir. Burada uzaklık miktarı,  $d'(\cdot)$ ,  $T(n)$  ve  $R_a(m)$  vektörlerinin noktasal çarpımının logaritması şeklindedir.  $T(n)$  ve  $R_a(m)$  sırası ile,  $n$  ve  $m$  aralıktan oluşan test ve referans sözcüklerinin spektral temsilleridir. Bu özellik vektörleri, sekizinci dereceden D.Ö.K. modelinden elde edilmiştir. Özellik vektörlerinin kullanılma nedenlerinin anlaşılması için (3.56)'da verilen uzaklık ölçümünün, orijinal şekli ele alınmalıdır:



Şekil 3.18. Üst üste Binme Durumunda Pencereleme.

$$d = \log \left[ \frac{\sum_{i=-p}^p b'_i R_i}{\sum_{i=-p}^p b_i R_i} \right] \quad (3.57)$$

Burada  $b'_i$  ve  $b_i$  'ler (3.58) ve (3.59) eşitliklerinden elde edilir

$$b'_i = \sum_{k=0}^{p-|i|} a'_k a'_{k+|i|} \quad , a'_0 = 1 \quad \text{ve} \quad -p \leq i \leq p \quad (3.58)$$

$$b_i = \sum_{k=0}^{p-|i|} a_k a_{k+|i|} \quad , a_0 = 1 \quad \text{ve} \quad -p \leq i \leq p \quad (3.59)$$

$a'_k$  ve  $a_k$  'lar sırası ile referans ve test sözcüklerinin D.Ö.K. katsayılarıdır. Dikkat edilecek olursa  $b'_i$  'ler (3.56)'da verilen  $R_a(m)$  'lere eşdeğerdir. (3.57) ifadesinde payda, bütün sözcükler için hesaplama süresinde sabittir ve sınıflandırmayı etkilemediğinden formülden çıkarılabilir. Eğer (3.57)'de  $R_i$  yerine normalize versiyonu  $T(n)$  alınırsa (3.56) denklemi elde edilir. (3.57) ifadesindeki mesafe ölçütü;

$$d = \log \left[ \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{|A'(e^{jw})|^2}{|A(e^{jw})|^2} dw \right] \quad (3.60)$$

şeklinde yazılabilir. Burada  $A'(z)$  ve  $A(z)$  sırası ile referans sözcüğünün D.Ö.K. katsayılarına ve test sözcüğünün D.Ö.K. katsayılarına karşılık gelen ters filtrelerdir. Ayrıca  $d$ 'de test ve referans sözcük şekillerinin spektral farklarının bir ölçütüdür. Bu çalışmada, D.Ö.K. hata payı ile normalize edilmiş öz ilişki katsayıları test sözcük şeklini temsil eder.

$$T_k(n) = R_k(n) / E \quad 0 \leq k \leq p \quad (3.61)$$

Burada  $R_k(n)$ ,  $n$ 'inci aralığın  $k$ 'inci öz ilişki katsayısı ve  $E$ 'de D.Ö.K. hatasıdır.  $R_{a_k}(m)$  ise  $m$ 'inci aralığın  $k$ 'inci D.Ö.K. katsayılarının öz ilişki katsayısıdır.

$$Ra_0(m) = \sum_{i=0}^p (a_i(m))^2 \quad (3.62)$$

$\alpha_0(m) = 1$  dir ve

$$Ra_k(m) = 2 \sum_{l=0}^{p-k} \alpha_l(m) \alpha_{l+k}(m) \quad 1 \leq k \leq p \quad (3.63)$$

ve burada  $\alpha_l(m)$ ,  $m$ 'inci aralığın  $l$ 'inci  $D.\ddot{O}.K.$  katsayısıdır.

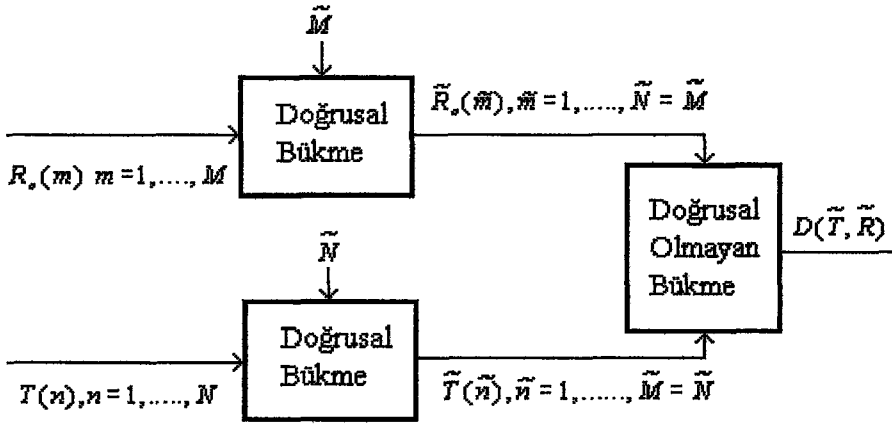
### 3.10. Dinamik Zaman Bükümü.

Referans ve test sözcüğünün zaman uyarlaması (çekme veya sıkıştırma), işlemine "Dinamik zaman Bükümü" (D.Z.B) adı verilir. Burada sistem konuşma (sözcük) hızı değişmelerini otomatik olarak kompanze eder (Sakoe ve Chiba,1978)

Doğrusal zaman bükümü ve doğrusal olmayan zaman bükümü diye iki tip hizalama metodu vardır. Doğrusal zaman bükümünde, karşılaştırılacak referans ve test sözcüklerinin şekilleri, aynı uzunlukta olacak şekilde, doğrusal olarak genişletilir veya sıkıştırılır. Fakat bir sözcüğün içindeki lineer zaman dönüşümleri, yine sözcük içinde birtakım uyumsuzluklara sebep olabilir. Böylesi durumlarda, dinamik programlama (DP) doğrusal olmayan dönüşümleri gerçekleştirerek bahsedilen uyumsuzluğu ortadan kaldırır (Dreyfus ve Law,1977). Bazı çalışmalarda iki metodun bileşimi kullanılarak daha iyi neticeler alınmıştır (Myers vd.,1980; Rabiner ve Wilpon,1981). Bu çalışmada da her iki metodun bileşimi kullanılmıştır. Bükme yöntemlerinin birlikte kullanımı Şekil 3.19.'da gösterilmiştir.

#### 3.10.1. Doğrusal Bükme.

Şekil 3.19' dan da görüldüğü gibi doğrusal bükme, dinamik zaman bükümü işlemi yapılmadan önce uygulanması gereken bir işlemdir. Doğrusal bükme, daha



Şekil 3.19. Bükme Yöntemlerinin Birlikte Kullanımı.

öncede bahsedildiği gibi referans ve test sözcük örneklerini aynı uzunluğa getirme işlemidir. Referans ve sözcük örneklerini sabit bir uzunluğa ( $\tilde{M}$ ) getirmek için gerekli formülasyon aşağıda verilmiştir.

$$Ra_k(\tilde{m}) = (1-s) Ra_k(m) + s(Ra_k(m+1)), \quad \tilde{m} = 1, 2, \dots, \tilde{M}, \quad k = 0, 1, \dots, p \quad (3.64)$$

Burada

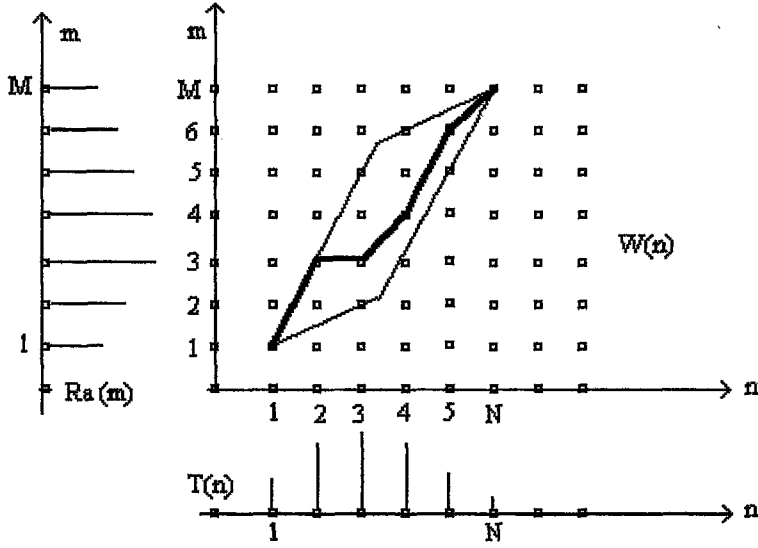
$$m = \left\lfloor (\tilde{m} - 1) \frac{M - 1}{\tilde{M} - 1} + 1 \right\rfloor \quad (3.65)$$

$$s = (\tilde{m} - 1) \frac{M - 1}{\tilde{M} - 1} + 1 - m \quad (3.66)$$

dir ve  $\lfloor x \rfloor$  ;  $x$ 'den küçük veya  $x$ 'e eşit en büyük tamsayı anlamındadır. Bu çalışmada,  $\tilde{M}$  ile gösterilen sabit uzunluk (Test ve referans sözcükleri bu uzunluğa ayarlanıyor), aralık sayısı 35 olarak alınmıştır.

### 3.10.2. Doğrusal Olmayan Bükme.

Zaman uzunluğu normalizasyonundan (Doğrusal bükme) sonra, doğrusal olmayan bir *D.Z.B* algoritması, doğrusal olarak bükülmüş sözcük örneklerine uygulanır (Rabiner ve Brown, 1983). *D.Z.B* Şekil 3.20'de görüldüğü gibi, sonlu bir



Şekil 3.20. Dinamik Zaman Bükümü.

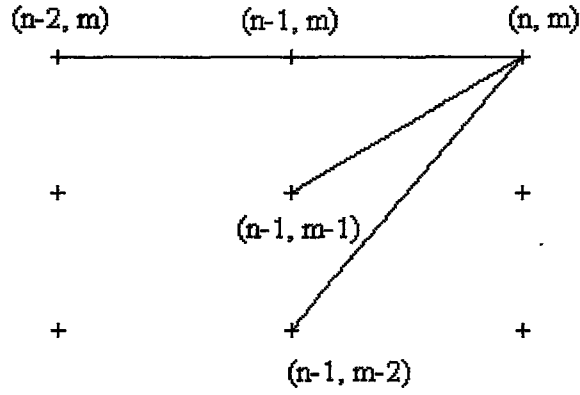
ızgara yol sisteminde patika(iz) bulma problemi olarak formüle edilebilir (Myers vd.,1980;Rabiner,1982).

Şekil 3.20’de görülen  $Ra(m)$  referans sözcüğünün,  $T(n)$  ise test sözcüğünün çok boyutlu özellik vektörüdür. Daha iyi anlaşılabilmesi bakımından,  $Ra(m)$  ve  $T(n)$ ,  $m$  ve  $n$ ’in fonksiyonu olarak ayrı ayrı da çizilmiştir. bu durumda  $D.Z.B$  problemi, optimal patika seçimine indirgenmiştir. Diğer bir ifade ile problem,  $(n,m)$  düzleminde  $Ra(m)$  ve  $T(n)$  arasındaki toplam mesafe fonksiyonu  $D$ ’yi minimize etme durumuna dönüşmüştür. Mesafe fonksiyonu  $D$ ,

$$D = \sum_{n=1}^N \tilde{d} ( T ( n ), Ra ( w ( n ) ) ) \quad (3.67)$$

olarak belirlenir. Burada  $\tilde{d}(\cdot)$ , test aralığı  $n$  ile referans aralığı  $m = w(n)$  arasındaki yerel mesafedir.

Dinamik zaman bükümü probleminin çözümü aşağıdaki belirlemeleri gerektirir (Myers vd., 1980;Rabiner ve Schimidth,1980).



**Şekil 3.21. Yerel Patika Sınırlamaları.**

**a) Son nokta sınırlamaları.**

Bu çalışmada test ve referans sözcüklerinin sınır noktaları,  $W(1) = 1$  ve  $W(N) = M$  alınarak birbirine uydurulurlar. Burada doğrusal bükme işleminin sonucunda  $N = M$  olacaktır.

**b) Yerel patika sınırlamaları.**

Şekil 3.21'de görüldüğü gibi yerel patika sınırlamaları, büküm fonksiyonu  $W(n)$ 'nin ortalama eğiminin 0.5 ile 2 arasında olacağını ve monoton olarak değişeceğini belirler. Itakura'nın (1975) patika sınırlamaları şöyledir.

$$0 \leq W(n) - W(n-1) \leq 2 \quad (3.68)$$

ve eğer

$$W(n-1) - W(n-2) > 0 \text{ ise}$$

$$W(n) - W(n-1) = 0 \quad (3.69)$$

**c) Global patika sınırlamaları.**

Global patika sınırlamalarının belirlenmesi, son nokta sınırlamaları ve yerel patika sınırlamalarını gerektirir.

$$m_L(n) \leq m \leq m_H(n) \quad (3.70)$$

Burada

$$m_H(n) = \min \left\{ 2(n-1) + 1, \left\lfloor M \cdot \frac{1}{2}(N-n) \right\rfloor, M \right\} \quad (3.71)$$

$$m_L(n) = \max \left\{ \left\lfloor \frac{1}{2}(n-1) + 1 \right\rfloor, M - 2(N - n), 1 \right\} \quad (3.72)$$

dir ve  $\lfloor x \rfloor$ ,  $x$ 'den büyük veya  $x$ 'e eşit en küçük tamsayı anlamındadır. Zaman hizalama patikalarının en geniş bölgesini ele alabilmek için  $N = M$  yapmamız gerekir. Bu suretle doğrusal zaman bükmenin önemi de anlaşılmıştır.

#### d) Eksen Yöneltilimi.

Bu çalışmada, referans sözcük indeksi  $m$ , ordinat (bağımlı değişken) olarak ve test sözcük indeksi  $n$ , ise apsis (bağımsız değişken) olarak alınmıştır.

#### e) Yerel Mesafe Ölçüsü.

Bu çalışmada yerel mesafe olarak, Itakura'nın "log ihtimal oranı" kullanılmıştır ve matematiksel olarak şöyle ifade edilebilir.

$$\tilde{d}(T(n), Ra(m)) = \log [T(n) \cdot Ra(m)] \quad (3.73)$$

Yerel mesafe ölçüsü, formülasyondan da anlaşılacağı gibi,  $T(n)$  ve  $Ra(m)$  vektörlerinin noktasal çarpımının logaritması şeklinde ifade edilmektedir.

Optimal büküm patikası ve mesafe, her sözcük için, yukarıdaki belirlemeleri de dikkate alarak, aşağıdaki şekilde hesaplanır.

$$D_i = \min \left[ \sum_{n=1}^N \tilde{d}(T(n), Ri(w(n))) \right] \quad i = 1, 2, \dots, K \quad (3.74)$$

Bu çalışmada  $K = 5$  olarak alınmıştır. Burada sözcük, ölçülen mesafenin en küçüğü esasına göre (en yakın komşu kuralı) tanınmaktadır. Önceki belirlemelere dayanarak, en iyi patikayı bulmak için, dinamik programlama metodu, (3.74) ifadesini iteratif olarak çözmek için aşağıdaki şekilde kullanılır. Yığılmış bir mesafe ölçüm fonksiyonu  $D_A(n, m)$ , (1,1) noktasından  $(n, m)$  noktasına olan en kısa toplam mesafe olarak tanımlanır ve aşağıdaki yerel sınırlamalara dayanan iterasyon kullanılabilir.

$$D_A(n, m) = \tilde{d}(T(n), Ra(m)) + \min \left\{ D_A(n-1, m)g(n-1, m), \right. \\ \left. D_A(n-1, m-1), D_A(n-1, m-2) \right\} \quad (3.75)$$

burada  $1 \leq n \leq N$ ,  $m_L(n) \leq m \leq m_H(n)$  dir ve  $g(n, m) = 1$  eğer  $w(n) \neq w(n-1)$

$$g(n,m) = \infty \text{ eğer } w(n) = w(n-1) \text{ dir.}$$

Böylece (3.74) ifadesini çözmek için istenilen sonuç

$$D = D_A(N, M) \tag{3.76}$$

olarak elde edilir.



## 4. ROBOTLAR.

### 4.1. Giriş.

Günümüzde, üretimlerini yeni teknoloji kullanarak gerçekleştiren gelişmiş ülkelerin üretim teknolojilerinde yeni bir aşama gözlenmektedir. Bu ülkelerde endüstriyel üretim için tasarlanmış ve üretilmiş robotlar, üretim süreçlerinin çeşitli kademelerinde yer almaktadırlar.

Endüstriyel iş robotları kullanmanın ekonomik yararlar sağladığı görüldükten sonra robotlar hızla yaygınlaşmış ve iş robotları, üretim süreçlerinde yerlerini alırken, bugün “robotik” olarak adlandırdığımız bilim dalı ortaya çıkmıştır.

İş robotu, istenen yere, istenen yönde ulaşabilen genel amaçlı bir mekanik kol ve bu kolu hareket ettiren denetim birimlerinden oluşur. Robot çevresindeki değişimleri, denetim birimindeki bilgisayarın programı ile izleyip değerlendirir ve mekanik kolunu programlandığı biçimde hareket ettirebilir. Robot, sadece belli bir işi yapmak üzere tasarlanan, otomat makinalarından bu özelliği ile ayrılmaktadır.

Bugün robot kullanımını cazip hale getiren pek çok sebep vardır. Her şartta çalışabilmeleri, yüksek hassasiyetleri, hızlı oluşları, yorulma gibi insana has sorunlarının olmayışı, ilk yatırımdan sonra işletme masraflarının, işgücü giderlerinden çok daha düşük olması bu sebeplerin başlıcaları olarak sayılabilir.

Günümüzde robotların denetim birimlerinde, mikroişlemciler ile geniş ölçekli tümleşik devreler kullanılmaktadır. Gelecekte kurulacak olan bilim toplumu da göz önüne alındığında, endüstride çalışacak olan teknik elemanların, teknolojik olarak ileri düzeyde bilgi seviyesine ulaşması gerekecektir.

## 4.2. Robotların Kullanım Alanları.

Robotların endüstride kullanımını hazırlayan temel etkenler, robotların insan gücüne oranla daha ucuz olması, tehlikeli ve sağlığa zararlı ortamlarda kullanılabilmesi, üretim hızını arttırması, kaliteyi yükseltmesi, üretimi esnekleştirmesi, malzeme kaybını ve işgücü döngüsünü azaltmasıdır. Bu etkenlerden, endüstrideki kullanımı en belirleyici olanları, robotların insan işgücüne oranla ucuzluğu ve sağlığa zararlı ortamlarda çalışabilmeleridir.

Robotları uygulama alanlarına göre beş grupta toplayabiliriz: Boyama robotları, montaj robotları, kaynak robotları, yükleme-boşaltma robotları ve işleme robotları. Robotlar, çok fonksiyonlu tekrar programlanabilir aygıtlar olduğundan, her robot yukarıdaki fonksiyonların hepsini yapabilir. Fakat, bugün endüstride, robot üreticileri robotlarını belli fonksiyonları yerine getirecek şekilde dizayn etmekte ve sınıflamaktadırlar. Bunun sonucunda daha verimli ve etkili robotlar ortaya çıkmaktadır.

## 4.3. Robotların Yapısı.

Bir robot genel olarak üç bölümden oluşur.

- 1)- Mekanik yapı. Robot dizaynında ilk önce ele alınacak kısımdır. Bu kısımda temel ilke rijitliktir. Taşıyıcı, ana kol, taşıyıcı kol, uç kolu, mafsallar, tutucu çeneleri, karşı ağırlık bölümlerinden oluşur.
- 2)- Kontrol birimi. Bu kısımda, merkezi işlem birimi ve çevre birimleri ile arabirimler bulunur. Robotun yapacağı hareketleri ihtiva eden program ile sistemin giriş

bilgilerinin alınması, çıkış bilgilerinin üretilmesi ve bilgilerin değerlendirilmesi bu kısımda yapılmaktadır.

3)- Tahrik birimleri. Bu kısımda eklemlerin hareketleri (dönme veya ilerleme) için kullanılan motorlar , hidrolik kısımlar veya pünomatik kısımlar bulunur. Bu kısımlar tek başlarına kullanılabildikleri gibi robotun yapısına bağlı olarak birlikte de kullanılabilirler.

#### **4.4 Kontrol sistemleri.**

Robotların kontrol sistemleri çok çeşitli türden olmakla birlikte en genel hali ile geribeslemeli ve geribeslemesiz kontrol olmak üzere iki başlık altında özetlenebilir. Özellikle büyük tipte ve ağır iş yapan robotlarda güçlü motorlara ihtiyaç duyulur. Bu motorların istenilen hassasiyette iş yapabilmeleri için geri besleme bilgisine ihtiyaç olur. Daha küçük tipteki robotlarda ise geri besleme bilgisine gerek duyulmayan adım motorları kullanılmaktadır.

#### **4.5 Koordinat Sistemleri.**

Koordinat sistemlerini aşağıdaki dört ana başlık altında toplayabiliriz.

- 1)- Kartezyen (lineer)koordinatlar.
- 2)- Silindirik koordinatlar.
- 3)- Polar (küresel) koordinatlar.
- 4)- Mafsallı küresel (döner) koordinatlar.

Seçilecek koordinat sistemleri manipülatörlerin yapılarına bağlı olarak, eksenleri boyunca yapacakları hareketin cinsine göre tespit edilir. Bu hareketler

doğrusal veya dönme şeklinde olabilir. Tüm hareketleri doğrusal olan manipülatörler genellikle kontrolü en kolay olanlardır. Tüm hareketleri dönme olan manipülatörlerin kontrolü zordur fakat mekanik yapıları kolaylık arz eder. Mafsallı küresel koordinatlarda çalışan manipülatörler ise daha büyük çalışma sahasına ve esnekliğe sahiptirler. Bu sistemde cisimlere alttan veya üstten kolaylıkla yaklaşılabilir.

#### **4.6. Robotların Hareket Uzayı.**

Robotların hareket uzayı çalışma hacmini belirleyen en önemli etkidir. Hareket uzayı ne kadar büyük olursa robot o kadar büyük bir hacimde içerisinde çalışır. Çalışma uzayını arttıran başlıca etkenler şunlardır.

- 1)- Manipülatör ve tutucu uzuvlarının büyüklüğü.
- 2)- Yüksek serbestlik derecesi.
- 3)- Tahrik edilecek elementer açı veya öteleme miktarının fazlalığı.

Yukarıda saydığımız ilkeler istenildiği kadar büyük tutulamaz. Uzuvların büyük yapılması robotun ağırlığını arttıracak dolayısı ile atalet kuvvetleri ve uzuvları hareket ettirmek için uygulayacağımız kuvvet miktarı artacaktır. Sistemin hızı ve ivmesinde azalma olacaktır.

Serbestlik derecesini arttırmak, sistemi karmaşık hale getirecektir. Her bir serbestlik derecesini ayrı ayrı tahrik ve kontrol etmemiz gerekir. Bu ise doğrudan maliyetin artması anlamına gelmektedir. Öteleme miktarını arttırmak ise sistemin ağırlık merkezinin büyük bir uzayı taramasına ve bu sebeple de yataklara gelen kuvvetlerin artmasına yol açacaktır.

#### **4.7. Tahrik Sistemleri.**

Robotlar genellikle elektrik, hidrolik ve pünomatik olarak tahrik edilirler. Tahrik sisteminin seçimi, robotun büyüklüğünü, hareket yeteneğini ve yük taşıma kapasitesini etkilemektedir. Bu bakımdan seçim ihtiyaca uygun olarak yapılır.

#### **4.8. Tahrik Kaynakları.**

Tahrik kaynağı olarak genellikle DC motorlar ve step motorlar kullanılmaktadır. Step motorlar ile ilgili daha geniş bilgi kısım 5.4'de verilmiştir.

## 5. KURULAN SİSTEME GENEL BAKIŞ.

Konuşmacıya ait ses işaretlerini elektriksel işarete dönüştürmek için kapasitif mikrofon kullanılmıştır.

Giriş işaretlerini filtre etmek amacı ile de Şekil 5.5 de devre şeması verilen Chbyshev tipi dördüncü dereceden alçak geçiren bir filtre yapılmıştır. Filtrenin band genişliği 4kHz dir. Ses işaretlerinin daha yüksek frekanslı bileşenleri tanımaları bakımından önemli bir avantaj sağlamadığından filtrenin bant genişliği dar tutulmuştur. Böylece aynı zamanda bozucu işaretlerin etkisi de azaltılmıştır.

Sistemdeki A/D dönüştürücüler 12 bitliktir ve giriş gerilim limitleri de  $\mp 0-5V$  arasındadır. Mikrofondan alınan ses işaretlerinin genlikleri düşük olduğundan işaretlerin kuvvetlendirilmesi için devre şeması Şekil 5.7 de görülen bir kuvvetlendirici yapılarak alçak geçiren filtrenin çıkışına bağlanmıştır.

Kuvvetlendiricinin kazancı, mikrofonta yeteri şiddette konuşulduğunda kuvvetlendirici doymaya girmeyecek şekilde ayarlanmıştır.

Step motorları sürmek için gerekli olan arabirimlerin yapımında ve limitleme arabirimlerinin yapımında opto-coupler'ler kullanılarak elektriksel izolasyon sağlanmıştır. Elektronik donanım ile ilgili diğer ayrıntılar kısım 5.2 de verilmiştir.

Kullanılan robotun mekanik yapısıyla ilgili gerekli değişiklikler yapılarak çalışmada kullanılabilir şekle getirilmiştir. Bu değişiklikler kısaca; tutucu kolun yukarı aşağı hareketini sağlayan ve pünomatik olarak çalışan valfler ile robotun daha önceki yapısında kullanılan DC motorların çıkartılıp yerine step motorların monte edilmesi şeklindedir.

Mekanik donanım ile ilgili diğer ayrıntılar kısım 5.1 de anlatılmıştır.

### 5.1. Mekanik Donanım.

Robotun yapımı bu çalışmanın konusu dışında olduğundan mevcut SCARA tipi robotun mekanik düzeneğinde gerekli değişiklikleri yapılarak tezde kullanılabilir hale getirilmiştir.

SCARA tipi robotun orijinalinde kullanılan DC motorlar çıkartılarak yerine adım açısı  $1.8^\circ$  olan step motorlar monte edilmiştir. Kullanılan step motorlara ilişkin teknik resim EK-7 de gösterilmiştir.

Tutucu kısım (tut, bırak işlemini yapan) ve bu kısmın yukarı aşağı hareketini sağlayan birim pünomatik kontrollü olarak çalışmaktadır. Pünomatik valflerin kumanda devresinde, Şekil.5.2..de görülen S1 ve S2 kontakları değiştirilerek çalışmaya uygun hale getirilmiştir. Normalde açık olan S1 kontağı kapalı hale getirilerek robotun tutucu kısmının, giriş bilgisi lojik 0 iken yukarıda kalması sağlanmıştır. S1 ve S2 kontakları sadece bir pünomatik valf ile kontrol edilebilir hale dönüştürüldüğünden mesela lojik 1 bilgisi geldiğinde, S2'den geçen basınçlı hava , S1 atmosfere açık olacağından tutucunun aşağıya doğru hareketini sağlayacaktır. Bu durum aynı zamanda daha önce bağımsız olarak kumanda edilmek zorunda olan ve iki bitlik bilgiye ihtiyaç gösteren S1 ve S2 valflerini kumandasının bir bitlik bilgi ile sağlanmasını temin etmiştir.

Tutucu kolun yukarı aşağı hareketini sağlayan pünomatik kısım ise S3 valfi'nin aktif hale gelmesi ile çalışmaktadır. Bu birimin içerisinde kullanılan bir yay vasıtası ile S3 valfi normal durumunda yani giriş bilgisi lojik 0 iken tutucu uçların açık vaziyette kalması sağlanmıştır. Giriş bilgisi lojik 1 olduğunda ise S3 valfi bir önceki durumunu değiştirerek basınçlı havanın geçmesine müsaade edecektir. Böylece tutucu kollar kapanacaktır. Giriş bilgisi lojik 0 olduğunda daha önce gerilmiş olan yayın, hava basıncının ortadan kalkması ile eski haline gelmesi sonucu tutucu uçlar açık hale gelir. Tutucu kısma ait pünomatik devre şeması şekil.5.2.de verilmiştir.

### 5.1.1. Kullanılan Robotun Yapısı.

Bu çalışmada kullanılan SCARA tipi robotun şematik gösterimi Şekil 5.1..de ve fiziksel görünümü ise Şekil 5.3. da verilmiştir.

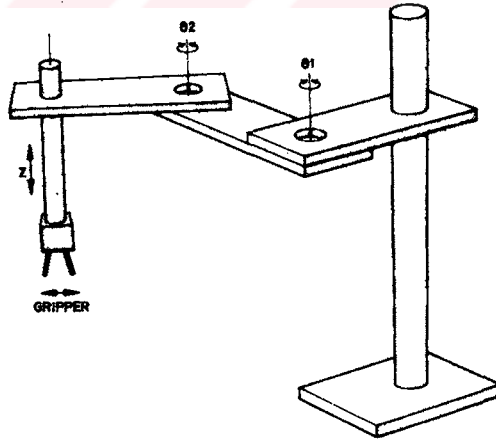
Robotun z düzlemindeki hareketi el ile kontrollüdür. Tutucu kısmın yukarı aşağı hareket miktarı 7.5cm'dir. Bu durum göz önüne alınarak öncelikle manipülatörün tutulacak olan parçaya uygun mesafeye getirilmesi sağlanır. Bu işlem yukarıda bahsedildiği gibi el kontrolünde gerçekleştirilir. Maximum hareket miktarı 50cm dir. Robota ilişkin diğer yapısal özelliklerden bazıları şöylece sıralanabilir.

Birinci kol uzunluğu = 250mm

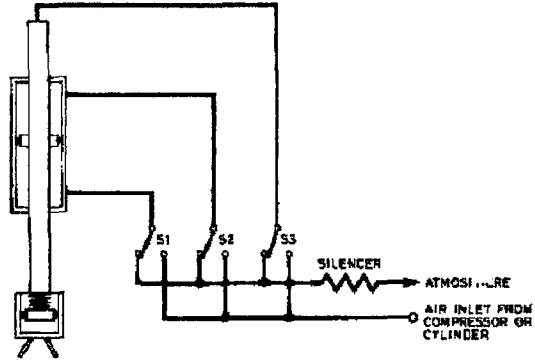
İkinci kol uzunluğu = 150mm

Birinci kolun dönme açısı ( $\theta_1$ ) Max = 200°

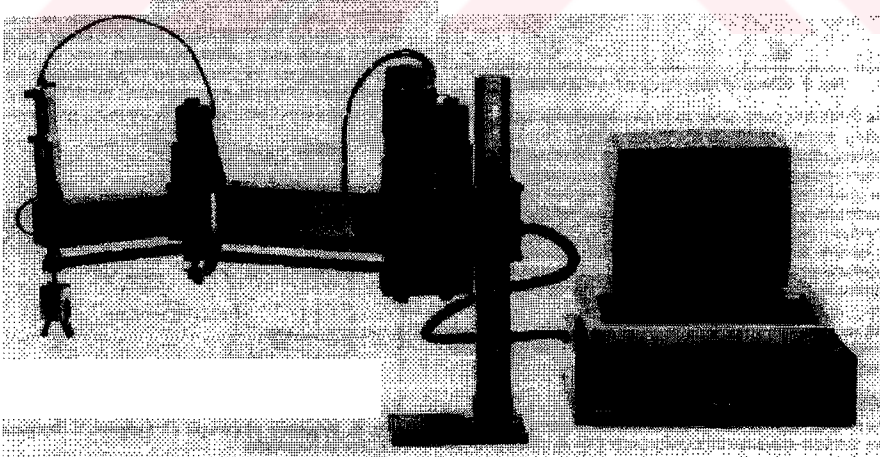
İkinci kolun dönme açısı ( $\theta_2$ ) Max = 250°



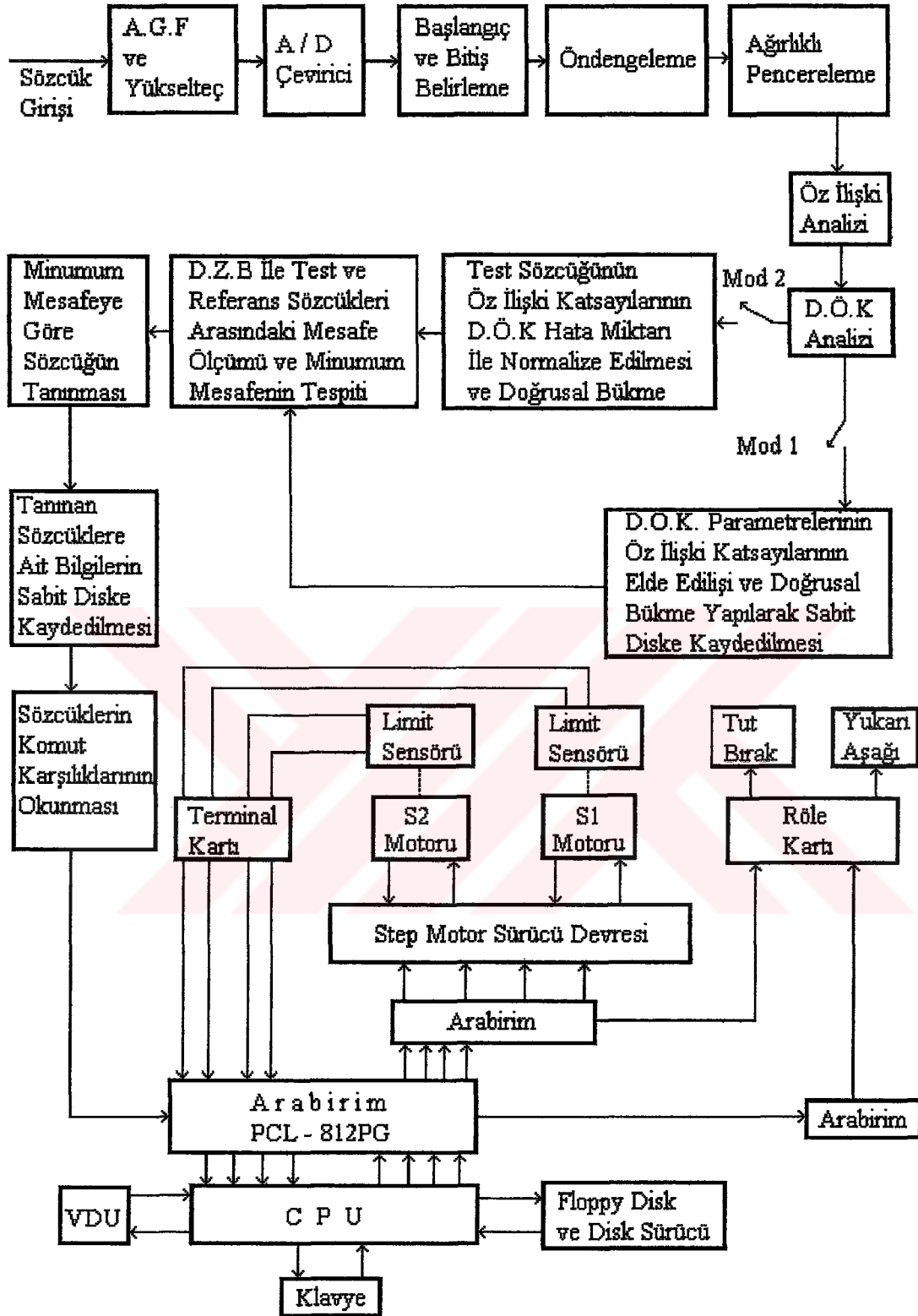
Şekil 5.1. SCARA Robotun Temsili Gösterimi.



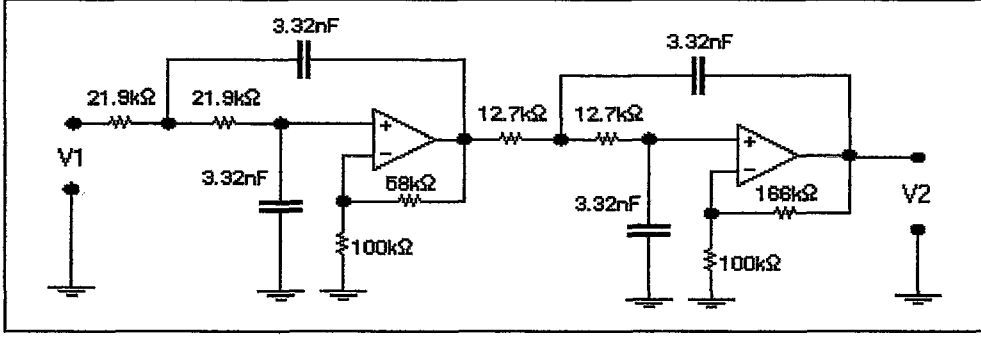
**Şekil 5.2.** Tutucu Kısım ve Pünomatik Şeması.



**Şekil 5.3.** SCARA Tipi Robotun Görünümü.



Şekil 5.4. Kurulan Sistemin Yazılım ve Donanım Blok Şeması.



Şekil 5.5. Alçak Geçiren Filtre.

## 5.2. Elektronik Donanım.

### 5.2.1. Alçak Geçiren Filtre ve Kuvvetlendirici.

Söz tanıma işlemi için, mikrofondan alınan ses işaretleri dördüncü dereceden Chebychev tipi alçak geçiren bir filtreden geçirilmiştir. Daha sonra bir kuvvetlendirici ile, filtre edilmiş işaretler PCL-812 kartının A / D dönüştürücüsüne girilmiştir. Bu çalışmada kullanılan filtreye ait devre şeması Şekil 5.5’de ve kuvvetlendiriciye ait devre şeması da Şekil 5.7’de verilmiştir.

Şekil 5.5.’de verilen eleman değerleri için gerilim oranı transfer fonksiyonu,

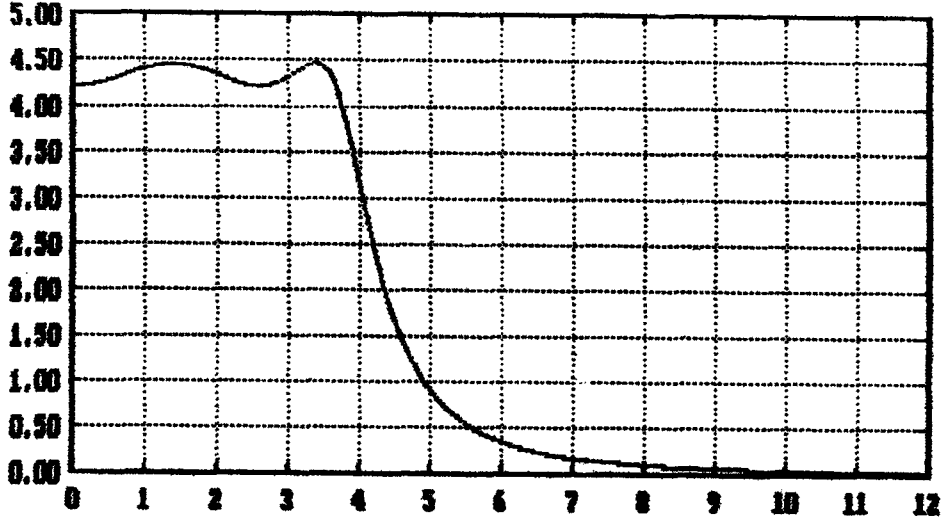
$$\frac{V_2}{V_1} = \frac{K}{S^4 + b_3 S^3 + b_2 S^2 + b_1 S + b_0}$$

biçimindedir. Burada  $K$  kazanç katsayısı,  $b_3, b_2, b_1, b_0$  dördüncü dereceden chebychev polinomunun katsayıları olup, devreden hesaplanan değerleri aşağıda verilmiştir.

$$K = 4.4718 \cdot 10^{17}$$

$$b_3 = 27593,9261$$

$$b_2 = 9.0914 \cdot 10^8$$

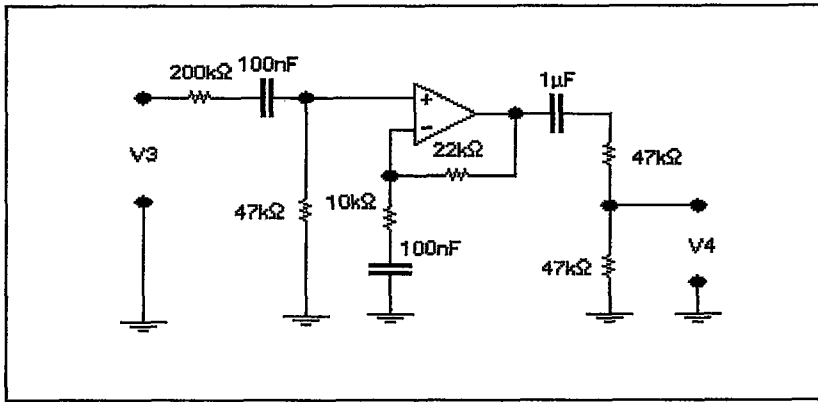


Şekil 5. 6. Filtre Karakteristiği.

$$b_1 = 1,2510 \cdot 10^{13}$$

$$b_0 = 1,0640 \cdot 10^{17}$$

Band genişliği 4khz, geçirme bandındaki zayıflama 0.5dB olan filtre karakteristiği Şekil 5.6' da gösterilmiştir.



Şekil 5. 7. Kuvvetlendirici Devre Şeması.

## 5.2.2. PCL - 812 Arabirim Kartı.

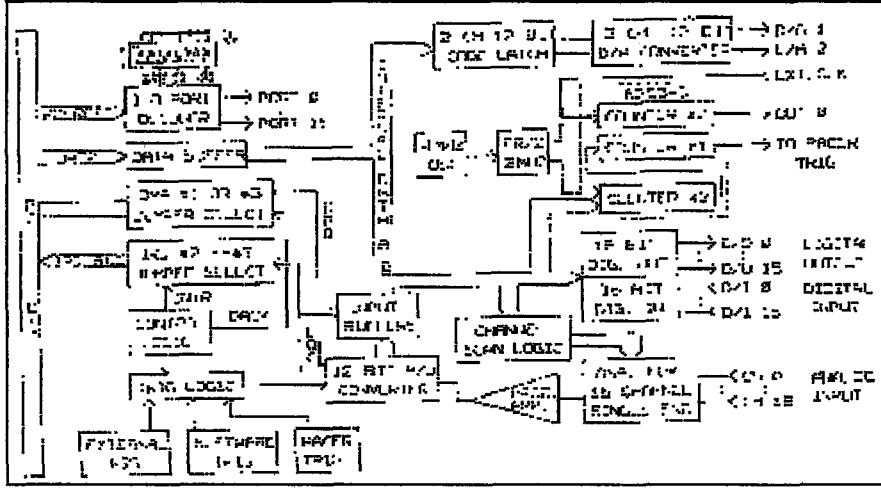
### 5.2.2.1. Giriş.

Bu çalışmada bilgisayar içindeki slota bir adet çok amaçlı arabirim kartı takılmıştır. Bu kartın A/D girişi, kuvvetlendiricinin çıkışına bağlanarak, giriş işaretleri digitale dönüştürülmüştür, kartın digital giriş-çıkış yapan kısımları ise step motorların kontrolünde ve hareket sahasını limitliyen, yani çalışma sahasının dışına çıkılıp çıkılmadığına bakan sensörlerden gelen bilgilerin değerlendirilmesinde kullanılmaktadır. Bu kart IBM , AT ve diğer uyumlu bilgisayarlarda kullanılabilen, yüksek performanslı, hızlı ve çok fonksiyonlu bir karttır. Kartın birçok ihtiyaca cevap verebilen yüksek nitelikli özellikleri ve üçüncü gurup kuruluşların geniş yazılım desteği, kartın kullanımını, birçok endüstri ve lâboratuar çalışmaları için geniş bir sahada ideal hale getirmektedir.

Bu çalışmada kullanılan PCL-812 kartının blok diyagramı Şekil 5.8'de verilmiştir.

### 5.2.2.2. PCL-812 Kartının Genel Özellikleri.

- 1- 16 tek-sonlu analog giriş kanalı.
- 2- Endüstri standardı 12 bitlik A/D dönüştürücü. DMA modunda max örnekleme hızı 30kHz'dir.
- 3- Yazılımla programlanabilir analog girişler. Bipolar:  $\pm 5V$ ,  $\pm 2.5V$ ,  $\pm 1.25V$ ,  $\pm 0.625V$ ,  $\pm 0.3125V$ .



Şekil 5. 8. PCL-812 Kartının Blok Diyagramı.

4- A/D dönüştürücünün üç adet çalışma modu vardır. Bunlar sırasıyla, yazılım ile, programlanabilir adımlayıcı ile ve harici tetikleme darbesidir.

5- Intel 8253-5 programlanabilir zamanlayıcı / sayıcı A/D'ye 0.5Mhz ile 35 dakika/darbe arasında tetikleme darbesi sağlar. Zamanlayıcının frekansı 2Mhz'dir. 16 bitlik bir sayıcı kanalı, kullanıcı uygulamaları için ayrılmıştır.

6- İki adet 12 bitlik monolitik çoğullayıcı D/A çıkış kanalı, 0-5V veya 0-10V'luk referans gerilimleri kullanılarak elde edilir. Harici referanslar başka D/A çıkışlar elde etmek için kullanılabilir.

7- TTL/DTL uyumlu 16 sayısal giriş ve 16 sayısal çıkış kanalı mevcuttur. Bu port çalışmada kullanılmıştır.

### 5.2.2.3. Yazılım Desteği.

PCI-812 kartı C, Pascal ve Basic dillerinde güçlü yazılım desteğine sahiptir. Bu çalışmada kullanılan program Qick Basic dili ile yazılmıştır.

#### 5.2.2.4. Temel adres Seçimi.

Çoğu PC çevre birimleri ve arabirim kartları I/O portlarından kontrol edilmektedir. PCL-812 kartı için I/O port temel adresi, 8 konumlu bir dip switch tarafından seçilebilir. I/O kartı birbirini izleyen 16 adet adres konumuna ihtiyaç duyar. Geçerli adresler HEX 200'den HEX 3'e0'a kadardır. Bu çalışmada HEX 220 adresi kullanılmıştır. Farklı adres kullanımı için Tablo 1'den yararlanılabilir.

#### 5.2.2.5. Pin Bağlantıları.

PCL-812 kartının arka tarafında , kolayca ulaşılabilen iki adet 20 pin'li konnektör bulunmaktadır. Bu konnektörlerden birisine digital giriş işaretlerine ait bilgiler girilmektedir. Diğerinde ise digital çıkış işaretleri bulunmaktadır. Digital bilgilerin girildiği konnektörler, temel adres+6 ve temel adres+7 adreslerini

Adres HEX	Switch Pozisyonu					
	1	2	3	4	5	6
	A8	A7	A6	A5	A4	A3
200-20F	0	0	0	0	0	X
210-21F	0	0	0	0	1	X
220-20F	0	0	0	1	0	X
220-23F	0	0	0	1	1	X
300-30F	1	0	0	0	0	X
3F0-3FF	1	1	1	1	1	X

**Tablo 5.1.** Temel Adres Seçimi.

kullanılırlar. Bu çalışmada, opto-coupler'lerden alınan bilgiler bu konnektörlere girilmiştir. Digital çıkışlar ise temel adres+13 ve temel adres+14 adreslerini kullanmaktadırlar. Şekil 5.9'da digital giriş ve çıkış konnektör pin bağlantıları görülmektedir.

### 5.2.3. Terminal Bağlantı Kartı.

Bu kart eklem sonlandırma sensörlerinden gelen bilgilerin PCL-812 kartına girişini temin etmektedir. Devre şeması ve konnektör bağlantıları EK-3'te verilmiştir.

### 5.2.4. Röle Kartı.

Bu kart PCL-812 kartından giriş bilgisi alarak, tutma ve bırakma işlemi ile tutma aparatını aşağı yukarı hareket ettiren ve pünomatik olarak çalışan kısımda, pünomatik valfleri enerjilendirmektedir. Bu amaçla kart üzerinde bulunan solit-state rölelerden yararlanılmıştır. Şekil 5.10'da röle kartı konnektör bağlantı şemaları ve Şekil 5.11'de malzeme yerleşim planı verilmiştir.

D / I	0	1	2	D / I	1	D / O	0	1	2	D / O	1
D / I	2	3	4	D / I	3	D / O	2	3	4	D / O	3
D / I	4	5	6	D / I	5	D / O	4	5	6	D / O	5
D / I	6	7	8	D / I	7	D / O	6	7	8	D / O	7
D / I	8	9	10	D / I	9	D / O	8	9	10	D / O	9
D / I	10	11	12	D / I	11	D / O	10	11	12	D / O	11
D / I	12	13	14	D / I	13	D / O	12	13	14	D / O	13
D / I	14	15	16	D / I	15	D / O	14	15	16	D / O	15
D. GND		17	18	D. GND		D. GND		17	18	D. GND	
5 V		19	20	12 V		5 V		19	20	12 V	

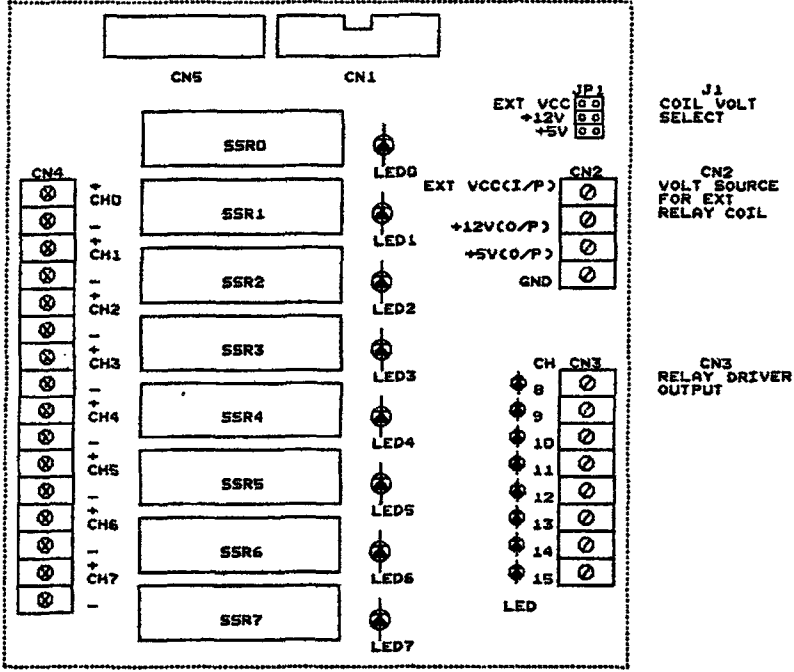
Şekil 5. 9. Dijital Giriş ve Çıkış İçin Konnektör Bağlantıları.

### 5.2.5. Eklem Limitleri Arabirim Kartı.

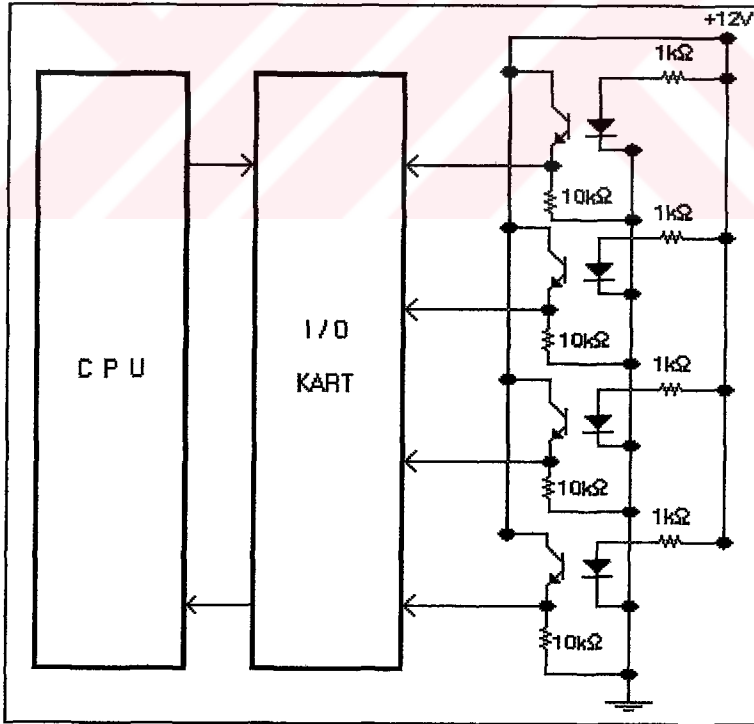
Eksenlerdeki 180° olan hareket sahasının dışına çıkılmaması gerekmektedir. Oluşabilecek herhangi bir arıza durumunda limitler dışına çıkılmak istenildiğinde, motorların zarar görmesi durumu söz konusudur. Bu durumların önüne geçilmesi amacı ile Şekil 5.12' de şeması görülen devre yapılmıştır. Kolların normal limitler arasında çalışması halinde, opto-coupler'ler iletimdedir ve alınan bilgi logic 1 seviyesindedir. Motorların hareket ettirmekte olduğu herhangi bir kol, çalışma sahasının dışında çalışmaya zorlanırsa ilgili opto-coupler'in çıkışı logic 0 durumuna geçer. PCL-812 kartının digital girişler ile ilgili port adresi HEX220+13'tür. Bu porttaki bilgi, normal çalışmada, girişlerin hepsi logic 1 durumunda bulunacaklarından 255 olacaktır. Diğer bir ifade ile bu porttaki bilgi 255 ise normal limitler arasında çalışılıyor demektir. Hex220+13 adresinden okunan bilginin 255 olup olmadığı sürekli kontrol edilir. İstenilen sonuçtan farklı bir durum ortaya çıkarsa, ilgili motor veya motorlar durdurulur. İlave olarak operatörün de durumu fark edebilmesi için bir uyarı sesi üretilmektedir.

D / I	0	1	2	D / I	1	D / I	8	1	2	D / I	9
D / I	2	3	4	D / I	3	D / I	10	3	4	D / I	11
D / I	4	5	6	D / I	5	D / I	12	5	6	D / I	13
D / I	6	7	8	D / I	7	D / I	14	7	8	D / I	15
D / I	8	9	10	D / I	9	N / C		9	10	N / C	
D / I	10	11	12	D / I	11	N / C		11	12	N / C	
D / I	12	13	14	D / I	13	N / C		13	14	N / C	
D / I	14	15	16	D / I	15	N / C		15	16	N / C	
D. GND		17	18	D. GND		D. GND		17	18	D. GND	
5 V		19	20	12 V		5 V		19	20	12 V	

Şekil 5.10. Röle Kartı Konnektör Bağlantı Şemaları.



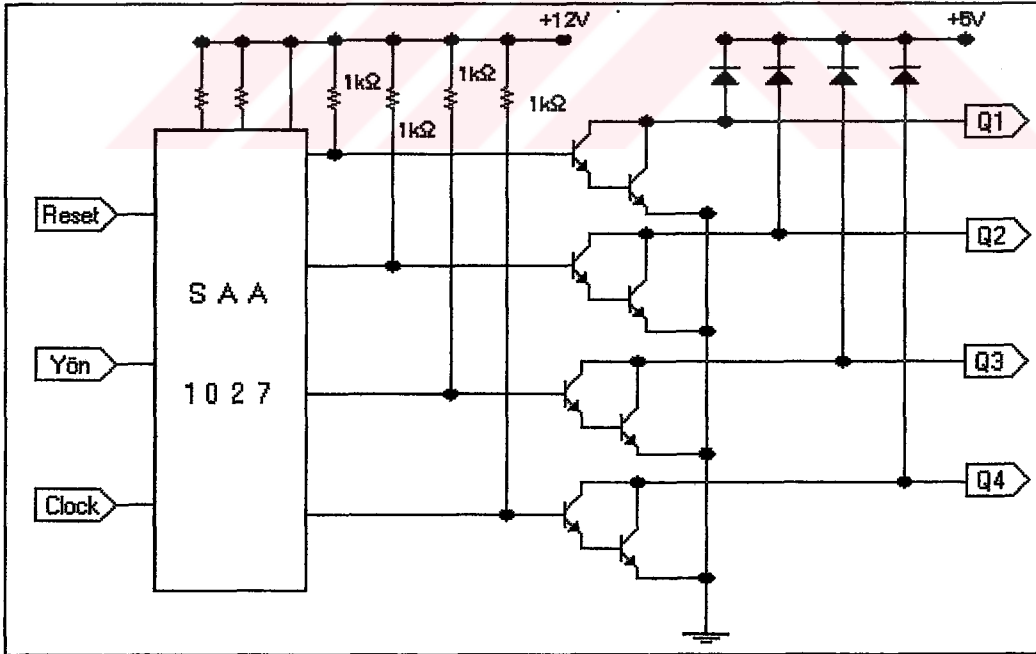
Şekil 5. 11. Röle Kartı Malzeme Yerleşim Planı.



Şekil 5. 12. Eklem Limitleri Arabirim Kartı.

### 5.2.6. Step Motor Sürücü Kartı.

Mikroişlemciden gelen bilgilere göre step motorlara kumanda eden birimdir. Step motorların sürülmesinde SAA1027 step motor sürücü entegresi kullanılmıştır. Entegre devrenin üç giriş ucu ve dört de çıkış ucu bulunmaktadır. Girişler sırasıyla clock, yön ve reset uçlarıdır. Çıkışlar ise Q<sub>1</sub>, Q<sub>2</sub>, Q<sub>3</sub> ve Q<sub>4</sub> olmak üzere doğrudan step motorlara bağlanabilmektedir. Ancak bu çalışmada kullanılan step motorların çektiği akımlar, entegrenin verebileceği akımdan fazla olduğundan motorlar ile entegre arasına akım kazancı sağlamak üzere darlington bağlantılı transistörler yerleştirilmiştir. Şekil 5.13'de step motor sürücü devresinin şeması verilmiştir. Ek-1 ve Ek-2 de ise SAA1027 entegre devresine ilişkin blok diyagramı, bacak bağlantıları ve dalga şekilleri verilmiştir.



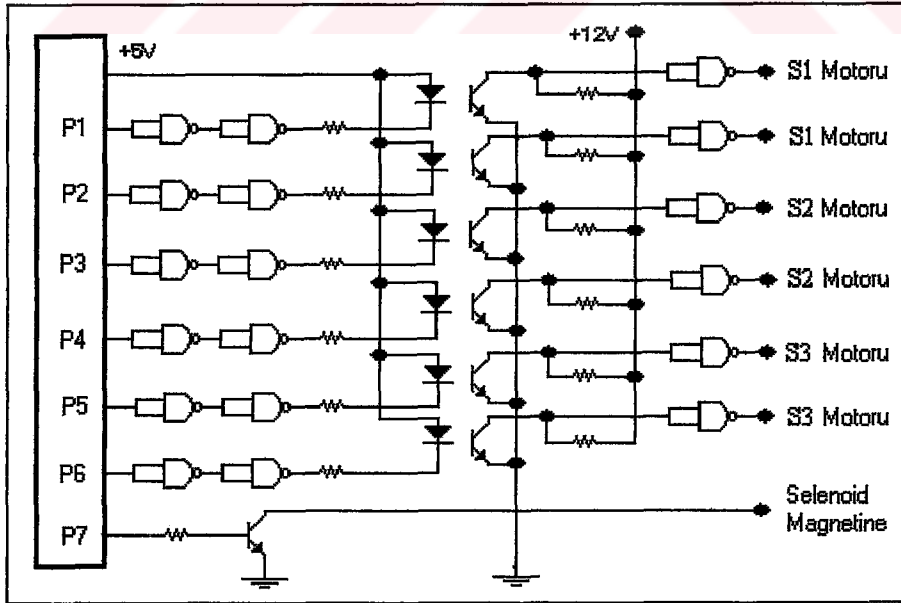
Şekil 5. 13. Step Motor Sürücü Devresi.

### 5.2.7. Step Motor Arabirim Kartı.

Bilgisayar ile step motor sürücü devresini birbirinden elektriksel olarak ayırmak için opto-coupler kullanılmıştır. Böylece gerilim seviyesi 5Volt olan bilgisayar çıkışları ile 12 volt gerilim ile beslediğimiz SAA1027 entegresi arasında uyum sağlanmıştır. Bilgisayar çıkışları iki invertörden geçirilerek opto-coupler'e bağlanmıştır. Opto-Coupler çıkışları ise yine invertörden geçirilerek sürücü kartına bağlanmıştır. Bu çalışmada kullanılan step motor arabirim kartına ait devre şeması Şekil 5.14'de verilmiştir.

### 5.3. Step Motorlar.

Robot manipülatörlerinin aksel dönme hareketlerinin oluşturulmasında DC motorlardan yararlanılabildiği gibi kontrolü daha kolay olan step motorlardan da sıklıkla yararlanır. Step motorlar, diğer tip motorlar gibi sürekli (kesiksiz) bir



Şekil 5.14. Step Motor Arabirim Kartı.

dönme hareketi yerine, adımlar halinde dönerler. Bu adımın kaç derece olacağı yada ne kadar olacağı motorun yapısına bağlıdır.

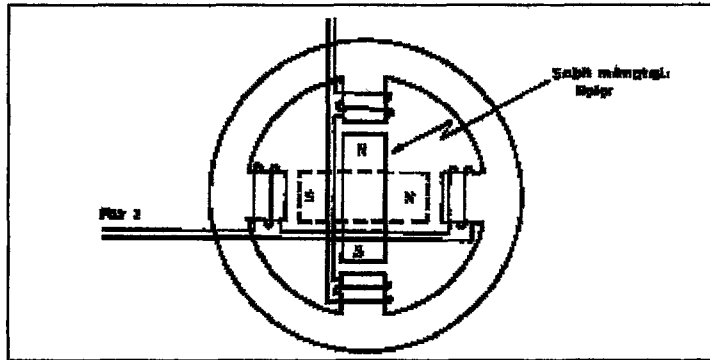
Genel olarak mekanik ve elektromagnetik olmak üzere iki ana gurubu vardır. Mekanik tipler nadir bulunduğundan sadece elektromagnetik step motorlar üzerinde duracağız. Elektromagnetik step motorlarda iki ana guruba ayrılır.

- 1- PM (Permanent-Magnet), Sabit mıknatıslı step motorlar.
- 2- VR (Variable-Reluctance), Değişken alanlı step motorlar.

### 5.3.1. Sabit Mıknatıslı Step Motorlar (PM).

Sabit mıknatıslı step motorların rotorları sabit mıknatıslıdır. Ancak sabit mıknatısın rotor olarak kullanılışı değişik biçimlerde olabilir. Şekil 5.15, 5.16 ve 5.18 bu tiplerden bazılarıdır. Şekil 5.16'deki motor hernekadar dairesel ve düz bir şekilde görünüyorsa da gerçekte hem rotor hem de stator girintili çıkıntılı biçimde yapılmışlardır. Bu tip motorlarda adım açısını küçültmek için, stator sargı sayısı ve rotor kutup sayısı artırılır. Bunun için de step motor Şekil 5.18'deki biçimde yapılır.

Şekil 5.17 step motorun çalışma prensibini açıklayan sembolik bir çizimdir. Görüldüğü gibi rotor iki kutupludur. Rotoru bulunduğu pozisyondan 2 no'lu sargının



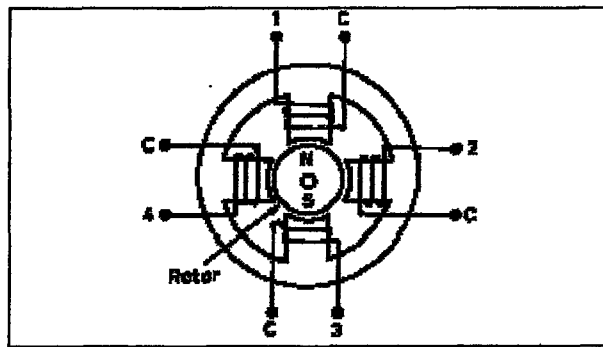
Şekil 5.15. 2 Faz 2 Kutuplu Step Motor.

karşısındaki pozisyona getirmek istersek, 2 no'lu sargıya bir pals uygulanır. Yani 2 no'lu sargıya enerji verilir. Bu durumda 2 no'lu sargıdan geçen akımın meydana getirdiği magnetik alanın etkisi ile rotor bir adım atarak belli bir derece dönme yapar. 2 no'lu sargının enerjisini kesip 3 no'lu sargı enerjilendiğinde ise rotor üçüncü adımı atarak ilk pozisyondan dördüncü sargının altındaki pozisyona kadar hareket eder.

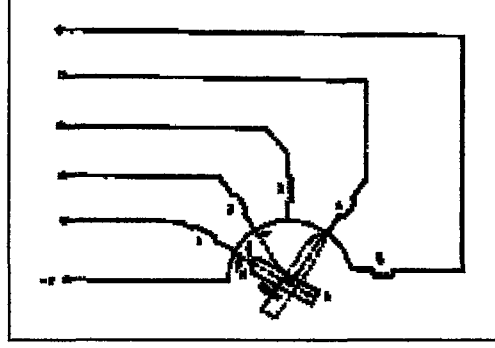
Eğer step motoru devamlı olarak döndürmek istersek, sargılara belli bir sıra dahilinde enerji uygulayıp kesmemiz gerekir.

Şekil 5.15'deki yapıda, step motorun her adımı 90 derecedir. Step motorlarda en büyük adım açısı 180 derece, bugün yapılabilen en küçük adım açısı ise 0.72 derecedir. Adım açısını küçültmek, rotorun ataletini azaltmak ve motorun toplam ağırlığını düşürmek için yapılmış step motor tipi Şekil 5.18'de görülmektedir. Rotorun daha hafif olması için klasik rotor tipi yerine disk tipi rotorlar geliştirilmiştir. Bu motorların ağırlığı diğer türlere kıyasla %60 daha az ve motor boyutları da daha küçüktür.

Disk manyetik olmayan bir maddeden yapılmıştır. Tabii mıknatıslar ise bu disk üzerine Şekil 5.18'te görüldüğü gibi bir N kutbu ve bir S kutbu şeklinde sıralanmıştır. Stator ise C tipi yumuşak demirden nüvenin üzerine faz sargılarının şekildeki gibi sarılması ile elde edilir. Faz sargılarından birinin enerjilenmesi ile



Şekil 5.16. 4 faz 2 Kutuplu Step Motor.



**Şekil 5.17. Step Motorun Çalışma Prensip Şeması.**

meydana getirilen manyetik alan vasıtası ile tabii mıknatıslı rotor etkilendirilmiş olur. 1 no'lu fazın enerjisi kesilip, 2 no'lu faz bobini enerjilendirilecek olursa rotor; rotor kutbunun 1/4'ü kadar hareket edecektir ( $90^0$ ). Eğer 2'inci faz bobininin enerjisi kesilip, 1'inci faz bobini enerjilendirilecek olursa, rotor aynı yönde bir  $90^0$  daha dönecektir. Çalıştırmaya anlatılan sırada devam edilecek olursa motorun aynı yönde dönmeye devam ettiği görülecektir.

### **5.3.1.1. Step Motor Parametreleri.**

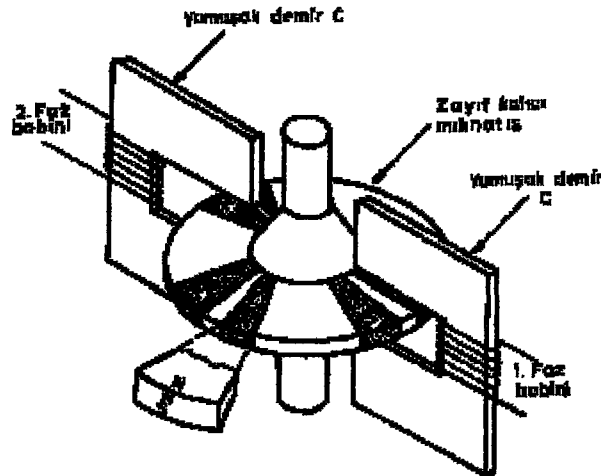
Step motorun parametreleri denildiğinde, motorun çalışma gerilimi, akımı, frekansı, devri, adım açısı, kutup ve faz sayıları ile hız-tork ilişkilerini bildiren değerler anlaşılır. Bu değerler genellikle motorun yapımçı firması tarafından, motorun katalogunda verilir.

Step motorlarda bir başka parametrede rotor ataletidir. PM motorların, motor ataleti genel olarak VR motorların ataletinden daha yüksektir. Atalet, motorun boyutları ve yapısında kullanılan malzemenin ağırlığı ile ilgilidir ve atalet momenti diye de adlandırılır.

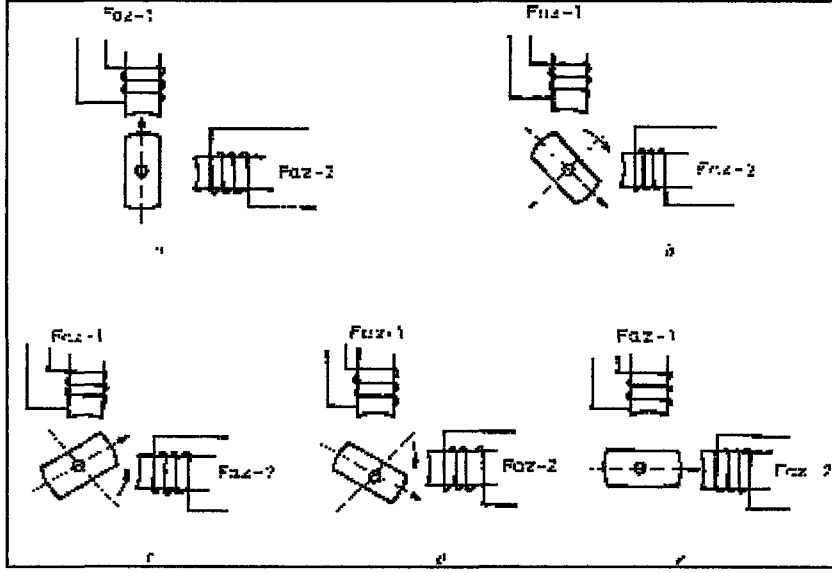
Parametrelerden bir diğeri olan adım açısı ise faz sargılarından birincisi ile ikincisinin enerjilenmesi sonucunda rotor milinin kaç derece döndüğünü belirler. Bu parametre de PM motorlarda, VR motorlardan daha büyüktür. Adım açısı 0.72 derece ile 90 derece arasındadır. Bu çalışmada adım açısı 1.8 derece olan step motorlar kullanılmıştır.

PM motorlar genel olarak 12 veya 24 kutupludurlar. Bu da, bu motordaki adım açısının 3.75 , 7.5 derece veya 15 derece olmasına müsaade eder. Kutup sayısının artırılması hem adım açısının küçülmesine hem de hızının artmasına sebep olur. Yine motorun ataleti ve boyutlarının büyüklüğü de adım hızını doğrudan etkileyen faktörlerden biridir. Motor kütlesinin artması ile orantılı olarak adım hızı azalır.

Bir başka step motor parametresi ise tutma torku olarak bilinen parametredir. Bu parametre PM (Tabii mıknatıslı) step motorlara has faydalı bir özelliktir. Tabii mıknatıslı motorlarda sargılardan birinin enerjilenmesi ile rotor hareket ederek, enerjilenen kutbun altına gelir ve burada durur. Başka bir komut verilmediği sürece bu kutbun altında sabit kalır. Bu özelliği bazı uygulamalarda son derece faydalıdır.



Şekil 5.18. Disk Tipi Rotora Sahip Step Motor.



Şekil 5.19. PM Step Motorlarda Kayma.

Bu andaki motor torku hareket halindeki torktan daha yüksektir. Ayrıca rotorun kutup sargılarının önünden geçmesi ile de sargılarda başka bir alan oluşur ve bu alandan dolayı rotor tutulmak istenir. Step motorun bu özelliği, motorun tam istenilen noktada tutulmasını sağlar.

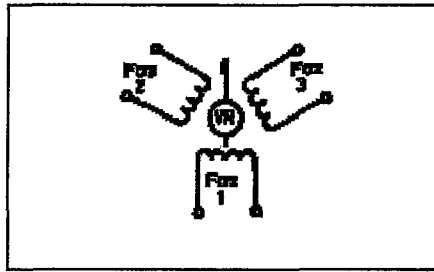
### 5.3.1.2. PM Step Motorlarda Kayma.

Şekil 5.19'deki a konumunda bulunan bir motorun ikinci sargısı enerjilendirilecek olursa, rotor ani olarak birinci durumdan ikinci duruma gider sonra üçüncü duruma sonrada dördüncü duruma gider ve daha sonra gerçek durması

gereken beşinci konuma gelir. Yani bir seferde hareket ettiği noktadan varması gereken noktaya gidip duramaz, salınım yapar. PM step motorlarda tutma ve defent torkları bu problemi ortadan kaldırırken VR step motorlarda sabit mıknatıs kullanılmadığından bu problemin çözümü başka metotlarla gerçekleştirilir.

### 5.3.2. Değişken Relüktanslı (VR) Step Motorlar.

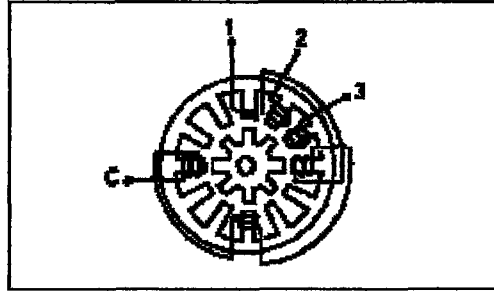
Şekil 5.20’de prensip şeması görülen değişken relüktanslı motorlar genelde üç fazlı ve (Y) biçimlidir. Bu tip motorlarda rotor olarak tabii mıknatıslar yerine ferromanyetik malzemeden ve çok kutuplu rotorlar kullanılır. Stator ise PM tabii mıknatıslı step motorlarda olduğu gibi elektromıknatıslardan oluşur. Şekil 5.21 üç fazlı olarak yapılmış on iki stator kutuplu sekiz rotor kutuplu bir VR değişken relüktanslı step motor prensip şemasını göstermektedir. Stator kutupları  $360:12=30^\circ$  aralıklı, rotor kutupları ise  $360:8=45^\circ$  aralıklı olarak yerleştirilmiştir. Bu motorun adım açısı ise  $45-30=15^\circ$  dir. O halde bir step motorun, bir adımının kaç derece olduğunu bulmak istersek, stator kutupları arasındaki açıdan rotor kutupları arasındaki açıyı çıkartmamız gerekir.



Şekil 5.20. VR Motorun Prensip Şeması.

### 5.3.2.1. VR Step Motorun Çalışma Prensibi.

Şekil 5.21'den görüleceği gibi on iki kutup sargısı üç faza bölündüğünden her faz için dört kutup kullanılmış ve bir faza ait kutuplar arasında  $90^\circ$ 'ar derecelik mekanik açı bırakılmıştır. Bir no lu faza ait bobinler enerjili olduğundan rotorun çıkıntılı kutupları, bu faza ait kutup pabuçları ile aynı doğrultuya gelmişlerdir. Bu durum, stator alanı ile rotor arasındaki manyetik relüktansın en az olduğu durumdur. Daha sonra birinci faza ait bobinin enerjisi kesilip ikinci faza ait dört bobine (şekilde gösterilmedi) enerji verildiğinde, rotorun çıkıntılı kutupları bu sefer ikinci faz sargılarının sarıldığı pabuçlarla aynı hizaya gelecektir. Bu da rotorun bir önceki konumundan başka bir konuma hareket ettiği anlamına gelir. İkinci sargının enerjisi kesilip üçüncü sargı enerjilenecek olursa bu durumda rotorun çıkıntılı kutupları üçüncü sargı pabuçları ile aynı hizaya gelecektir. İşleme böylece devam edilecek olursa motorun devamlı olarak sağa veya sola döndüğü görülür. Değişken relüktanslı motorlarda adım açısı yine stator kutup açısından rotor kutup açısının çıkarılması ile bulunur.

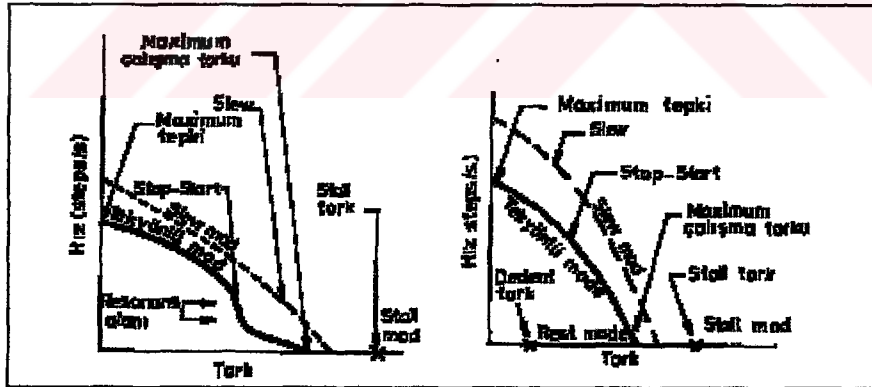


Şekil 5.21. Üç Faz (VR) Değişken Relüktanslı Step Motor.

Her iki tip step motorun tork-hız grafikleri Şekil 5.22'de verilmiştir. Bugün endüstride en yüksek hızı olan step motor (VR) değişken relüktanslı step motordur ve 18000devir/dak. hıza sahiptir.

### 5.3.3. Step Motorların Sürülmesi.

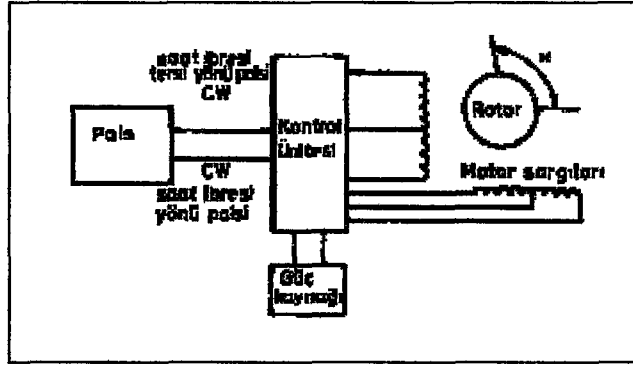
İlk step motorların fazlarına istenilen sırada ve hızda pals uygulama işlemi mekanik komütatörler yardımı ile yapılıyordu. Ancak günümüzde bu işlemler elektronik devrelerle yapılmaktadır. Bu devrelere step motor sürücü devreleri adı verilir. Bir step motor ve sürücü devresine ait blok diyagram Şekil 5.23 de verilmiştir.



a) VR Motorun Tork-Hız Grafiği

b) Tabii Mıknatıslı Motorun Tork-Hız Grafiği

Şekil 5.22. Tork - Hız Grafikleri.



Şekil 5.23. Step Motor Sürücü Devresi Blok Şeması.

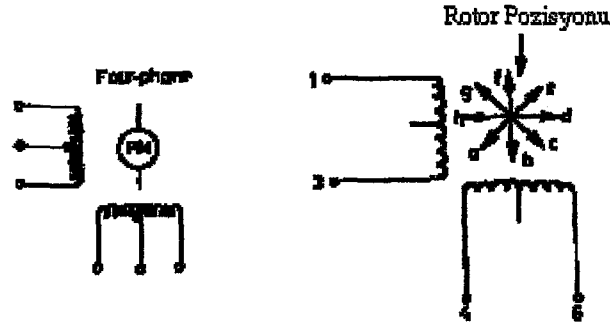
### 5.3.3.1. Step Motor Uyartım Metotları.

Stator sargılarına ve step motordan istenilen performansa bağlı olarak, step motorlar birkaç değişik metotla uyartılırlar.

- |                                    |   |
|------------------------------------|---|
| 1 - a) İki fazlı tek tek uyartım.  | b) İki fazlı iki sargının aynı anda uyartımı. |
| 2 - a) Üç fazlı tek tek uyartım.   | b) Üç fazlı iki sargının aynı anda uyartımı.  |
| 3 - a) Dört fazlı tek tek uyartım. | b) Dört fazlı iki sargının aynı anda uyartım  |

#### 5.3.3.1.1. İki Fazlı Tek Tek Uyartım.

Bu tip uyartımda motor statoru iki adet faza sahiptir. Her faz sargısı orta uçlu olmasına rağmen orta uç kullanılmaz. Sargıların çıkış uçları birleştirilip pulsler sırasıyla giriş uçlarına uygulanır. Bu uyartım metodunda faz direnci, dört fazlı uyartımdaki faz direncinin iki katı, devre akımı ve motor gücü yarı değerdedir. Sargıların tamamı kullanıldığından açığa çıkan ısı enerjisi de yüksek olur.

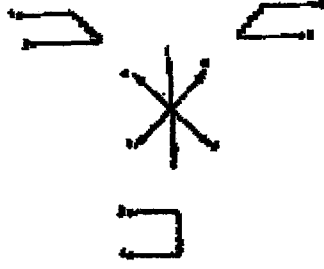


Şekil 5.24. İki Fazlı Tek Tek Uyarım.

Tablo 5.2 iki fazlı uyarım ve iki fazın aynı anda enerjilendirildiği uyarımda, enerjilenen sargıları, rotor pozisyonunu ve dönüş yönünü gösteren bir tablodur. Tablo 5.2'den anlaşılacağı gibi iki fazlı uyarımda her seferde sadece bir sargı uyarılırken her iki fazın aynı anda enerjilendirildiği uyarım da her seferde iki sargı aynı anda uyarılır.

Uyarım Modu	Enerjilenen Sargılar	Rotor Pozisyonu	Dönüş Yönü
2 fazlı tek tek Uyarım	3,1	f	
	6,4	h	CCW
	1,3	b	CCW
	4,6	d	CCW
2 fazlı 2sargının aynı anda uyarımı	3,1 ve 6,4	g	
	1,3 ve 6,4	a	CCW
	1,3 ve 4,6	c	CCW
	3,1 ve 4,6	e	CCW

Tablo 5.2. İki Fazlı Tek Tek ve Aynı Anda Uyarım.



**Şekil 5.25. Üç Fazlı VR Step Motorun Prensip Şeması.**

#### **5.3.3.1.2. İki Fazlı İki Sargının Aynı Anda Uyartımı.**

Bu metot da ise sargılar tek tek yerine aynı anda enerjilendirilir. Bu durumda motor daha güçlü olarak çalışır. Ancak bu sistem için sürücü devresi daha karmaşık ve pahalıdır.

#### **5.3.3.1.3. Üç Fazlı Tek Tek Uyartım.**

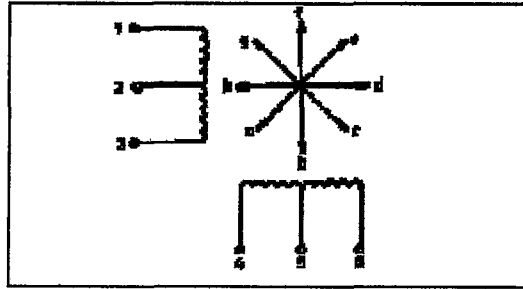
Bir çok (VR ) değişken relüktanslı step motor üç fazlı olarak imal edilirler. İki ve dört fazlı step motorlar genelde (PM) tabii mıknatıslı step motorlardır. Üç fazlı değişken relüktanslı step motor prensip şeması Şekil 5.25 de gösterildiği gibidir. Bu metotta da sargıların birer uçları ortak bağlanır, diğer uçlara ise Tablo 5.3 de belirtildiği sırada enerji uygulanır.

Uyartım Modu	Enerjilenen Sargılar	Rotor Pozisyonu	Dönüş Yönü
3 fazlı tek tek Uyartım	3, 1	a	
	6, 4	c	CCW
	1, 3	e	CCW
	4, 6		CCW
3 fazlı 2 sargının aynı anda uyartımı		g	Index
	2,1 ve 3,4	d	CCW
	3,4 ve 5,6	g	CCW
	5,6 ve 2,1		CCW

**Tablo 5.3.** Üç Fazlı Tek Tek ve Aynı Anda Uyartım.

#### 5.3.3.1.4. Üç Fazlı İki Sargının Aynı Anda Uyartımı.

Bu uyartımda da sargılara enerji, her sargıya sırayla tek tek değil de, belli sırada her iki sargıya da aynı anda uygulanır. Bu durumda rotor iki sargının oluşturduğu manyetik alan tarafından çekileceği için çekme daha güçlü olacaktır. Tablo 5.3 üç fazlı iki uyartım moduna ait bilgiler içermektedir.



**Şekil 5.26.** Dört Fazlı Uyartım.

Uyartım Modu	Enerjilenen Sargılar	Rotor Pozisyonu	Dönüş Yönü
4 fazlı tek tek Uyartım	2,1	f	Index
	5,4	h	CCW
	2,3	b	CCW
	5,6	d	CCW
4 fazlı 2sargının aynı anda uyartımı	2,1 ve 5,4	g	Index
	2,3 ve 5,4	a	CCW
	2,3 ve 5,6	c	CCW
	2,1 ve 5,6	e	CCW

**Tablo 5.4. Dört Fazlı Tek Tek ve Aynı Anda Uyartım.**

#### 5.3.3.1.5. Dört Fazlı Tek Tek Uyartım.

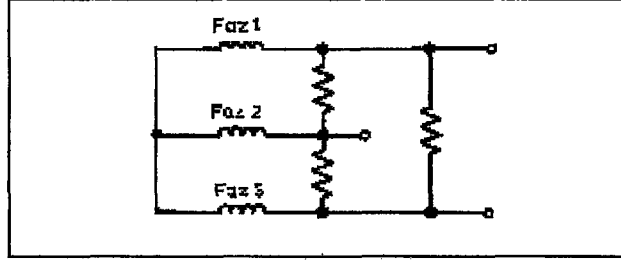
Bu tip uyartım modu, iki fazlı uyartımda kullanılmayan orta uçların da kullanıldığı uyartım seklidir. Her iki faza ait sargıların orta uçları birleştirilir ve geriye kalan ikişerden dört uç sırayla enerjilendirilir.

#### 5.3.3.1.6. Dört Fazlı İki Sargının Aynı Anda Uyartımı.

Dört fazlı iki sargının aynı anda enerjilendirildiği uyartım modunda ise dört tane giriş ucu olmasına rağmen sargılardan ikisi (her sargının yarısı) aynı anda Tablo5.4 de belirtildiği sırayla enerjilendirilir. Yine bu durumda motor gücü tek tek uyartıma göre iki kat artırılmış olur.

#### 5.3.3.2. Step Motorlarda Kaymayı Önleme Metotları.

Kısım 5.3.1.2. de kaymanın ve salınımın ne olduğu anlatılmıştı. Bu bölümde ise kaymayı önleme metotlarından bahsedilecektir. Bu metotlar şunlardır.



**Şekil 5.27.** Kaymanın Önlenmesinde Rezistif Metot.

- 1 ) Mekanik metot.
- 2) Rezistif metot.
- 3) Kapasitif metot.

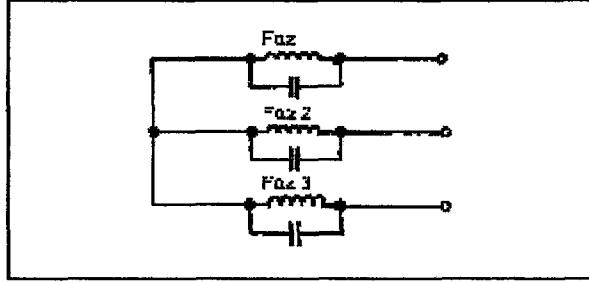
#### 5.3.3.2.1. Mekanik metot.

Temel prensip motor miline bağlı bir ağırlıktır. Volan adı da verebileceğimiz bu ağırlık adım hızını ilk anda yavaşlatmasına karşılık motorun istenilen noktayı aşmasına da izin vermez. Volan motor miline ve yaylar arasına yerleştirilmiştir. Ayarı yayların basıncının değiştirilmesi ile yapılır.

#### 5.3.3.2.2. Rezistif Metot.

Bu metotta kaymanın önlenmesi, stator sargılarına harici olarak Şekil 5.27 de gösterildiği gibi bağlanan dirençler yardımıyla sağlanır. Bu tür bağlantıda enerji uygulanacak sargının akımında hiçbir sınırlama olmazken enerjisiz kalması gereken diğer iki sargıya dirençler aracılığı ile yinede bir gerilim uygulanır. Ancak bu gerilimdirençler tarafından bölündüğünden bu sargılardan geçen akımlar daha azdır.

Bu akımların oluşturduğu manyetik alan sayesinde rotor, varması gereken noktaya gider ve salınım olmadan durur.



**Şekil 5.28.** Kaymanın Önlenmesinde Kapasitif Metot.

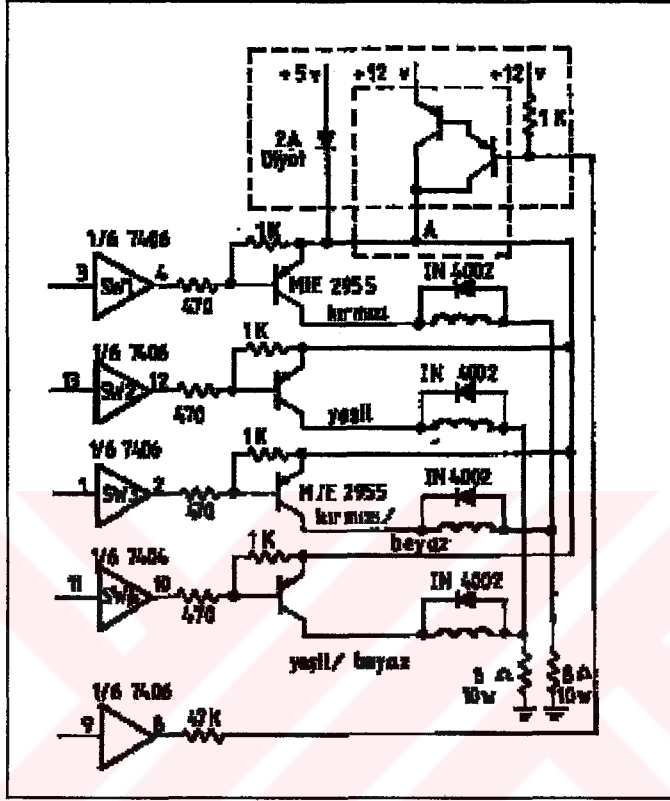
### 5.3.3.2.3. Kapasitif Metot.

Bu metotta da direnç yerine ters tork oluşturmak için kondansatörler kullanılmıştır Şekil 5.28. Önce birinci fazın enerjilendiğini ve sonrada ikinci faz, sırası ile enerjilendi kabul edelim. Birinci faza bağlı kondansatör şarj olur. Birinci fazın enerjisi kesilip ikinci faz enerjilendiğinde, birinci faza bağlı kondansatör henüz boşalmadığından, bu kondansatör birinci faz sargısı üzerinden akım geçirmeye ve o sargı kutbunda manyetik alan oluşturmaya devam edecektir. Bu enerji miktarı küçükse de yine ikinci faz sargısının etkisini az da olsa yok edecektir. Bu da motorun ani hareketini önleyecektir.

### 5.3.3.2.4. Transistörlü Step Motor Sürücü Devresi.

Transistörlü bir step motor sürücü devresinden amaç, step motorun stator

sargılarının ortak ucunu bataryanın artısına doğrudan bağlayıp, faz uçlarına uygulanacak palsların transistörler yardımı ile akım kazancı sağlanarak step motorun

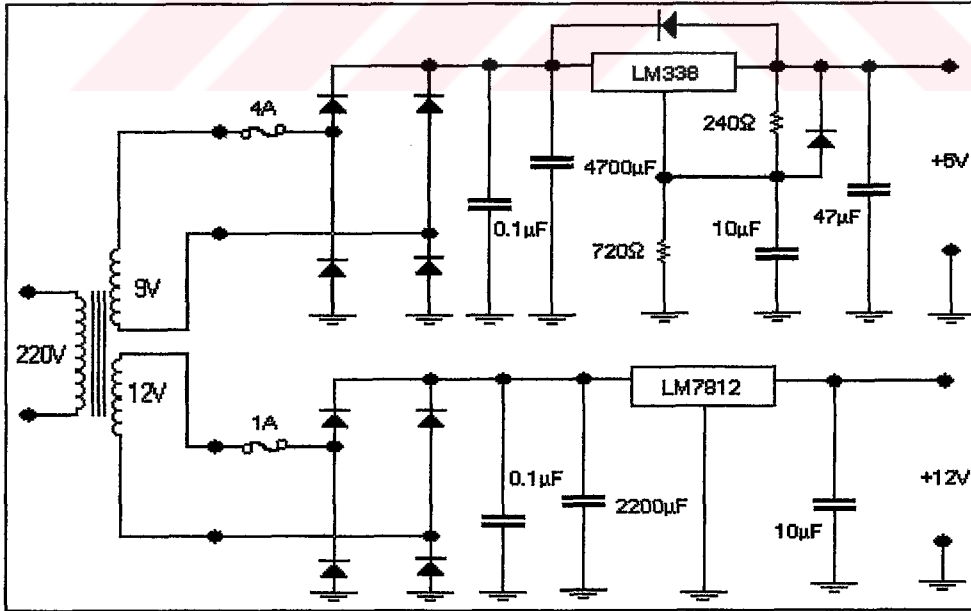


Şekil 5.29. Step Motor Sürücü Devre Şeması.

çalıştırılmasıdır. Bu amaçla ayrı elemanlardan kurulu devreler kullanılabileceği gibi özel olarak step motorun sürülmesi için yapılmış entegre devrelerden de faydalanılabilir. Step motorun kumandası bir mikrobilgisayar, mikroişlemci veya mikrokontrolör tarafından yapılmak istenirse bu durumda genel olarak entegre sürücü devrenin çıkışına transistörlerden yapılmış bir kat daha ilave etmek gerekir. Bu türden bir step motor sürücü devresi Şekil 5.29 da verilmiştir.

#### 5.4. Besleme Devresi.

Sistemin beslenmesinde 5 volt'luk ve 12 volt'luk kaynaklar kullanılmıştır. 5 volt'luk kaynak logic sistemler ile step motorların beslenmesinde kullanıldığından daha güçlü yapılmıştır. LM338 entegre devresi, 5 ampere kadar akım akıtabilen ayarlı bir gerilim regülatörüdür. Giriş veya çıkış gerilimlerinin kısa devre olması durumunda, entegrenin giriş ve çıkış uçlarına bağlı olan kondansatörlerin olumsuz etkilerini yok etmek üzere iki adet diod ile LM338'in korunması yapılmıştır (Nationale.S.C,1982). Besleme devresinin şeması Şekil 5.30'da verilmiştir.

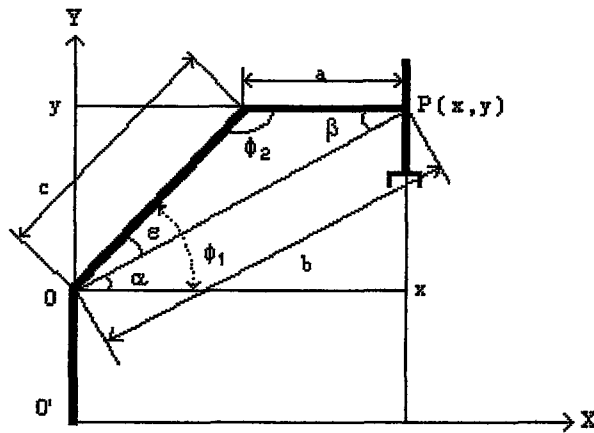


Şekil.5.30. Besleme Devresi Şeması.

### 5.5. Koordinat Sisteminde Bir Noktanın Tanımlanması.

Kısım 5.1.1 de bahsedildiği gibi robot kolunun z düzlemindeki hareketi el ile kontrol edilmektedir. ve bir sefer ayarlandıktan sonra aynı iş için yaptırılacak hareketlerde  $0-0'$  uzunluğu sabit kalacağından ve yine tutucu birimin z düzlemindeki hareketi sadece yukarı ve aşağı hareketler şeklinde olduğundan, varılması hedeflenen  $P(x,y)$  koordinatlarına ulaşılması için  $\theta_1$  ve  $\theta_2$  açılarının hesaplanması yeteli olacaktır.  $\theta_1$  ve  $\theta_2$  açıları verilirse, kol uzunlukları da robotun yapısından belli olduğu için ulaşılacak nokta yani  $P(x,y)$  koordinatları hesaplanabilir. Açılar verildiğinde  $P(x,y)$  koordinatlarını bulma işlemine kinematik, bunun tersine, verilen bir  $P(x,y)$  koordinatlarına ulaşılması için açıların hesaplanması işlemine de invers kinematik denilmektedir.

Açıların hesaplanmasında matris formunda yazılmış denklemlerin çözümünden faydalanmak mümkün isede burada analitik çözüm tercih edilmiştir (Schilling, J., 1990).



Şekil 5.31. Koordinat Sisteminde Bir Noktanın Tanımlanması.

Şekil 5.31'den görüleceği gibi P(x,y) verildiğinde , c ve a kol uzunlukları belli olduğundan sırasıyla

$$\alpha = \text{arc tan} \frac{y}{x} \quad (5.1)$$

$$b = \frac{y}{\text{Sin} \alpha} \quad (5.2)$$

$$\varepsilon = \text{arc Cos} \left[ \frac{b^2 + c^2 - a^2}{2bc} \right] \quad (5.3)$$

$$\beta = \text{arc Cos} \left[ \frac{a^2 + b^2 - c^2}{2ab} \right] \quad (5.4)$$

$$\theta_1 = \alpha + \varepsilon \quad (5.5)$$

$$\theta_2 = \pi - \varepsilon - \beta \quad (5.6)$$

hesaplanarak  $\theta_1$  ve  $\theta_2$  açıları bulunmuş olur. Bilgisayarda açılar radyan cinsinden hesaplandığından bilinen dönüşüm formülü yardımı ile dereceye çevrilirler.

Bundan sonra çözülmesi gereken problem adım motorlarının kaç adım attığında istenilen açılara ulaşılacağıdır.

Kullanılan adım motorlarının adım açıları  $1.8^\circ$  ve motorlara kuple edilen redüktörlerin dönüşüm oranları 1/180 olduğundan, step motorun bir adım atması halinde redüktör çıkışı  $0.01^\circ$  lik bir açı katetmiş olacaktır. Mesela  $10^\circ$  lik bir dönme için 1000 adım atılması gerekmektedir.

## 6. YAZILIM.

Kurulan sistemin yazılımı üç ayrı kısımdan oluşmaktadır.

- 1) Öğrenme kısmı.
- 2) Test kısmı.
- 3) Kumanda kısmı.

Sistemin yazılım olarak gerçekleştirilen kısmına ait programlar QuickBasic 4.5 derleyicisi kullanılarak Basic dilinde yazılmıştır.

Öğrenme kısmında, referans listesini oluşturacak sözcükler teker teker mikrofondan girilir. Her sözcük belirli işlemlerden geçirilerek, sözcükle ilgili gerekli katsayılar, verilecek bir kütük ismine kaydedilir. Girilen her sözcük istenilen sayıda tekrar edilerek bir referans listesi oluşturulur ve sabit diske kaydedilir. Bu çalışmada her sözcük beş'er defa tekrar edilmiştir. Daha fazla tekrarlar test kısmında sözcük tanınırken, tanıma işleminin süresinin uzamasına neden olmaktadır. Bu sebep ile tekrar sayısı sınırlı tutulmuştur.

Test kısmında ise, referans kalıpları oluşturan sözcüklerden herhangi birisi söylenerek sistem çalıştırılır. Test edilen sözcük gerekli işlemlerden geçirildikten sonra, referans kalıplarının her birisi ile teker teker bir dinamik zaman bükme yöntemi ile karşılaştırılır. Bu karşılaştırma yapılırken amaçlanan test edilen sözcüğün referans sözcüklerden hangisine en yakın olduğunu tespit etmektir (White ve Fong, 1975). Ölçülen mesafelerin minimumu saptanarak sözcük tanınır.

Kumanda kısmında, tanınan sözcüklere ilişkin bilgiler sabit diskten okunarak robotun yapacağı iş için gerekli bilgiler elde edilmiş olur.

## 6.1. Öğrenme Kısmı Yazılımı.

Oğrenme adlı altprogramın akışı maddeler halinde aşağıda sıralanmıştır.

- 1 - ORNEKALMA altprogramı çağrılarak sözcüğe ait giriş sinyalinde onbeşbin adet örnek alınır ve SOZCUK adlı diziye atılır.
- 2 - BBN isimli altprogram çağrılarak SOZCUK(15000) dizisine depolanan alınmış örneklere göre, sözcüğün başlangıç ve bitiş noktaları belirlenir. Alınan onbeşbin örneğin tamamı sözcüğe ait değildir. Örneğin ilk binbeşyüz veya ikibin örnek ve son üçbin veya dörtbin örnek genellikle boşluğa yani ortamın gürültüsüne karşılık gelmektedir. Örnekleme hızı 10kHz olduğuna göre 15000 örneği almak için 1.5 sn süre gerekmektedir. Bu süre uzun olmayan sözcükler için yeterli olmaktadır.
- 3 - Başlangıç ve bitiş örnekleri belirlenen sözcüğe ön dengeleme işlemi uygulanır. Bunun için ONDENGİ altprogramı çağrılır.
- 4 - HAMPENCERE altprogramı çağrılarak ağırlıklı pencereleme işlemi yerine getirilir.
- 5 - OZILISKI altprogramı vasıtası ile özilişki analizi yapılır.
- 6 - Doğrusal öngörü kodlaması analizi yapılarak, sözcüğün ilgili ayrılmış kısmına ait doğrusal öngörü katsayıları elde edilir. Bu işlem DOK altprogramı çağrılarak yapılır.
- 7 - Doğrusal öngörü kodlaması katsayılarının öz ilişki katsayıları OZILDOK isimli program ile elde edilir.
- 8 - Bir önceki kısımda elde edilen doğrusal öngörü kodlaması katsayılarının öz ilişki katsayıları doğrusal bükülerek bütün referans kalıpların (katsayıların) aynı sayıda değerler ihtiva etmesi sağlanır. Yani farklı uzunluktaki referans sözcüklerin hepsi aynı uzunluğa getirilir.
- 9 - Son adımda ise 8. basamakta elde edilen doğrusal bükülmüş katsayıları, verilen bir kütük ismi ile sabit diske kaydetmektir. Bu işlemi DEPOLAMA

altprogramı gerçekleştirir. Yeni bir referans kalıp oluşturulacak ise yukarıdaki işlemler baştan itibaren tekrarlanır.

## 6.2. Test Kısmı Yazılımı.

Programın kullandığı algoritma maddeler halinde aşağıda sıralanmıştır.

1 - Öğrenme kısmında olduğu gibi örnek alma işlemi ilk adımdır. Aynı altprogram kullanılarak test sözcüğüne ait örnekler alınır.

2 - Başlangıç ve bitiş noktaları belirlenir.

3 - Öndengeleme işlemi öğrenme kısmında kullanılan altprogram ile gerçekleştirilir.

4 - HAMPENCERE altprogramı ile ağırlıklı pencereleme işlemi yapılır.

5 - Özilişki analizi yapılarak işleme devam edilir.

6 - D.Ö.K. Analizi yapılarak DOKHATA hesaplanır.

7 - Beşinci adımda elde edilen öz ilişki katsayıları normalize edilerek bir diziye atılır.

8 - Normalize edilen öz ilişki katsayıları doğrusal bükülerek TEST isimli dizi oluşturulur.

9 - Daha sonra referans kalıplarının kaydedildiği bütün kütükler tek tek açılarak buradaki katsayılar ve test isimli dizideki katsayılar kullanılarak dinamik zaman bükümü işlemi yapılır ve test sözcüğü ile referans kalıplar arasındaki benzerlik mesafe olarak ölçülür. Elde edilen mesafenin minimumuna göre, test sözcüğünün referans sözcüklerden hangisi olduğu belirlenir ve sonuçta sözcük tanınmış olur.

## 6.3. Kumanda Kısmı Yazılımı.

Kumanda kısmının algoritması aşağıdaki gibi sıralanabilir.

1 - Sistem ilk çalıştırıldığında kendisini denetleyerek robot kolunu ve pünomatik sistemi başlangıç durumuna getirir ve bekler.

2 - İlk komutun “ başla “ sözcüğü olup olmadığı denetlenir.

3 - İlk komut “başla” ise “git ” komutunun söylenmesini bekler.

4 - Git komutundan sonra beklenen sözcükler iki tanedir. Bunlardan ilk söylenen x koordinatlarına karşılık gelir. İkinci söylenen ise y koordinatlarına karşılık gelir.

5 - Verilen koordinatlardan eklemlerin dönme açıları hesaplanır, Bu açıları oluşturmak için step motora uygulanacak pals sayıları hesaplanır. Sistem hareket ederek verilmiş olan koordinatlara gelir.

6 - Beklenen komut “ aşağı “ veya “ yukarı “ komutudur. Yukarı denilirse sistemde herhangi bir değişiklik olmaz. Çünkü başlangıç şartından tutucu kol yukarıda durmaktadır. “ Aşağı “ komutu girilmiş ise tutucu birim aşağıya doğru hareket eder ve durur.

7 - Beklenen komut “ tut “ komutudur. “ Bırak “ komutu söylenmiş ise tutucunun durumu değişmez. Çünkü başlangıç durumunda tutucular açık durumdadır. “ Tut “ komutu tanınmış ise malzeme tutulur.

8 - “ Yukarı “ komutu tanınarak tutucu kol yukarıya kaldırılır. “ Aşağı “ komutu tanınmış ise durumda bir değişiklik olmaz.

9 - Yeni komut beklenir. Bu komut “ git “ komutu olarak tanınmış ise (burada git komutunun kaçınıcı defa söylendiği aynı zamanda denetlenmektedir), iki parametre daha beklenir.

10 - Birinci tanınan sözcük yeni x koordinatlarına karşılıktır. İkinci tanınan sözcük ise y koordinatlarına karşılık gelmektedir. Bulunduğu koordinatlar ile yeni koordinatlar arasındaki farklar hesaplanır. Yeni açılar ve uygulanacak pals sayısı hesaplanır. Sistem yeni koordinatlara hareket eder.

11 - Beklenen komutlar “ aşağı “ olarak tanınmış ise tutucu kol aşağıya doğru hareket eder ve durur. “ Yukarı “ sözcüğü olarak tanınmış ise sistemin durumu değişmez.

12 - 11’inci adımda “ bırak “ komutu tanınırsa tutucu kollar açılarak malzeme bırakılır. “ tut “ komutunda ise sistem durumunu değiştirmez.

## 7. SONUÇLAR.

Bu tezde Türkçede seçilmiş bir grup kelimenin kullanılması ile robot kolunun hareketi kontrol edilmiştir.

Seçilen belirli kelimelerin (Başla, Git, Tut, Bırak, vb) konuşmacıya bağımlı bir ayrık sözcük tanıma algoritması ile bilgisayar kullanılarak tanınması sağlanmıştır.

Tanıma oranları seçilen sözcüklerin birbirine benzerliğine bağlı olarak değişmektedir. Farklı zamanlarda farklı sayıda yapılan denemelerde, tanıma oranının %75 ile %85 arasında değiştiği gözlenmiştir.

Mevcut SCARA tipi robotun mekanik düzeneğindeki DC motorların yerine adım açısı  $1.8^\circ$  olan step motorlar monte edilmiştir. Robotun tutma, bırakma ve tutucu kısmını yukarı aşağı hareket ettiren sistem pnömomatik olarak çalışmaktadır. Bu sisteme kumanda eden birimde yeni bir düzenleme yapılarak daha az sayıda giriş işareti ile çalışabilir hale getirilmiştir.

Sistem çalıştırıldığında kendisini otomatik olarak başlangıç (referans) konumuna getirir ve durur. Başla sözcüğünün alınması ile aktif hale gelir. Git komutu tanındıktan sonra x ve y koordinatlarının girilmesini bekler. Koordinatlar sözlü olarak verildikten sonra robot kolu hareket ederek istenilen noktaya ulaşır ve yeni bir komut almak üzere durur. Bu safhada beklenen komutlar yukarı, aşağı, tut veya bırak komutlarıdır. Aşağı kelimesi tanındığında tutucu birim aşağıya doğru hareket eder ve durur. Tut sözcüğü tanındığında malzeme tutulur. Yukarı komutu ile tutucu kol yakaladığı malzeme ile birlikte yukarıya kalkar. İkinci git komutu beklenmektedir. İkinci git komutu alındıktan sonra (bu arada kaçınıcı git komutunun alındığı yazılımda denetlenmektedir) yine iki adet (koordinatlar için) sözcük bekler. Bu iki sözcük girildikten sonra sistem verilen yeni koordinatlara hareket eder ve durur. Beklenen komutlar aşağı ve bırak komutlarıdır. Aşağı komutu ile tutucu birim aşağı hareket ederek durur, bırak komutu ile de yakaladığı malzemeyi bırakır. Sistem

böylece bir noktadan alınan malzemenin başka bir noktaya götürülmesini sağlamaktadır.

Bu sistemin tanıma kısmından , başlangıç ve bitiş örneklerini belirlemek için kullanılan algoritmanın geliştirilmesi, farklı pencereleme yöntemlerinin ve farklı aralık genişliklerin denenmesi, tanınan kelimelerin sayısının arttırılması, konuşmacıdan bağımsız sürekli söz tanıma yöntemlerinin denenmesi ile daha iyi sonuçlar alınabilecektir. Step motorların daha güçlü olanlarının kullanılması, pünomatik olarak çalışan kısımların elektriksel hale dönüştürülmesi ve tanınacak komutların sayısının arttırılması ile robotun farklı işlerde kullanılması sağlanabilir.



**KAYNAKLAR**

BAHL, L.R., JELINEK, F., ve MERCER, R.L., (1983). A Maximum Likelihood Approach to Continuous Speech Recognition. **IEEE Trans. Pattern Anal. Machine Intell.** 5: 179-190.

DREYFUS, S.E., ve LAW, A.M., (1977). **The Art and Theory of Dynamic Programming.** Academic Press. New York.

FU K.S., (1968). **Sequential Methods in Pattern Recognition and Machine Learning.** Academic Press.

HARRIS, F.J., (1978). On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. **Proc. of IEEE.** 66(1):51-83

ITAKURA, F., (1975). Minimum Residual Principle Applied to Speech Recognition. **IEEE Trans. Acoust., Speech, Signal Processing.** 23(1):67-72

ITO, M.R., ve DONALDSON, R.W., (1971). Zero Crossing Measurement for Analysis and Recognition of Speech Sounds. **IEEE Trans. on Audio and Electroacoustics**. 19(3): 235-242.

JELINEK, F., (1976). Continuous Speech Recognition by Statistical Methods. **Proc. IEEE**. 64: 532-556.

LAMEL, L.F., RABINER, L.R., ROSENBERG, A.E., ve WILPON, J.G., (1981). An Improved Endpoint Detector for Isolated Word Recognition. **IEEE Trans. Acoust., Speech, Signal Processing**. 29(4): 777-785.

MAKHOUL, J., (1975). Linear Prediction: A Tutorial Review, **Proceeding of the IEEE**. 63(4): 561-580.

MAKHOUL, J., ve WOLF, J., (1972). Linear Prediction and the Spectral Analysis of Speech. **BBN Report**. 2304.

MYERS, C., RABINER, L.R., ve ROSENBERG, A.E., (1980). Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition. **IEEE Trans. Acoust., Speech, Signal Processing.** 28(6):623-635.

NATIONALE.S.C.,(1982). **Voltage Regulator Handbook.** Nationale Semiconductor Corporation.

RABINER, L.R., ve LEVINSON, S.E., (1981). Isolated and Connected Word Recognition. Theory and Selected Applications. **IEEE Trans. on Communications.** 29: 621-659.

RABINER, L.R., ATAL, B.S., ve SAMBUR, M.R., (1977). LPC Prediction Error Analysis of its Variation With the Position of the Analysis Frame. **IEEE Trans. Acoust., Speech, Signal Processing.** 25(5):434-442.

RABINER, L.R., ve BROWN, M.K., (1982). An Adaptive, Ordered, Graph Search Technique for Dynamic Time Warping for Isolated Word Recognition. **IEEE Trans. Acoust., Speech, Signal Processing.** 30(4):535-544

RABINER, L.R., ve SAMBUR, M.R., (1975). An Algorithm for Determining the End Points of Isolated Utterances. **The Bell System Technical Journal**. 54(2):297-315.

RABINER, L.R., ve LEVINSON, S.E., ROSENBERG, A.E., ve WILPON, J.G., (1979). Speaker Independent Recognition of Isolated Word Using Clustering Techniques. **IEEE Trans. Acoust., Speech, Signal Processing**. 27(4):336-349.

RABINER, L.R., ve SCHAFER, R.W., (1978). **Digital Processing of Speech Signals**. Prentice-Hall. New York.

RABINER, L.R., ve SCHMIDT, C.E., (1980). Application of Dynamic Time Warping to Connected Digit Recognition. **IEEE Trans. Acoust., Speech, Signal Processing**. 28(4):377-388.

RABINER, L.R., ve WILPON, J.G., (1981). A Two-Pass Pattern Recognition Approach to Isolated Word Recognition. **The Bell System Technical Journal**. 60(5): 739-766.

**SCHILLING, J., (1990 ). Fundamentals of Robotics Analysis and Control.**  
**Prentice-Hall International Editions.**

**SAKOE, H., ve CHIBA, S., (1978). Dynamic Programming Algorithm Optimization  
for Spoken Word Recognition. IEEE Trans. Acoust., Speech, Signal  
Processing. 26(1):43-49.**

**SNYERS, D., (1986). Artificial Intelligence Methods For Speech Understanding.**  
**Philips Research Lab. Brussels.**

**STANTON, R.G., (1981). Bilim ve Mühendislikte Nümerik Metodlar, Çev. Prof.**  
**Dr. Çelik Aktaş, Mühendislik Bilimleri Yayınları No:5**

**PAPOULIS, A., (1965). Probability, Random Variables, and Stochastic  
Processes, McGraw-Hill Inc.**

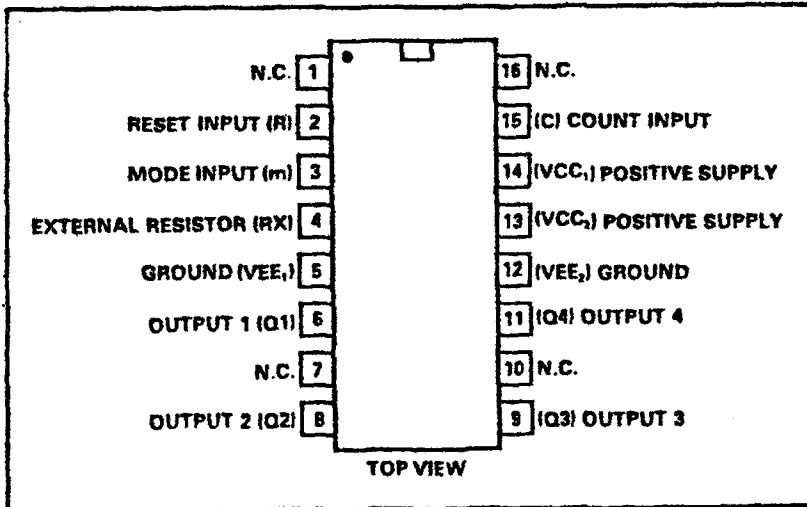
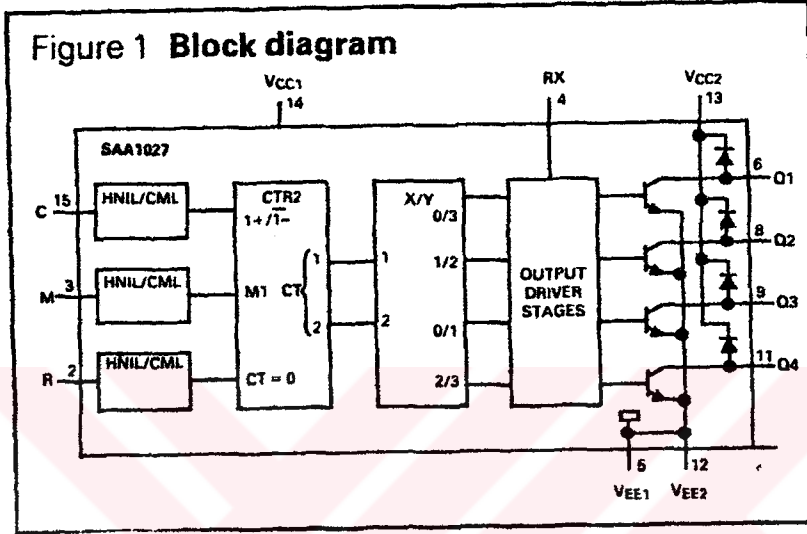
TRIBOLET, J.M., RABINER, L.R., ve SONDHI, M.M., (1979). Statistical Properties of an LPC Distance Measure. **IEEE Trans. Acoust., Speech, Signal Processing.** 27(5): 550-558.

WHALEN, A.D., (1971). **Detection Of Signals in Noise.** Academic Press.

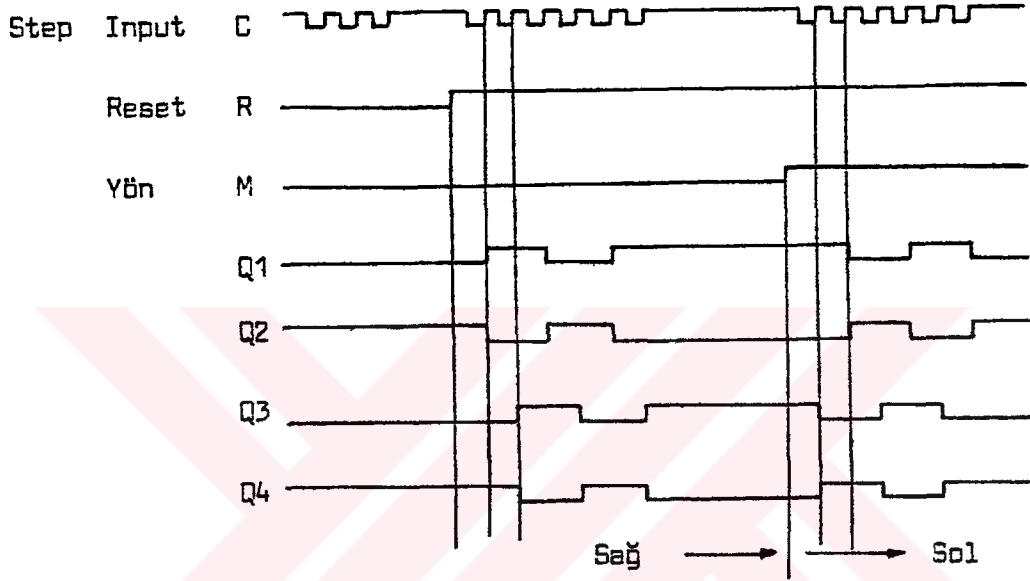
WHITE, G.M., ve FONG, P.J., (1975). K-Nearest Neighbor Decision Rule Performance in a Speech Recognition System. **IEEE Trans. on Systems, Man and Cybernetics.** 5(3): 389.

## EKLER

## EK - 1 - SAA 1027 STEP MOTOR SÜRÜCÜ ENTEGRESİNİN BLOK DİYAGRAMI VE BACAK BAĞLANTILARI.



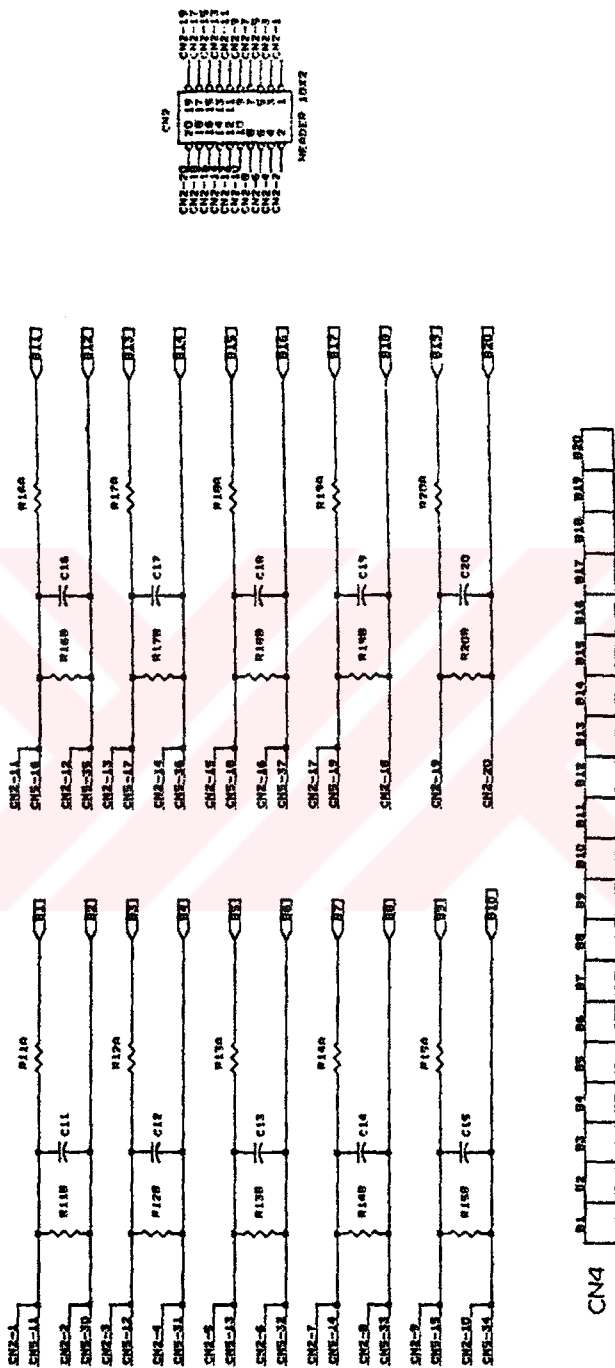
## EK - 2 - SAA1027'NİN GİRİŞ ÇIKIŞ İŞARETLERİ VE DURUM DİYAGRAMI



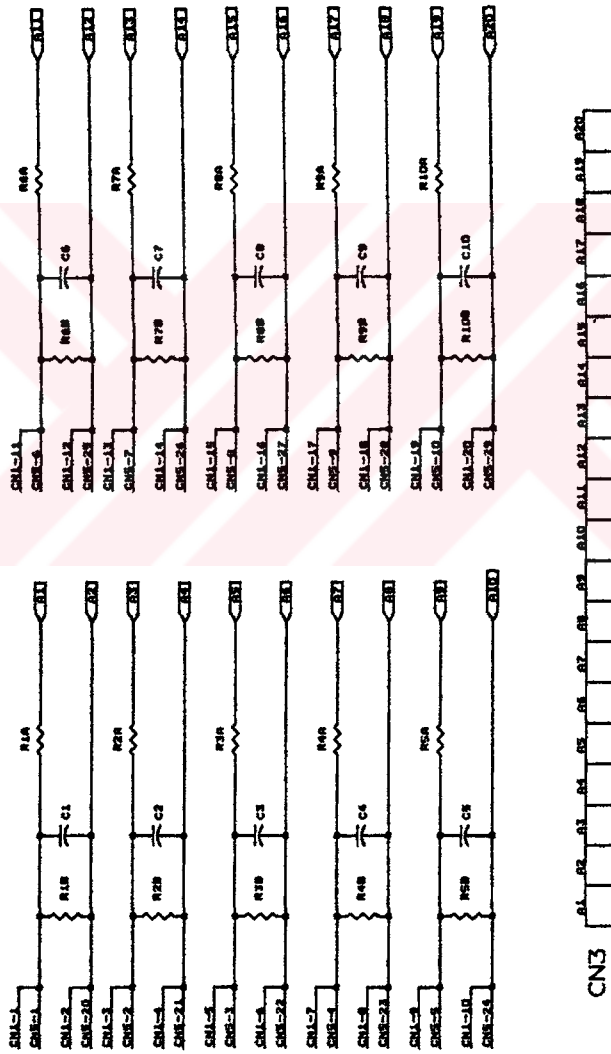
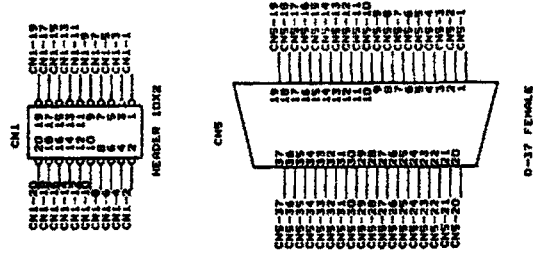
Start  
Pozisyonu

Step No	Q1	Q2	Q3	Q4
	ON	OFF	OFF	ON
1	ON	OFF	ON	OFF
2	OFF	ON	ON	OFF
3	OFF	ON	OFF	ON
4	ON	OFF	OFF	ON
5				

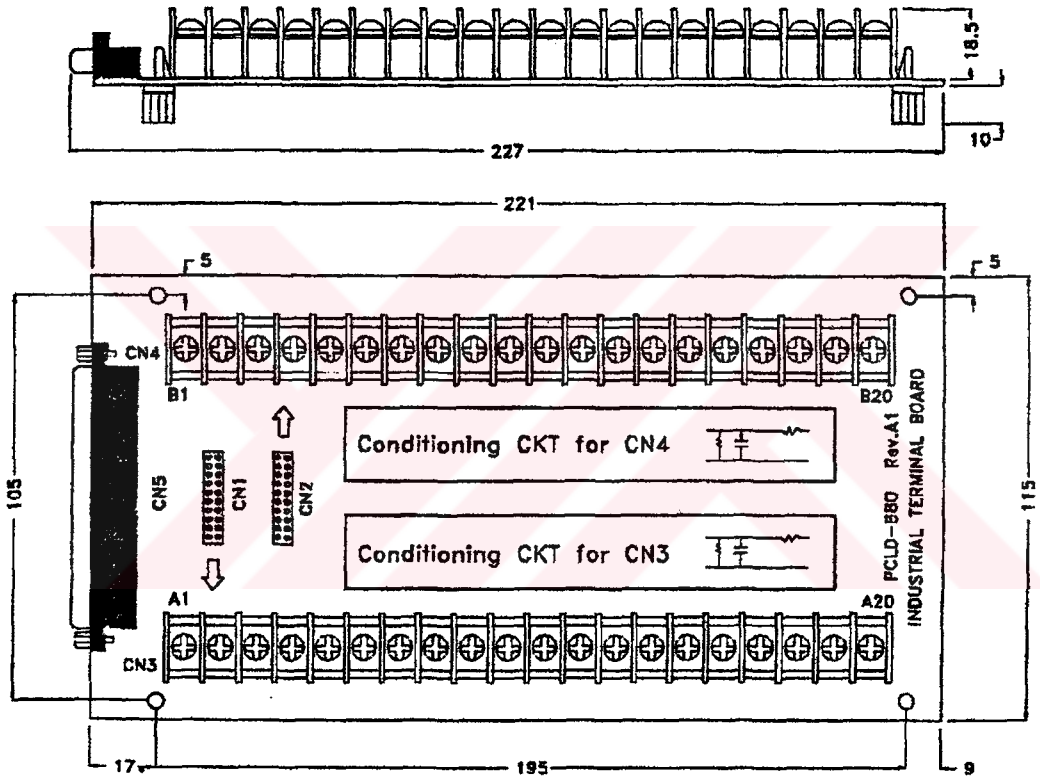
### EK - 3 - TERMINAL BAĞLANTI KARTI I.



EK - 4 - TERMİNAL BAĞLANTI KARTI II.

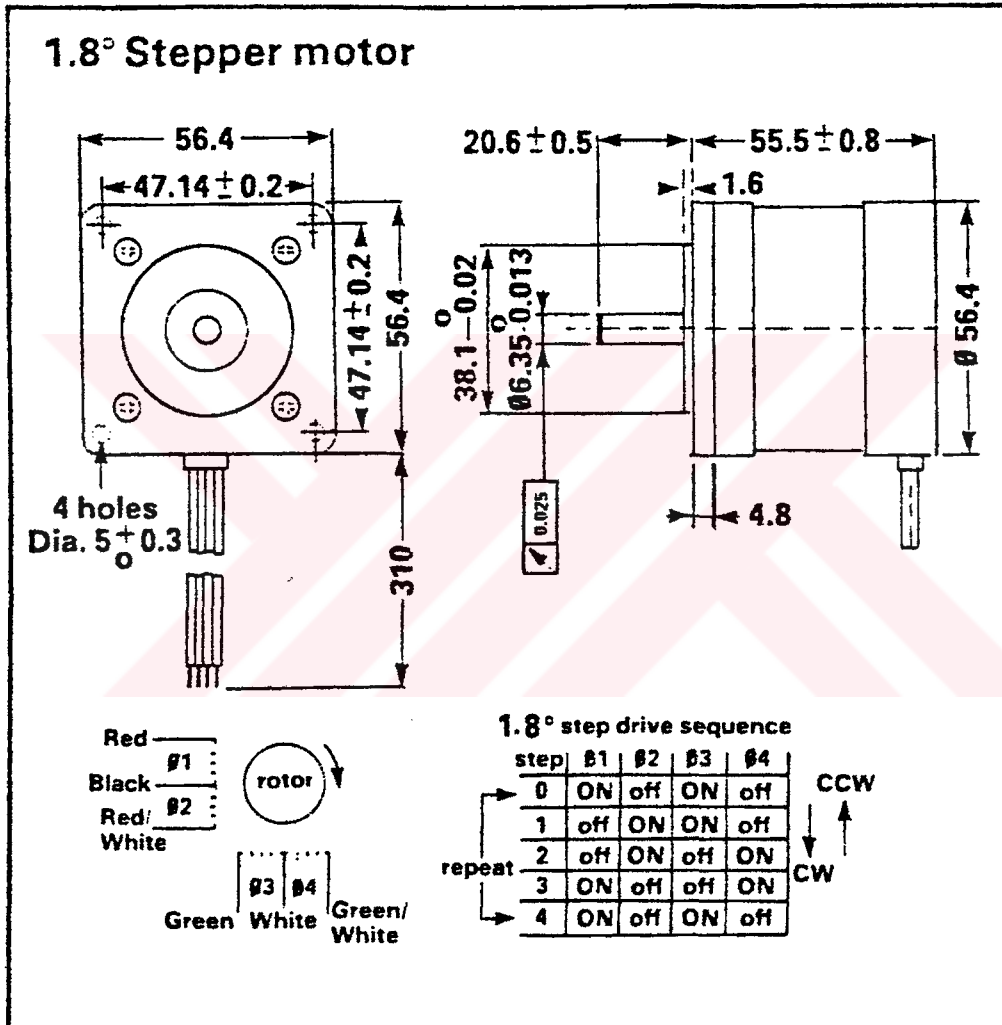


## EK - 5 - TERMİNAL BAĞLANTI KARTI MALZEME YERLEŞİM PLANI



Unit : mm

**EK - 6 - STEP MOTORUN TEKNİK RESMİ.**



### EK - 7. ÖĞRENME KISMI YAZILIMI.

REM MOD-1 : OGRENME PROGRAMI

DECLARE SUB ORNEKALMA (SOZCUK%())

DECLARE SUB BASBITNOK (SOZCUK%(), ENERJI(), SKORANI(),  
ILKOR%, SONOR%,  
KOD%)

DECLARE SUB ONDENGİ (SOZCUK%(), ORNEK%, X())

DECLARE SUB HAMPENCERE (X())

DECLARE SUB OZILISKI (X(), R())

DECLARE SUB DOK (R(), A(), P(), DOKHATA)

DECLARE SUB OZILDOK (P(), ARALIK%, RA())

DECLARE SUB DOGBUKME (RA(), DOGBUKRA(), ARALIK%)

DECLARE SUB DEPOLAMA (DOGBUKRA())

DIM RA(70, 9), ENERJI(150), SKORANI(150)

DIM SOZCUK(0 TO 15001) AS INTEGER

DIM ILKOR AS INTEGER, SONOR AS INTEGER, KOD AS INTEGER

DIM X(300), SAKX(200), DOGBUKRA(35, 9), R(9), A(9), P(9)

DIM ORNEK AS INTEGER, ARALIK AS INTEGER, N AS INTEGER, ILKAR  
AS INTEGER

DIM SONAR AS INTEGER

CLS

100 INPUT "SOZCUK GIRECEKMISINIZ ? (E/H): "; CEVAP\$

IF CEVAP\$ = "E" OR CEVAP\$ = "e" THEN

GOTO 150

ELSE STOP

END IF

150 CALL ORNEKALMA(SOZCUK())

CALL BASBITNOK(SOZCUK(), ENERJI(), SKORANI(), ILKOR, SONOR,  
KOD)

IF KOD <> 0 THEN

GOTO 100

END IF

PRINT "BASLANGIC ORNEGI = "; ILKOR

PRINT "BITIS ORNEGI = "; SONOR

ILKAR = ILKOR + 1

SONAR = ILKOR + 199

N = 1

ILK = SOZCUK(ILKOR - 1)

IKINCI = SOZCUK(ILKOR)

X(N) = IKINCI - .95 \* ILK

FOR ORNEK = ILKAR TO SONAR

N = N + 1

ILK = IKINCI

IKINCI = SOZCUK(ORNEK)

```

X(N) = IKINCI - .95 * ILK
NEXT ORNEK
ARALIK = 1
FOR ORNEK = ILKOR TO SONOR STEP 100
CALL ONDENG(SOZCUK(), ORNEK, X())
FOR N = 1 TO 200
SAKX(N) = X(N + 100)
NEXT N
CALL HAMPENCERE(X())
CALL OZILISKI(X(), R())
CALL DOK(R(), A(), P(), DOKHATA)
CALL OZILDOK(P(), ARALIK, RA())
FOR N = 1 TO 200
X(N) = SAKX(N)
NEXT N
ARALIK = ARALIK + 1
NEXT ORNEK
ARALIK = ARALIK - 1
CALL DOGBUKME(RA(), DOGBUKRA(), ARALIK)
CALL DEPOLAMA(DOGBUKRA())
GOTO 100
200 END

```

#### EK - 8. TEST KISMI YAZILIMI.

```

REM MOD-2 : TEST KISMI PROGRAMI
  DECLARE SUB ORNEKALMA (SOZCUK%())
  DECLARE SUB BASBITNOK (SOZCUK%(), ENERJI(), SKORANI(), ILKOR%,
SONOR%,KOD%)
  DECLARE SUB ONDENG (SOZCUK%(), ORNEK%, X())
  DECLARE SUB HAMPENCERE (X())
  DECLARE SUB OZILISKI (X(), R())
  DECLARE SUB DOK (R(), A(), P(), DOKHATA)
  DECLARE SUB DOGBUKME (RN(), TEST(), ARALIK%)
  DECLARE SUB KARAR (TEST(), KUTISIMS(), MH%(), ML%())
  DECLARE SUB DZB (TEST(), REFERANS(), MESAFE AS DOUBLE, MH%(),
ML%())
  DECLARE SUB FARKLILIK (TEST(), REFERANS(), N%, M%, YOL)
  DIM RN(70, 9), ENERJI(150), SKORANI(150)
  DIM SOZCUK(0 TO 15001) AS INTEGER
  DIM ILKOR AS INTEGER, SONOR AS INTEGER, KOD AS INTEGER
  DIM X(300), SAKX(200), REFERANS(35, 9), TEST(35, 9), R(9), A(9), P(9)
  DIM ORNEK AS INTEGER, ARALIK AS INTEGER, N AS INTEGER, ILKAR
AS INTEGER
  DIM KUTISIMS(25), SONAR AS INTEGER, MH%(35), ML%(35)
  CLS
100 INPUT "SOZCUK TEST EDECEKMISINIZ ? (E/H): "; CEVAP$
  IF CEVAP$ = "E" OR CEVAP$ = "e" THEN

```

```

GOTO 150
ELSE STOP
END IF
150 CALL ORNEKALMA(SOZCUK())
  CALL BASBITNOK(SOZCUK(), ENERJI(), SKORANI(), ILKOR, SONOR,
KOD)
  IF KOD <> 0 THEN
    GOTO 100
  END IF
  PRINT "BASLANGIC ORNEGI = "; ILKOR
  PRINT "BITIS ORNEGI = "; SONOR
  ILKAR = ILKOR + 1
  SONAR = ILKOR + 199
  N = 1
  ILK = SOZCUK(ILKOR - 1)
  IKINCI = SOZCUK(ILKOR)
  X(N) = IKINCI - .95 * ILK
  FOR ORNEK = ILKAR TO SONAR
    N = N + 1
    ILK = IKINCI
    IKINCI = SOZCUK(ORNEK)
    X(N) = IKINCI - .95 * ILK
  NEXT ORNEK
  ARALIK = 1
  FOR ORNEK = ILKOR TO SONOR STEP 100
    CALL ONDENGGE(SOZCUK(), ORNEK, X())
    FOR N = 1 TO 200
      SAKX(N) = X(N + 100)
    NEXT N
    CALL HAMPENCERE(X())
    CALL OZILISKI(X(), R())
    CALL DOK(R(), A(), P(), DOKHATA)
    FOR I% = 1 TO 9
      RN(ARALIK, I%) = R(I%) / DOKHATA
    NEXT I%
    FOR N = 1 TO 200
      X(N) = SAKX(N)
    NEXT N
    ARALIK = ARALIK + 1
  NEXT ORNEK
  ARALIK = ARALIK - 1
  CALL DOGBUKME(RN(), TEST(), ARALIK)
  RESTORE
  FOR I% = 1 TO 25
    READ KUTISIM$(I%)
  NEXT I%
  DATA BIR,BIR1,BIR2,BIR3,BIR4,IKI,IKI1,IKI2,IKI3,IKI4
  DATA UC,UC1,UC2,UC3,UC4,DORT,DORT1,DORT2,DORT3,DORT4

```

```

DATA BES,BES1,BES2,BES3,BES4
FOR I% = 1 TO 35
READ MH%(I%)
NEXT I%
DATA 1,3,5,7,9,11,13,15,17,19,21,23,24,24,25,25,26,26,27,27,28,28,29,29
DATA 30,30,31,31,32,32,33,33,34,34,35
FOR I% = 1 TO 35
READ ML%(I%)
NEXT I%
DATA 1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9,10,10,11,11,12,12,13,15,17,19,21
DATA 23,25,27,29,31,33,35
CALL KARAR(TEST(), KUTISIMS(), MH%(), ML%())
GOTO 100
END

```

### EK - 9. ÖRNEKALMA ALTPROGRAMI.

```

REM ORNEKALMA ALT PROGRAMI
SUB ORNEKALMA (ary1%())
10 CLS
390 ***** STEP 1: INITIALIZE DRIVER USING FUNC 0
*****
400 '
410 DIM DAT%(4), ARY2%(15000) 'DEFINE INTEGER ARRAYS
420 PORT% = &H220 'SET I/O PORT ADDRESS□
430 DAT%(0) = PORT% 'GET I/O PORT ADDRESS
440 DAT%(1) = 3 'SELECT IRQ 3
450 DAT%(2) = 3 'SELECT DMA 3
460 ER% = 0 'ERROR RETURN CODE
470 FUN% = 0 'FUNCTION 0
480 CALL PCL812(FUN%, SEG DAT%(0), SEG ary1%(0), SEG ARY2%(0),
ER%)
490 IF ER% <> 0 THEN PRINT "DRIVER INITIALIZATION FAILED!": STOP
500 '
510 ***** STEP 2: SET SCAN CHANNEL RANGE USING FUNC 1
*****
520 '
530 CHAN% = 0
535 IF CHAN% > 15 THEN PRINT "ERROR ! OVER THE SCAN CHANNEL
NUMBER (15)":
GOTO 530
540 PRINT
550 DAT%(0) = CHAN% 'SET START CHANNEL NUMBER
560 DAT%(1) = CHAN% 'SET STOP CHANNEL NUMBER
570 FUN% = 1 'FUNCTION 1
580 CALL PCL812(FUN%, SEG DAT%(0), SEG ary1%(0), SEG ARY2%(0),
ER%)
590 IF ER% <> 0 THEN PRINT "SET SCAN CHANNEL FAILED! ": STOP

```

```

600 '
610 '***** STEP 3: SET A/D RANGE USING FUN 23 *****
650 GR% = 0
660 DAT%(1) = GR%      'A/D RANGE
670 FUN% = 23          'FUNCTION 23
680 CALL PCL812(FUN%, SEG DAT%(0), SEG ary1%(0), SEG ARY2%(0),
ER%)
690 IF ER% <> 0 THEN PRINT "SET A/D RANGE FAILED! ": STOP
700 PRINT
710 '***** STEP 4: SET PACER TRIGGER USING FUNC 17
*****
720 '
730 'SET PACER=1000 HZ (2MHZ DIVIDED BY 2000 <DAT%(0)*DAT%(1)>)
740 DAT%(0) = 10      'COUNTER 2 DIVISOR
750 DAT%(1) = 20      'COUNTER 1 DIVISOR
760 FUN% = 17          'FUNCTION 17
770 CALL PCL812(FUN%, SEG DAT%(0), SEG ary1%(0), SEG ARY2%(0),
ER%)
780 IF ER% <> 0 THEN PRINT "SET PACER TRIGGER FAILED!": STOP
790 '
800 '***** STEP 5: PERFORM N A/D CONVERSIONS AND DMA
*****
810 '
820 INPUT "ALINACAK ORNEK SAYISINI GIRINIZ"; N
825 'PRINT "SOZCUK GIRMEK ICIN BIR TUSA BASINIZ";
828 'B$ = INKEY$: IF B$ = "" GOTO 940
830 DAT%(0) = N        'NUMBER OF CONVERSIONS
840 DAT%(1) = &H8000   'MEMORY SEGMENT TO HOLD A/D DATA
850 DAT%(2) = 1        'PACER TRIGGER
860 DAT%(3) = 0        'NON-RECYCLE SCANNING
870 FUN% = 6           'FUNCTION 5
880 CALL PCL812(FUN%, SEG DAT%(0), SEG ary1%(0), SEG ARY2%(0),
ER%)
890 IF ER% <> 0 THEN PRINT "PERFORM FUNCTION 5 FAILED!": STOP
900 '
910 '***** STEP 6: CHECK CONVERSION STATUS USING FUNC 8
*****
920 'FOR I = 1 TO 100: NEXT I
930 PRINT "CHECK A/D STATUS (Y/N)? ";
940 A$ = INKEY$: IF A$ = "" GOTO 940
950 PRINT A$
960 IF A$ = "Y" OR A$ = "y" THEN GOTO 970 ELSE GOTO 1050
970 FUN% = 8
980 CALL PCL812(FUN%, SEG DAT%(0), SEG ary1%(0), SEG ARY2%(0),
ER%)
990 PRINT "OPERATION MODE      = "; DAT%(0)
1000 PRINT "STATUS              = "; DAT%(1)
1010 PRINT "CURRENT CONVERSION  = "; DAT%(2)

```

```

1020 PRINT
1030 GOTO 910
1040 '
1050 '***** STEP 7: TRANSFER DATA TO ARRAY USING FUNC 9
*****
1060 '
1070 DAT%(0) = N           'NUMBER OF DATA TO BE MOVED
1080 DAT%(1) = &H8000     'SOURCE MEMORY SEGMENT
1090 DAT%(2) = 0         'STARTING DATA NUMBER
1100 FUN% = 9             'FUNCTION 9
1110 CALL PCL812(FUN%, SEG DAT%(0), SEG ary1%(0), SEG ARY2%(0),
ER%)
1120 IF ER% <> 0 THEN PRINT "PERFORM FUNCTION 9 FAILED!": STOP
1130 '
1140'           *****STEP8:           DISPLAY           DATA
*****
1150 '
1160 'FOR I = 0 TO (N - 1)
1170 ' PRINT I; "A/D RANGE CODE:"; GR%; " CH: "; CHAN%; " DATA: ";
ary1%(I)□
1180 'NEXT I
1190 PRINT
1200 '
1210 '***** STEP 9: DISABLE INTERRUPT ACTIVITY USING FUNC 7
*****
1220 '
1230 FUN% = 7
1240 CALL PCL812(FUN%, SEG DAT%(0), SEG ary1%(0), SEG ARY2%(0),
ER%)
1250 END SUB

```

### **EK - 10. BAŞLANGIÇ VE BİTİŞ ÖRNEKLERİNİ BELİRLEYEN ALTPROGRAM.**

```

REM BASLANGIC VE BITIS ORNEKLERINI BELIRLEYEN ALTPROGRAM
SUB BASBITNOK (SOZCUK() AS INTEGER, ENERJI(), SKORANI(), ILKOR
AS
INTEGER, SONOR AS INTEGER, KOD AS INTEGER)
DIM LAST AS INTEGER, LIMIT AS INTEGER, ZCR AS INTEGER,
MAXZCR AS
INTEGER□
DIM I AS INTEGER, J AS INTEGER
DIM N AS INTEGER, M AS INTEGER, N1 AS INTEGER, N2 AS INTEGER
DIM N1CAP AS INTEGER, N2CAP AS INTEGER
KOD = 0: N = 1: MAXE = 0: MAXZCR = 0
FOR I = 1 TO 14901 STEP 100
ZCR = 0: EINT = 0
LAST = I + 99

```

```

FOR J = I TO LAST
  IF SOZCUK(J) < 0 THEN 100
  EINT = EINT + SOZCUK(J)
  IF SOZCUK(J + 1) < 0 THEN ZCR = ZCR + 1: GOTO 120
100 EINT = EINT - SOZCUK(J)
  IF SOZCUK(J + 1) >= 0 THEN ZCR = ZCR + 1
120 NEXT J
ENERJI(N) = EINT: SKORANI(N) = ZCR
IF EINT > MAXE THEN MAXE = EINT
IF ZCR > MAXZCR THEN MAXZCR = ZCR
N = N + 1
NEXT I
ZCR = 0: SQRZCR = 0: E = 0
FOR I = 131 TO 150
  ZCR = ZCR + SKORANI(I)
  SQRZCR = SQRZCR + SKORANI(I) * SKORANI(I)
  E = E + ENERJI(I)
NEXT I
IZC = ZCR * .05
STD = SQR(SQRZCR * .05 - IZC * IZC)
IZC = IZC + 3 * STD
IZCT = 15
IF IZCT > IZC THEN IZCT = IZC
IMN = E * .05
IF IMN > 4000 THEN
  PRINT "BOSLUGUN UST SINIR ENERJI DEGERI GECILDI,LUTFEN
SOZCUGU TEKRAR GIRINIZ"
  KOD = 1: GOTO 900
END IF
IMX = MAXE
I1 = .03 * (IMX - IMN) + IMN: I2 = 4 * IMN
ITL = I1
IF ITL > I2 THEN ITL = I2: ITU = 5 * ITL
M = 1
360 IF ENERJI(M) >= ITL THEN 380
  M = M + 1: GOTO 360
380 I = M: N1 = M
390 IF ENERJI(I) < ITL THEN 430
  IF ENERJI(I) >= ITU THEN 450
  I = I + 1
  GOTO 390
430 M = M + 1
  GOTO 360
450 IF I = M THEN N1 = N1 - 1
  IF N1 < 1 THEN PRINT "LUTFEN SOZCUGU TEKRAR GIRINIZ": KOD = 1:
GOTO 900
  M = 150
480 IF ENERJI(M) >= ITL THEN 500

```

```

M = M - 1: GOTO 480
500 I = M: N2 = M
510 IF ENERJI(I) < ITL THEN 550
    IF ENERJI(I) >= ITU THEN 570
    I = I - 1
    GOTO 510
550 M = I - 1
    GOTO 480
570 IF I = M THEN N2 = N2 + 1
    REM Buraya kadar, kaba bir sekilde N1 (Baslangic) ve N2 (Bitis) degerleri
    REM belirlendi.
    REM Simdi sifir kesme orani kullanilarak bu degerler daha kesin bir
    REM sekilde belirlenecek.
    LIMIT = 10: I = N1 + 1: N1CAP = N1
600 I = I - 1
    IF I < 3 OR (N1 - I) > LIMIT THEN 690
    IF SKORANI(I) <= IZCT THEN 600
    J = 1: I = I - 1
640 IF SKORANI(I) <= IZCT THEN 670
    J = J + 1: I = I - 1
    IF I >= 3 AND (N1 - I) <= LIMIT THEN 640
670 IF J < 3 THEN 600
    N1CAP = I + 1: GOTO 600
690 LIMIT = 10
    I = N2 - 1: N2CAP = N2
710 I = I + 1
    IF I > 130 OR (I - N2) > LIMIT THEN 800
    IF SKORANI(I) <= IZCT THEN 710
    J = 1: I = I + 1
750 IF SKORANI(I) <= IZCT THEN 780
    J = J + 1: I = I + 1
    IF I <= 130 AND (I - N2) <= LIMIT THEN 750
780 IF J < 3 THEN 710
    N2CAP = I - 1: GOTO 710
800 REM N1CAP=Baslangic araligi,N2CAP=Bitis araligi
    IF (N2CAP - N1CAP) < 9 THEN
        PRINT "SOZCUK COK KISA,LUTFEN TEKRAR GIRINIZ": KOD = 1: GOTO
900
    END IF
    IF (N2CAP - N1CAP) > 69 THEN
        PRINT "SOZCUK COK UZUN,LUTFEN TEKRAR GIRINIZ": KOD = 1:
GOTO 900
    END IF
    ILKOR = N1CAP * 100 - 99
    SONOR = N2CAP * 100 - 99
900 END SUB

```

**EK - 11. ÖNDENGE ALT PROGRAMI.**

```

REM ONDENGE ALTPROGRAMI
SUB ONDENGE (SOZCUK() AS INTEGER, ORNEK AS INTEGER, X())
INDEX% = ORNEK + 199
FIRST = SOZCUK(INDEX%)
INDEX% = INDEX% + 1
SECOND = SOZCUK(INDEX%)
X(201) = SECOND - .95 * FIRST
FOR N% = 202 TO 300
INDEX% = INDEX% + 1
FIRST = SECOND
SECOND = SOZCUK(INDEX%)
X(N%) = SECOND - .95 * FIRST
NEXT N%
END SUB

```

**EK - 12. HAMPENCERE ALTPROGRAMI.**

```

REM HAMPENCERE ALTPROGRAMI
SUB HAMPENCERE (X())
KN% = FREEFILE
N% = 1
OPEN "HAMWIN.DAT" FOR INPUT AS #KN%
DO UNTIL EOF(KN%)
  INPUT #KN%, W
  X(N%) = X(N%) * W
  N% = N% + 1
LOOP
CLOSE #KN%
END SUB

```

**EK - 13. OZILISKI ALTPROGRAMI.**

```

REM OZILISKI ALTPROGRAMI
SUB OZILISKI (X(), R())
DIM M AS INTEGER
RM = 0
FOR M = 1 TO 300
RM = RM + X(M) * X(M)
NEXT M
R(1) = RM
RM = 0
FOR M = 2 TO 98 STEP 2
RM = RM + X(M) * (X(M - 1) + X(M + 1))
NEXT M
R(2) = RM + X(299) * X(300)
RM = 0

```

```

FOR M = 3 TO 295 STEP 4
RM = RM + X(M) * (X(M - 2) + X(M + 2)) + X(M + 1) * (X(M - 1) + X(M + 3))
NEXT M
R(3) = RM + X(297) * X(299) + X(298) * X(300)
RM = 0
FOR M = 4 TO 292 STEP 6
RM = RM + X(M) * (X(M - 3) + X(M + 3)) + X(M + 1) * (X(M - 2) + X(M + 4)) +
X(M + 2) * (X(M - 1) + X(M + 5))
NEXT M
R(4) = RM + X(295) * X(298) + X(296) * X(299) + X(297) * X(300)
RM = 0
FOR M = 5 TO 293 STEP 8
RM = RM + X(M) * (X(M - 4) + X(M + 4)) + X(M + 1) * (X(M - 3) + X(M + 5)) +
X(M + 2) * (X(M - 2) + X(M + 6)) + X(M + 3) * (X(M - 1) + X(M + 7))
NEXT M
R(5) = RM
RM = 0
FOR M = 6 TO 286 STEP 10
RM = RM + X(M) * (X(M - 5) + X(M + 5)) + X(M + 1) * (X(M - 4) + X(M + 6)) +
X(M + 2) * (X(M - 3) + X(M + 7)) + X(M + 3) * (X(M - 2) + X(M + 8)) + X(M +
4) * (X(M - 1) + X(M + 9))
NEXT M
R(6) = RM + X(291) * X(296) + X(292) * X(297) + X(293) * X(298) + X(294) *
X(299) + X(295) * X(300)
RM = 0
FOR M = 7 TO 283 STEP 12
RM = RM + X(M) * (X(M - 6) + X(M + 6)) + X(M + 1) * (X(M - 5) + X(M + 7)) +
X(M + 2) * (X(M - 4) + X(M + 8)) + X(M + 3) * (X(M - 3) + X(M + 9)) + X(M + 4)
* (X(M - 2) + X(M + 10)) + X(M + 5) * (X(M - 1) + X(M + 11))
NEXT M
R(7) = RM + X(289) * X(295) + X(290) * X(296) + X(291) * X(297) + X(292) *
X(298) + X(293) * X(299) + X(294) * X(300)
RM = 0
FOR M = 8 TO 274 STEP 14
RM = RM + X(M) * (X(M - 7) + X(M + 7)) + X(M + 1) * (X(M - 6) + X(M + 8)) +
X(M + 2) * (X(M - 5) + X(M + 9)) + X(M + 3) * (X(M - 4) + X(M + 10)) + X(M +
4) * (X(M - 3) + X(M + 11)) + X(M + 5) * (X(M - 2) + X(M + 12)) + X(M + 6) *
(X(M - 1) + X(M + 13))
NEXT M
R(8) = RM + X(281) * X(288) + X(282) * X(289) + X(283) * X(290) + X(284) *
X(291) + X(285) * X(292) + X(286) * X(293) + X(287) * X(294) + X(288) *
X(295) + X(289) * X(296) + X(290) * X(297) + X(291) * X(298) + X(292) *
X(299) + X(293) * X(300)
RM = 0
FOR M = 9 TO 281 STEP 16
RM = RM + X(M) * (X(M - 8) + X(M + 8)) + X(M + 1) * (X(M - 7) + X(M + 9)) +
X(M + 2) * (X(M - 6) + X(M + 10)) + X(M + 3) * (X(M - 5) + X(M + 11)) + X(M

```

```

+ 4) * (X(M - 4) + X(M + 12)) + X(M + 5) * (X(M - 3) + X(M + 13)) + X(M + 6) *
(X(M - 2) + X(M + 14)) + X(M + 7) * (X(M - 1) + X(M + 15))
NEXT M
R(9) = RM + X(289) * X(297) + X(290) * X(298) + X(291) * X(299) + X(292) *
X(300)
END SUB

```

#### EK - 14. DOK. ALTPROGRAM.

```

REM DOK ALTPROGRAMI
SUB DOK (R(), A(), P(), DOKHATA)
DIM I AS INTEGER, J AS INTEGER, LASTJ AS INTEGER
K = R(2) / R(1)
P(1) = 1
P(2) = K
DOKHATA = (1 - K * K) * R(1)
FOR I = 3 TO 9
  AR = 0
  LASTJ = I - 1
  FOR J = 2 TO LASTJ
    AR = AR + P(J) * R(I - J + 1)
  NEXT J
  K = (R(I) - AR) / DOKHATA
  A(I) = K
  FOR J = 2 TO LASTJ
    A(J) = P(J) - K * P(I - J + 1)
  NEXT J
  DOKHATA = (1 - K * K) * DOKHATA
  FOR J = 2 TO I
    P(J) = A(J)
  NEXT J
NEXT I
REM BU ALTPROGRAM SONUNDA DOK KATSAYILARI P() DIZISINE
YERLESMIS OLACAK
FOR J = 2 TO 9
  P(J) = -P(J)
NEXT J
END SUB

```

#### EK - 15. OZILDOK ALTPROGRAMI.

```

REM OZILDOK ALTPROGRAMI
SUB OZILDOK (P(), ARALIK%, RA())
DIM I AS INTEGER
ROFA = 0
FOR I = 1 TO 9
  ROFA = ROFA + P(I) * P(I)
NEXT I

```

```

RA(ARALIK%, 1) = ROFA
ROFA = 0
FOR I = 2 TO 8 STEP 2
ROFA = ROFA + P(I) * (P(I - 1) + P(I + 1))
NEXT I
RA(ARALIK%, 2) = 2 * ROFA
ROFA = 0
FOR I = 3 TO 4
ROFA = ROFA + P(I) * (P(I - 2) + P(I + 2))
NEXT I
RA(ARALIK%, 3) = 2 * (ROFA + P(6) * P(8) + P(7) * (P(5) + P(9)))
ROFA = 0
FOR I = 4 TO 6
ROFA = ROFA + P(I) * (P(I - 3) + P(I + 3))
NEXT I
RA(ARALIK%, 4) = 2 * ROFA
RA(ARALIK%, 5) = 2 * (P(5) * (P(1) + P(9)) + P(2) * P(6) + P(3) * P(7) + P(4) *
P(8))
RA(ARALIK%, 6) = 2 * (P(1) * P(6) + P(2) * P(7) + P(3) * P(8) + P(4) * P(9))
RA(ARALIK%, 7) = 2 * (P(1) * P(7) + P(2) * P(8) + P(3) * P(9))
RA(ARALIK%, 8) = 2 * (P(1) * P(8) + P(2) * P(9))
RA(ARALIK%, 9) = 2 * (P(1) * P(9))
END SUB

```

#### **EK - 16. HAMMING WINDOW KATSAYILARINI HESAPLAYIP BİR KÜTÜĞE YAZAN PROGRAM.**

```

REM HAMMING WINDOW KATSAYILARINI HESAPLAYIP BİR KUTUĞE
YAZAN PROGRAM
DIM W(300)
KN% = FREEFILE
OPEN "HAMWIN.DAT" FOR OUTPUT AS #KN%
X = 2 * 3.14159 / 299
FOR I% = 1 TO 300
W(I%) = .54 - .46 * COS((I% - 1) * X)
WRITE #KN%, W(I%)
NEXT I%
CLOSE #KN%
END

```

#### **EK - 17. DOGRUSAL BUKME ALTPROGRAMI.**

```

REM DOGRUSAL BUKME ALTPROGRAMI
SUB DOGBUKME (RA(), DOGBUKRA(), ARALIK AS INTEGER)
NORM = (ARALIK - 1) / 34
FOR J% = 1 TO 35
SMM% = INT((J% - 1) * NORM + 1)
S = (J% - 1) * NORM + 1 - SMM%

```

```

FOR I% = 1 TO 9
DOGBUKRA(J%, I%) = (1 - S) * RA(SMM%, I%) + S * RA(SMM% + 1, I%)
NEXT I%
NEXT J%
END SUB

```

### EK - 18. DZB ALTPROGRAMI.

```

REM DZB ALT PROGRAMI
DECLARE SUB FARKLILIK (TEST!(), REFERANS!(), N%, M%, YOL!)
  SUB DZB (TEST(), REFERANS(), MESAFE AS DOUBLE, MH%(), ML%())
    DIM DA(35) AS DOUBLE, DANEXT(35) AS DOUBLE, G(35) AS DOUBLE,
GNEXT(35)AS DOUBLE
    DIM MINDA AS DOUBLE
    SONSUZ = 99999
    G(1) = 1
    CALL FARKLILIK(TEST(), REFERANS(), 1, 1, YOL)
    DA(1) = YOL
    FOR N% = 2 TO 35
    FOR M% = MH%(N%) TO ML%(N%) STEP -1
    MINDA = SONSUZ
    GNEXT(M%) = 1
    LAST% = M% - 2
    FOR J% = M% TO LAST% STEP -1
    IF J% > MH%(N% - 1) OR J% < ML%(N% - 1) THEN
    GOTO 30
    END IF
    IF J% = M% THEN
    GOTO 20
    END IF
    DANEXT(M%) = DA(J%)
    GOTO 40
20 DANEXT(M%) = DA(J%) * G(M%)
    GOTO 40
30 DANEXT(M%) = SONSUZ
40 IF DANEXT(M%) > MINDA THEN
    GOTO 50
    END IF
    MINDA = DANEXT(M%)
    K% = J%
50 NEXT J%
    CALL FARKLILIK(TEST(), REFERANS(), N%, M%, YOL)
    DANEXT(M%) = MINDA + YOL
    IF K% = M% THEN
    GNEXT(M%) = SONSUZ
    END IF
    NEXT M%
    FOR M% = MH%(N%) TO ML%(N%) STEP -1

```

```

DA(M%) = DANEXT(M%)
G(M%) = GNEXT(M%)
NEXT M%
NEXT N%
MESAFE = DA(35)
END SUB

```

### EK - 19. FARKLILIK ALTPROGRAMI.

```

REM FARKLILIK ALT PROGRAMI.
SUB FARKLILIK (TEST(), REFERANS(), N%, M%, YOL)
YOL = 0
FOR K% = 1 TO 9
YOL = YOL + TEST(N%, K%) * REFERANS(M%, K%)
NEXT K%
IF YOL <= 0 THEN
YOL = 0: GOTO 10
END IF
YOL = .4342945 * LOG(YOL)
10 END SUB

```

### EK - 20. KARAR ALTPROGRAMI.

```

REM KARAR ALT PROGRAMI.
DECLARE SUB DZB (TEST!(), REFERANS!(), MESAFE AS DOUBLE, MH%(),
ML%())
SUB KARAR (TEST(), KUTISIMS$, MH%(), ML%())
DIM REFERANS(35, 9), MESAFE AS DOUBLE, MINMES AS DOUBLE
MINMES = 1000
FOR J% = 1 TO 25
KN% = FREEFILE
OPEN KUTISIMS$(J%) + ".DAT" FOR INPUT AS #KN%
INPUT #KN%, A$
FOR K% = 1 TO 35
FOR L% = 1 TO 9
INPUT #KN%, REFERANS(K%, L%)
NEXT L%: NEXT K%
CALL DZB(TEST(), REFERANS(), MESAFE, MH%(), ML%())
IF ABS(MESAFE) < MINMES THEN
MINMES = ABS(MESAFE)
SOZ$ = A$
END IF
CLOSE #KN%
NEXT J%
PRINT
PRINT "MINMES="; MINMES
PRINT
PRINT

```

```
PRINT "          SOZCUK"; "" "; SOZ$; " "; "OLARAK TANINDI"  
PRINT  
PRINT  
100 END SUB
```

### **EK - 21. DEPOLAMA ALTPROGRAMI.**

```
REM DEPOLAMA ALTPROGRAMI  
SUB DEPOLAMA (DOGBUKRA())  
DIM I AS INTEGER, J AS INTEGER  
INPUT "SOZCUK ICIN KUTUK ISMI GIRINIZ"; A$  
KUTNUM = FREEFILE  
OPEN A$ + ".DAT" FOR OUTPUT AS #KUTNUM  
WRITE #KUTNUM, A$  
FOR I = 1 TO 35  
FOR J = 1 TO 9  
WRITE #KUTNUM, DOGBUKRA(I, J)  
NEXT J: NEXT I  
CLOSE #KUTNUM  
END SUB
```