

SES YOLLUNUN MATEMATİK MODELİ

MEHMET BİLGİNER GULMEZOĞLU

DOKTORA TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

1992

Anadolu Üniversitesi

Elektronik Mühendisliği

SES YOLLUNUN
MATEMATİK MODELİ

Mehmet Bilginer Gülmezoglu

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik Elektronik Mühendisliği Anabilim Dalı
Elektronik Bilim Dalında
DOKTORA TEZİ
Olarak Hazırlanmıştır

Danışman: Prof. Dr. Atalay Barkana

Eylül-1992

Mehmet Bilginer GULMEZOGLU'nun DOKTORA TEZİ olarak hazırladığı "Ses Yolunun Matematik Modeli" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Üye: Prof.Dr.Atalay BARKANA

Üye: Prof.Dr.Atila BARKANA

Üye: Doç.Dr.Enis ÇETİN

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun **17 ARALIK 1992**ün
ve **335-1**... sayılı kararıyla onaylanmıştır.

Prof.Dr.Rüstem KAYA
Enstitü Müdürü

ÖZET

Bu çalışmada, sekiz ünlü için çeşitli modellerden elde edilen ses bölgesi alan fonksiyonunun bilgisayar ekranında çizilerek sağır ve dilsizlerin eğitiminde kullanılabilecek algoritmaların geliştirilmesi amaçlanmıştır.

İlk aşamada, 10 bölümden oluşan ses bölgesi için durum_uzay modeli geliştirilmiştir. Elde edilen model lineer olmadığından, çözüm için enine kesit alanlarına verilen başlangıç değerleri önemli rol oynamaktadır. Ayrıca lineer tahmin parametrelerinden bulunan formant frekanslarının değişkenliği de göz önüne alınırsa bu modelin geçerli bir model olmadığına karar verilmiştir. Yine de dört bölümden oluşan ses bölgesi için geliştirilen durum_uzay modelinden sekiz ünlü için enine kesit alanları elde edilmiş ve sonuçlar diğer modeller ile karşılaştırılmıştır.

Ses bölgesinin, girdi ve çıktısı belli olan bilinmeyen bir sistem gibi düşünülerek, sistem belirleme yöntemi ile transfer fonksiyonu yani tüm kutup modeli geliştirildi. Modele girdi olarak, çeşitli genlikte ve çeşitli noktalarda impuls zinciri ile gırtlaktan ölçülen değerler uygulanmıştır.

Ses bölgesinin transfer fonksiyonunun kutup ve sıfırlarını kullanarak enine kesit alanlarının bulunması için kutup-sıfır modeli geliştirildi. Çeşitli kriterler uygulanarak transfer fonksiyonunun optimum dereceleri belirlendi. Bu dereceler için elde edilen kutup ve sıfırlara, formant frekanslarından elde edilen kutup ve sıfırlar yaklaşıncaya kadar ses bölgesinin alan fonksiyonunun katsayıları değiştirilerek sekiz ünlü için enine kesit alanları elde edildi.

Dilin daralma yerini en iyi, kutup-sıfır modelinden elde edilen sonuçlar göstermektedir. Fakat bu değerler, uzun süren iterasyonlar sonucu elde edildiğinden, kullanıcıya hemen aktarılamamaktadır. Ancak değerler dosyalanıp, istenildiği zaman ekranda görülebilir. Enine kesit alanları hemen kullanıcıya iletilmek istenirse tüm-kutup modeli kullanılabilir.

Ses bölgesinin enine kesitinin dairesel olduğu kabul edilerek, elde edilen alan değerlerinden bulunan yarıçapların fonksiyonları çizdirildi. Böylece ses bölgesinin sekiz ünlü için aldığı şekil kafa içinde bilgisayar ekranında görülebilmektedir. Ayrıca, dosyalanmış değerler ile o anda söylenmiş olan ünlüye ait değerler karşılaştırmalı olarak ekranda izlenebilmektedir.

SUMMARY

The purpose of this study is to draw the vocal tract for eight vowels on the computer screen, thus to help in training of the deaf.

In the first stage, the state_space model of the vocal tract which consists of ten sections is obtained. Since the obtained model is not linear, initial values of the cross-sectional areas are very important to obtain a solution which satisfies the constraints. In addition, considering the variability of the formant frequencies obtained from the linear predictive coefficients this model is decided not to be valid. However, the cross-sectional areas for eight vowels are obtained from the state_space model which is developed for a vocal tract composed of four sections and the results are compared with that of the other models.

Assuming the input and output of the vocal tract are known, parameters of the vocal tract transfer function are determined. Vocal tract transfer function is assumed to be an all-pole model. Measured values from the glottis and the impulse train which has various amplitudes and initial points are applied to the model as inputs.

Pole_zero model is developed to obtain the cross-sectional areas by using the poles and zeros of the vocal tract transfer function. Optimal degrees of the pole-zero model is determined by applying various criteria. Obtained poles and zeros for these optimal degrees are compared with the results obtained from the formant frequencies. The cross-sectional areas for eight vowels are obtained by varying the coefficients of area function of the vocal tract until both poles and zeros obtained from different methods are close to each other.

The constriction point of the tongue is best obtained using the pole_zero model. Since iterations take a long time, the cross-sectional areas can not be transferred to the user in a short time. But, these values are stored and shown on the screen whenever it is required. If the transfer of these values to the user is required in a short time, the all-pole model must be used.

Since the shapes of the cross-sections of the vocal tract are assumed as circular, radius of the cross-sectional areas are determined and drawn on the computer screen. Thus, the shape of the vocal tract for eight vowels is shown in human head on the screen. In addition, the values of a vowel which is said at that instant and stored values are shown on the screen together.

TEŞEKKÜR

Bu çalışmanın gerçekleşmesine önemli görüşleri ile katkıda bulunan ve değerli yardımlarını esirgemeyen sayın hocam Prof.Dr.Atalay Barkana'ya ve Prof.Dr.Atila Barkana'ya en içten dileklerle sonsuz teşekkürlerimi sunarım.

Ayrıca, Dğr.Gör.Rıfat Edizkan'a, Arş.Gör.Hakan Tora'ya ve Arş.Gör.Umit Künkçü'ye yardımlarından dolayı teşekkür ederim.

Aynı zamanda tez süresince benden desteğini esirgemeyen ve bütün zorlukları benim ile paylaşan eşim ve oğluma teşekkürü borç bilirim.

İÇİNDEKİLER

	<u>sayfa</u>
ÖZET	iv
SUMMARY	v
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	ix
1. SES VE ÖZELLİKLERİ HAKKINDA GENEL BİLGİ	1
1.1. Giriş	1
1.2. Ses ve İnsan Sesinin Oluşumu	1
1.3. Temel Fiziksel Özellikler	3
1.4. Türkçede Seslerin Sınıflandırılması	3
1.4.1. Ünlüler	4
1.4.2. Ünsüzler	4
1.5. Formant Frekansı ve Ses ile İlişkisi	5
1.6. Nitelikler ve Kısımlar	6
2. ÖNCEKİ ÇALIŞMALAR	7
3. SES BÖLGESİNİN DURUM_UZAY MODELİ	12
3.1. Giriş	12
3.2. Modelin Oluşumu	12
3.3. Model Parametrelerinin Bulunması	14
3.3.1. Kayıpsız durum	14
3.3.2. Kayıplı durum	17
4. SES BÖLGESİNİN BELİRLENMESİ	20
4.1. Giriş	20
4.2. Sistem Belirleme Prosedürü	21
4.2.1. Üç temel esas	21
4.2.2. Modelin geçerliliği	21
4.2.3. Sistem belirleme döngüsü	21
4.3. Sistem Belirleme Yöntemleri	22

İÇİNDEKİLER (devam)

Sayfa

4.4. En Küçük Kareler Yöntemi ile Transfer Fonksiyonunun Belirlenmesi	24
4.4.1. Transfer fonksiyonunun derecesinin belirlenmesi ...	25
4.4.1.1. En küçük kareler yöntemi	25
4.4.1.2. "Signal to residue ratio" yöntemi	25
4.4.1.3. "Akaike information criterion" yöntemi ...	27
4.4.1.4. Optimum derecelerin bulunması	27
4.5. Bilgisayara Ses Verilerinin Alınması	27
4.6. Ses Bölgesinin Enine Kesit Alanlarının Bulunması	28
4.6.1. Kutup katsayılarını kullanarak alanların bulunması	28
4.6.1.1. Transfer fonksiyonunun derecesinin belirlenmesi	28
4.6.1.2. Kullanılacak verilerin sayısını ve cinsini belirleme	29
4.6.1.3. Kutup katsayıları ile yansıma katsayıları arasındaki ilişki	30
4.6.1.4. Yansıma katsayıları ile enine kesit alanları arasındaki ilişki	33
4.6.2. Kutup ve sıfırları kullanarak alanların bulunması .	33
4.6.3. Enine kesit alanlarının bilgisayarda karşılaştırılması çizimi	37
4.6.4. Enine kesit alanlarının insan kafasının içine yerleştirilmesi	37
5. TÜM MODELLER ÜZERİNDE YAPILAN ÇALIŞMALAR VE KARŞILAŞTIRMALAR ..	38
6. SONUÇ VE ÖNERİLER	62
KAYNAKLAR DİZİNİ	64
EKLER.....	66
1. Tavır Nitelikleri.....	67
2. Konum Nitelikleri.....	69
3. Durum Uzay Modelinin Kayıpsız Durumu İçin Programlar.....	70
4. Durum Uzay Modelinin Kayıplı Durumu İçin Programlar.....	82
5. Bilgisayara Ses Verilerinin Alınmasında Kullanılan Devreler...	88
6. Tüm Kutup Modelinden Enine Kesit Alanları Bulan Program.....	91
7. Eşitlik (4.37)'ye Dayanan Kutup Sıfır Modelinden Enine Kesit Alanları Bulan Program.....	95

İÇİNDEKİLER (devam)

8. Eşitlik (4.38)'e Dayanan Kutup Sıfır Modelinden Enine Kesit Alanları Bulan Program.....	115
9. Enine Kesit Alanları İnsan Kafasına Yerleştiren Program.....	138
10.20 Kişiden Alınan 8 Ünlü İçin Elde Edilen Enine Kesit Alanlarının Ortalaması.....	147

ÖZGEÇMİŞ

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
1.1. Ses organlarının şematik gösterimi	2
2.1. "Coker" modelinin parametreleri	8
3.1. Ses bölgesinin bir bölümünün elektriksel eşdeğer devresi	12
3.2. Dört tüpten oluşan ses bölgesinin elektriksel eşdeğer devresi	13
4.1. Belirlenecek sisteme ait blok diagramı	20
4.2. Sistem belirleme döngüsü	22
4.3. Analog ses sinyalinin sayısal olarak elde edilmesi	28
4.4. 256 veri için eşitlik (4.29)'dan elde edilen fft spektrumu ..	31
4.5. 256 veri için eşitlik (4.30)'dan elde edilen fft spektrumu ..	31
4.6. Dudak sinyaline "preemphasis" uygulayarak eşitlik (4.29)'dan elde edilen fft spektrumu	32
4.7. Dudak sinyaline "preemphasis" uygulayarak eşitlik (4.30)'dan elde edilen fft spektrumu	32
4.8. Ses bölgesinin genel gösterimi	34
5.1. "A" harfi için 18 adet LPC katsayısından elde edilen frekans spektrumu ve formant frekansları.....	39
5.2. "A" harfi için 20 adet LPC katsayısından elde edilen frekans spektrumu ve formant frekansları.....	39
5.3. "ü" harfi için 16 adet LPC katsayılarından elde edilen frekans spektrumu ve formant frekansları.....	40
5.4. "ü" harfi için 20 adet LPC katsayılarından elde edilen frekans spektrumu ve formant frekansları.....	40
5.5. Tüm kutup modelinden impals girdi için elde edilen değişimler	43
5.6. Tüm kutup modelinden ölçülen girdi için elde edilen değişimler.....	44
5.7. Kutup sıfır modelinden impals girdi için elde edilen değişimler.....	45
5.8. Kutup sıfır modelinden ölçülen girdi için elde edilen değişimler.....	46
5.9. "A" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması	52
5.10. "E" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması	52
5.11. "U" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması	53
5.12. "ü" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması	53
5.13. "I" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması	54
5.14. "I" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması	54

ŞEKİLLER DİZİNİ (devam)

5.15."O" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması	55
5.16."Ü" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması	55
5.17.İki farklı kişiden alınan "A" harfinin karşılaştırması	56
5.18.İki farklı kişiden alınan "E" harfinin karşılaştırması	56
5.19.İki farklı kişiden alınan "I" harfinin karşılaştırması	57
5.20.İki farklı kişiden alınan "U" harfinin karşılaştırması	57
5.21."A" harfi için ses bölgesinin gösterimi	58
5.22."E" harfi için ses bölgesinin gösterimi	58
5.23."I" harfi için ses bölgesinin gösterimi	59
5.24."U" harfi için ses bölgesinin gösterimi	59
5.25."O" harfi için ses bölgesinin gösterimi	60
5.26."İ" harfi için ses bölgesinin gösterimi	60
5.27."Ü" harfi için ses bölgesinin gösterimi	61
5.28."U" harfi için ses bölgesinin gösterimi	61

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
5.1. "A" harfi için çeşitli modellerden elde edilen enine kesit alanlar	48
5.2. "E" harfi için çeşitli modellerden elde edilen enine kesit alanlar	48
5.3. "I" harfi için çeşitli modellerden elde edilen enine kesit alanlar	49
5.4. "O" harfi için çeşitli modellerden elde edilen enine kesit alanlar	49
5.5. "I" harfi için çeşitli modellerden elde edilen enine kesit alanlar	50
5.6. "U" harfi için çeşitli modellerden elde edilen enine kesit alanlar	50
5.7. "D" harfi için çeşitli modellerden elde edilen enine kesit alanlar	51
5.8. "Ü" harfi için çeşitli modellerden elde edilen enine kesit alanlar	51

BÖLÜM 1

SES VE ÖZELLİKLERİ HAKKINDA GENEL BİLGİ

1.1 Giriş

İlkçağlardan bu yana sürekli olarak araştırma içinde olan insanoğlu, toplumlar büyüyüp geliştikçe, iletişim konusundaki araştırmalarını daha yaygınlaştırmak gereğini hissetmiştir. Ses üzerindeki araştırmalar da bu yüzden ağırlık kazanmıştır.

Ses analizi, tanıma ve üretme yöntemleri, yıllardır birçok araştırmacının ilgilendiği ve son yıllarda bilgisayar teknolojisindeki gelişmelerle (hızlarının ve bilgi hattı bit sayısının artması gibi) daha da önem kazanan bir konudur. Özellikle bilgisayar programlarını kullanarak, matematiksel dönüşümlerden faydalanarak ses analizi yapmak ve bu programlar için sürekli-zaman, kesikli-zaman geçişlerini sağlayan dönüşümleri bulmak gerekmiştir. Bunların bulunmasıyla, matematiksel modeller belirlemiş ve ses yapısının analiz edilebilmesi ve hatta sentezlenmesi sağlanmıştır.

1.2 Ses ve İnsan Sesinin Oluşumu

Nesnelerin titreşmesi, hava moleküllerinin titreşmesine ve ses dalgalarının oluşmasına neden olur. Bu dalgalar, kulağa ulaşıp kulak zarını titreştirdiğinde işitme duyumu meydana gelir.

Ses genel olarak bütün maddesel ortamlarda (katı, sıvı, gaz) yayılır fakat boşlukta yayılmaz. Kulak ancak frekansı 20 Hz ile 20 KHz arasındaki sesleri duyabilir. Ancak insan normalde frekansı 100 Hz ile 4500 Hz arasında olan sesleri rahat üretebilir (Makhoul and Wolf, 1972).

İnsan sesi, gırtlaktaki ses tellerinin titreşmesi ile oluşur. Akciğerler bir miktar havayı sıkıştırarak soluk borusundan ağıza doğru iterler. Bu hava ses tellerini titreştirir ve titreşime uyarak ağıza ve burun boşluklarından geçer. Bu boşluklarda kuvvetlenerek dil, dudak ve dişlerin yardımıyla istenilen şekli alır (Ladefoged, 1962). Sesin oluşumu için kullanılan organlar şunlardır (Ladefoged, 1962):

1. Ciğerler (Enerji kaynağı) : Soluk alıp verme sırasında basınç altında dışarı atılan havadır. Basınç akciğerlerin esnekliği

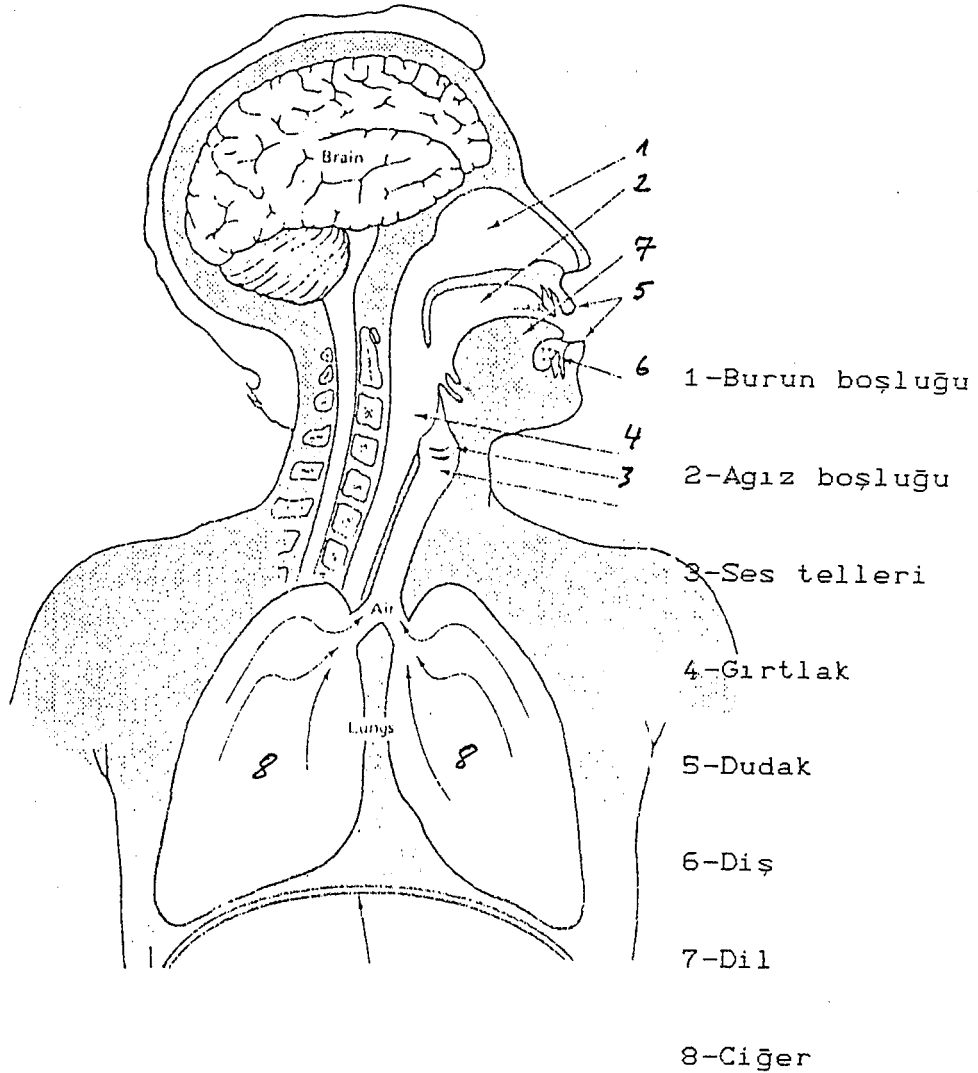
ve solunum kaslarının kasılmasıyla sağlanır.

ii. Ses telleri (Titreşim kaynağı) : İkisi sağda, ikisi solda olmak üzere gırtlakın içinde bulunan ve havanın geçmesiyle ses çıkarmayı sağlayan dört ince kastır.

iii. Ağız veya burun boşluğu (Rezonans kaynağı) : Havanın gırtlaktan geçerken çıkardığı sesi kuvvetlendirir ve sese özel perdesini verir; buna göre ses göğüs sesi (kalın) veya kafa sesi (ince) adını alır.

iv. Ses organları (Dil, yanak, damak, çene, diş, dudak) : Havaya duyduğumuz ses titreşimlerini vermeye yararlar.

Ses organlarının şematik gösterimi şekil 1.1. de verilmiştir.



Şekil 1.1 Ses organlarının şematik gösterimi

1.3 Temel Fiziksel Özellikler

Akustik konuşma dalgası şekil 1.1 de gösterilen yapının serbest fizyolojik hareketinden oluşan akustik basınç dalgasıdır. Seslerin meydana gelişleri sırasında, ciğerlerden dudaklara itilen hava, ses tellerinin fizyolojik ayarlanmasına ve nefes borusunda hava basıncına bağımlı bir oranda açılan ve kapanan ses tellerinin titreşmesine neden olur. Bu ayarlama ses tellerinin uzunluğu, inceliği ve gerilmesindeki değişimleri içerir (Markel and Gray, 1976).

Seslerin genliği, göğüs kafesinin yaptığı basınca ve ses tellerinin buna karşı gösterebileceği dirence bağlıdır. Seslerin frekansı, ses tellerinin boyutlarına ve gerilimine, rezonans boşluklarının büyüklüğüne bağlıdır. Çocuk ve kadınlarda ses telleri ve rezonans boşlukları küçük olduğu için sesler ince çıkar, erkeklerde ise ses telleri ve rezonans boşlukları büyük olduğu için sesler kalın çıkar. Bir kimsede rezonans boşlukları, çıkartılmak istenen sese göre ayarlanır; kalın ses çıkartılacaksa gırtlak ve yutak aşağı iner, boşluklar büyür; ince ses çıkartılacaksa gırtlak ve yutak yukarı kalkar (Aktop, 1988).

Bir sesin oluşmasında üç eylem bulunur (Aktop, 1988):

1. Gerilme : Konuşma organları istenen sesi çıkarma durumuna geçerler.
2. Duralama : Çok kısa bir süre gerilme durumunda kalıp istenilen ses çıkartılır.
3. Çözülme : Organlar görevlerini bitirdikten sonra gevşeyerek bekleme durumuna geçerler.

Ses yolu, nefes borusunun ağızdan dudaklara uzanan ve zamanın fonksiyonu olarak değişen, düzgün olmayan bir akustik tüptür. Zamana bağımlı değişmeye yol açan büyük anatomik bileşenler dudaklar, çene, dil ve (yumuşak) damaktır. Örneğin, açılan dudanın alanı, dudaklar kapalı iken 0 cm²'den dudaklar açık iken 20 cm² ye kadar değişir. Genizden gelmeyen seslerin oluşu esnasında damak geniz boşluğundan ses yolunu kapatır. Geniz boşluğu genizden gelen seslerin üretiminde kullanılan ses iletimi için akustik bir tüptür (Markel and Gray, 1976).

1.4. Türkçede Seslerin Sınıflandırılması :

Türkçedeki sesler dilbilgisi açısından iki bölümde incelenir.

- i- Ünlüler
- ii- Ünsüzler

1.4.1. Ünlüler

Ses yolunda önemli bir engelle karşılaşmadan, yalnızca ses tellerinin titreşmesiyle oluşur. Bu nedenle yüksek genlikli ve periyodiktirler (Makhoul and Wolf, 1972).

Türkçedeki ünlüler a,e,ı,i,o,ö,ü,ü olmak üzere sekiz tanedir. Ünlüler hem kendileri ses verir, hemde ünsüzleri seslendirirler. Dilin, alt çenenin ve dudakların aldıkları duruma göre üç gruba ayrılırlar.

- a) Dilin aldığı duruma göre kalın veya ince ünlüler
- b) Alt çenenin aldığı duruma göre geniş veya dar ünlüler
- c) Dudakların aldığı duruma göre yuvarlak veya düz ünlüler.

	Düz		Yuvarlak	
	Geniş	Dar	Geniş	Dar
Kalın	a	ı	o	u
İnce	e	i	ö	ü

1.4.2. Ünsüzler

Ünsüzlerin söylenişinde ciğerden gelen hava akışına, ses borusunda veya ses organlarında çeşitli engellemeler yapılır. Ünlülere göre ünsüzlerin söylenme süresi çok daha kısa ve genlikleri çok düşüktür (patlamalı ünsüzler hariç). Ünsüzler, dudakların, dişlerin ve dilin durumuna göre oluşurlar. Ünlülerden önce veya sonra gelerek genlik ve frekanslarına etki ederler (Makhoul and Wolf, 1972).

Ünsüzler kendi aralarında oluşumlarına göre üç gruba ayrılır (Yannakoudakis and Hutton, 1987):

A) Çıkaklarına (sesin ağızdan çıkarken biçimlendiği yer) göre :

- a) Dudak ünsüzleri : b,f,m,p,v.
- b) Diş ünsüzleri : c,ç,d,j,l,n,r,s,ş,t,z.
- c) Damak ünsüzleri : g,ğ,k,y.
- d) Gırtlak ünsüzleri : h

B) Ses Yolunun Tıkalı veya Aralık Oluşuna Göre :

- a) Havanın, tamamen tıkalı olan ses yolunun bir noktasını itip açmasıyla biçimlenen süreksiz ünlüler : b,c,ç,d,g,k,p,t.

b) Havanın daralmış olan ses yolunun bir noktasından sürtünüp çıkmasıyla biçimlenen ünsüzler : f,ğ,h,j,l,m,n,r,s,ş,v, y,z.

c) Ses tellerinin titreşimine göre :

i) Yumuşak ünsüzler : Çıkaklarında (sesin ağızdan çıkarken biçimlendiği yer) biçimlenirken ses tellerininide titreştiren ünsüzler : b,c,d,g,ğ,j,l,m,n,r,v,y,z.

ii) Sert ünsüzler : Çıkaklarında biçimlenirken ses tellerini titreştirmeyen, gürültü veya fısıltı halinde çıkan ünsüzler : ç,f,h,k,s,ş,t,p.

İnsan dili bir sesi söylerken hemen ikinci sese ani geçiş yapamaz. Örneğin, bir ünlü ile bir ünsüz söylenecekse ünlüye göre ünsüzün söyleniş biçimi değişmektedir. Bu durumda ünsüz harfleri tek başına değil de beraber söylendikleri ünlü ile birlikte incelemek daha uygundur. Benzer şekilde ünlü de ünsüzden etkilenmektedir.

1.5. Formant Frekansı ve Ses ile İlişkisi

Akciğerlerden gelen hava ses tellerinden "puls" lar halinde çıkar. Bu "puls" lar T_0 zaman aralıklarıyla tekrar eder ve $F_0=1/T_0$ sesin "pitch" frekansını verir. "Pitch" frekansı, ses sinyalinin frekans spektrumundaki iki temel frekanstan biridir ve spektrum şeklini belirler. Temel frekans olmasına rağmen kişiye göre değişir ve ünlüler arasında ayırdedici bir özelliği yoktur (Fry, 1979).

Ses sinyalinin frekans spektrumundaki ikinci temel frekans formant frekansıdır. Formant frekansları, sesin frekans spektrumundaki en büyük genlikli değerlerdir ve en büyük spektral enerjiye sahiptirler. Genelde bir sesin yapısında ondan fazla format frekans bulunabilir, fakat bunlardan en etkili olanları ilk üçüdür (Fry, 1979).

Bir sesin fonetik kalitesi, ilk adımda formant frekansları ile şekillenen spektrumdaki enerji dağılımına bağlıdır. Bu frekanslar, ses üretimi esnasında, ses yolunun rezonans frekanslarına karşılık gelen frekanslar olarak tanımlanır (Fallside and Woods, 1985) Formant frekansları, ses tellerinin boyutlarına ve gerilimine, ses yolunun uzunluğuna ve rezonans boşluklarının büyüklüğüne bağlıdır (Fry, 1979).

Farklı kişiler tarafından söylenen bir ünlünün formant frekansları benzerdir. Aynı zamanda farklı ünlülerin formant frekansları da farklıdır. Bu yüzden formant frekansları ünlüler için ayırdedici bir özelliktir. Bir kadın tarafından üretilen ünlü, bir erkek tarafından üretilen aynı ünlüden yaklaşık % 15 daha yüksek

formant frekansına sahiptir. Bu da sekiz yaşında bir çocuğun ürettiği aynı ünlüden % 15 daha düşük bir formant frekansıdır. Ayrıca bir kişinin formant frekansının ortalama değeriyle, kişinin ses yolunun tüm uzunluğu ters orantılıdır (Fallside and Woods, 1985).

Ses dalgası, bir kaynak ile ses yolu filtresinin ya da ses yolu transfer fonksiyonunun çarpımından ibarettir. Kaynak, ses tellerinin periyodik açılıp kapanma hareketini sağlayan dışarı atılan havanın modülasyonudur. Ses yolu transfer fonksiyonu yalnızca ses yolu boşluklarının şekline bağlıdır. Kaynak ve transfer fonksiyonu birbirinden bağımsızdır. Sesin spektrumunun formantları ses yolu transfer fonksiyonunun maksimumlarından meydana gelirler (Fallside and Woods, 1985).

Formant frekansları, sesin çıkış anındaki ağız yapısına bağlıdır. Ses çıkışında genelde iki değişken etkili olmaktadır. Bunlardan biri dil ile üst damak arası boşluk, ikincisi ise gırtlaktaki ses telleri arasındaki boşluktur (Fallside and Woods, 1985).

1.6. Nitelikler ve Kısımlar

Ayırt edici nitelikler kavramı ses analizinde güçlü bir faktördür. Niteliklerin tanımı araştırmacıların özel ilgisine göre değişebilir. İdeal olarak bu tanımlar, ses üretimi, ses dalgası, ses idraki safhalarında ve en az lisan birimlerindeki mesaj düzeyinde eşit olarak iyi iş görürler.

Bu tanımların asıl amacı, ses üretimindeki telaffuz hareketi yerlerini sınıflandırmak ve üretim bölümünden ses dalgası karakteristiklerine geçiş için kuralları özetlemektir.

Nitelikler iki kısım halinde toplanabilir. Tavır nitelikleri (manner features) ve yer nitelikleri (place features) (Fallside and Woods, 1985). Tavır nitelikleri Ek-1'de ve konum nitelikleri Ek-2'de tablo halinde verilmiştir.

BÖLÜM 2

ÖNCEKİ ÇALIŞMALAR

Chiba ve Kajiyama (1941), ses üretimi için "dalga yayılması" teorisini geliştirmişlerdir. Akustik dalganın ağız ve burun boşluğunda tek boyutlu olarak yayılması ile sesin üretilebileceğini göstermişlerdir. Akış hacmi ve basıncın sürekli dalgalarının bir kaç rezonans modu için ses bölgesinde mevcut olduğu iddia edilmektedir (Carre and Mrayati, 1989).

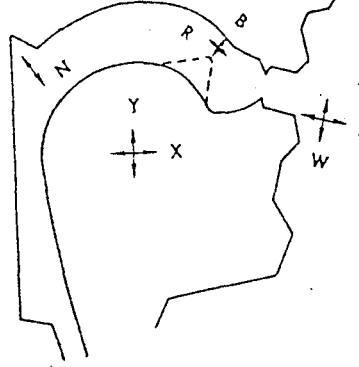
Stevens ve House (1955), ses üretimi için "üç parametre" modelini geliştirmişlerdir. Bu modelde, model parametrelerindeki değişime karşı üç formant frekansındaki değişimleri hesaplayan ses bölgesinin elektriksel eşdeğer devresi kullanılmıştır. Akustik parametrelerin (F_1 , F_2 , F_3) telaffuza ait parametreler (daralmanın yeri, daralmanın derecesi ve dudak) ile olan ilişkilerini gösteren birkaç nomogram verilmiştir (Carre and Mrayati, 1989).

Fant (1960), ses bölgesini temsil eden ve dört boşluktan oluşan çok eğitici bir modeli ayrıntılı bir şekilde incelemiştir. Beş formant frekansın bu dört boşluğun boyutları ile ilişkilerini veren mükemmel nomogramlar verilmiştir. Bu nomogramlar daralmanın yeri ve enine kesit alanındaki değişimlere göre formant frekansındaki değişimleri yansıtmaktadır.

Yine Fant (1960), dilin boynuza benzetilmesine dayanan başka bir modeli geliştirmiştir. Bu modeldeki üç parametre şunlardır : a) dilin yarattığı minimum daralmanın yeri, b) bu koordinatdaki enine kesit alan, c) dudak bölümünün uzunluğunun enine kesit alanına oranı (L_0/A_0)

Mermelstein (1967), ses bölgesinin uyarım sinyalini modüle etmeye yaradığını ve böylece üretilen sinyale, dile ait karakter verildiğini belirlemiştir.

Coker (1968), ses bölgesinin enine kesit alan fonksiyonunun yedi parametre cinsinden tanımlandığı bir model geliştirmiştir. Bu parametreler : dil gövdesinin pozisyonu (X, Y), dudak çıkıntısı (L), dudak yuvarlaması (W), dil ucunun yarattığı daralmanın yeri ve derecesi (R, B) ve yumuşak damağa bağlantı derecesi (N). Bu parametrelerin ağız boşluğundaki gösterimi şekil 2.1'de verilmiştir.



Şekil 2.1. "Coker" modelinin parametreleri (Carre and Mrayati, 1989).

Stevens (1972), tarafından sunulan, sesin "quantal" teorisi, ses üretiminin teorik formülasyonunda önemli bir başarı olarak değerlendirilmiştir. Stevens, telaffuza ait parametrelerin akustik parametreler ile ilişkilerinin "quantal" olduğunu ve bunun, telaffuza ait parametreler belli bir aralıkta değişirken akustik şablonun bir durumdan diğerine değiştiği anlamına geldiğini kabul etmiştir (Carre and Mrayati, 1989).

Mermelstein (1973), çene, dil gövdesi, dil ucu, dudaklar, yumuşak damak ve dil kemiğinin pozisyonunu belirleyen değişkenler vasıtasıyla ses bölgesi taslağını temsil eden başka bir model tasarlamıştır (Carre and Mrayati, 1989).

Wakita (1973), konuşmalardaki formantların ve enine kesit alan fonksiyonlarının dinamik değişimlerinin elde edilebileceğini göstermiştir. Ters filtre modelinin filtreleme işlemi ile sesin akustik tüp modeli analiz edilmiş ve bunların, yansıma katsayılarının aynı olduğu gösterilmiştir. Böylece ses sinyalinin ters filtreleme işlemi ile yansıma katsayılarının bulunabileceği sonucuna varılmıştır. Wakita'ya göre, kesikli alan fonksiyonu, yansıma katsayılarından kolaylıkla elde edilebilir.

Lindblom ve Sundberg (1976), altı parametreden türetilen telaffuz ile ilgili bir model geliştirmişlerdir. Bu parametreler : dudak yüksekliği, dudak genişliği, çene, dil gövdesi ve gırtlak yüksekliğidir. Bu model, dil yüksekliğinin klasik ölçülere uygunluğunu yeniden

yorumlamak için kullanılmıştır (Carre and Mrayati, 1989).

Ladefoged, Lindau ve Maeda (1979), istatistiki analizlerin sonuçlarına dayanan ve dilin, telaffuza ait bir modelini incelemişlerdir. Bir ses olayının tanımı için dilin, ya iki bağımsız parametrenin bir bileşimi olarak yada çene ve dil gövdesi, dil ucu ve dil arkası gibi bağımsız olarak kontrol edilebilen sistemlerin bir bileşimi olarak dikkate alınabileceğini savunmuşlardır.

Maeda (1979), ses olayının tanımı için dilin; çene ve dil gövdesi, ağız boşluğunun ön ve arka tarafı gibi bağımsız olarak kontrol edilebilen mekanik sistemlerin bir bileşimi olarak dikkate alınabileceğini ortaya atmıştır. Bu gibi telaffuza yarayan organları temsil eden lineer elemanların, yanal X-ışını filmlerinden belirlenen ses bölgesi şekillerinin istatistiki bir analizi ile bulunabileceği savunulmuştur. Dört elemanın toplamının, gözlenen dil şekillerini gayet iyi tanımladığı ve lineer gösterimin telaffuz ile ilgili bir model olarak dikkate alınabileceği belirtilmektedir.

Maeda (1982), sabit bir hava basıncı kaynağından, gırtlığı temsilen zamanla değişen dar bir bölümden ve burun boşluğu ile bağlantılı ses bölgesine uygun bir tüpten oluşan sistemin ses bölgesi modeli olarak kabul edilebileceğini savunmuştur.

Morikawa ve Fujisaki (1984), ses sinyallerinin analizi için yeni bir yöntem geliştirmişlerdir. Bu yöntem, kendiliğinden geriye doğru ilerleyen ortalama (autoregressive moving average = ARMA) işlemi ile ilgili bir gösterime dayanmaktadır.

Maeda (1985), bitiş sağlayan ünsüzlerin üretilmesi üzerinde çalışmıştır. Bitirici ünsüzlerin salıverilmesindeki patlama sinyallerinin, ses bölgesinin geçici olarak uyarılması ve bir gürültü kaynağı ile üretildiği belirlenmiştir. Bu çalışma, sadece sürtme sesi çıkartılırken yaratılan gürültünün incelenmesi ile sınırlanmaktadır.

Miyanaga, Miki ve Nagai (1986), ses sinyallerindeki zamanla değişen "ARMA" parametrelerini tahmin etmek için "Kalman" filtre algoritmasının genişletilmiş şekli olan bir algoritma geliştirmişlerdir.

Maeda (1987), ünsüz-ünlü heceleri için telaffuz ile ilgili bir modelin, ses bölgesinin enine kesit alan fonksiyonu cinsinden formül şeklinde ifade edilebileceğini göstermiştir. Burada ünsüzlerin, p,t veya k, ünlülerin de i,a veya u olduğu kabul edilmiştir. Ünsüzlerin,

patlamalılardaki iki spektral niteliğin var veya yok olmasına bağlı olarak oldukça iyi tahmin edilebileceği vurgulanmaktadır.

Carre ve Mrayati (1989), gırtlak ve dudak arasındaki ses bölgesi boyunca, ayırıcı bölümlerin varlığını gösteren yeni bir kavram geliştirmişlerdir. Bu kavramın aynı zamanda, bir ses bölgesindeki büzülmede daralmanın derecesi ile ilgili üç ayırdedici modeli tanımladığı iddia edilmektedir.

Perkell (1989), telaffuza ait değişik verilerin, teorik sonuçların anlaşılmasına yardımcı olduğunu göstermiştir. Telaffuza ait hareketlerin programlanmasının; i) telaffuza ait amaçların sırasının ve yapısının, ii) ses bölgesinin herhangi bir andaki durumunun ve iii) ayrı sözlerin akustik koşullarının, fonksiyonu olduğu belirtilmiştir. Telaffuza ait veriler ile ilgili üç örnek sunulmuştur. Bunların her birinin yukarıda bahsedilen fikirler ile ilgili bazı hususları açıkladığı ve geniş ölçüde değişkenlik ihtiva ettiği iddia edilmektedir. Verilerin bir kümesinin telaffuza ait amaçların tanımı için bir mekanizma fikri verdiği; ikinci kümesinin, ortak telaffuz şeklinin stratejisi üzerindeki muhtemel etkenleri ortaya çıkardığı ve üçüncü kümenin, telaffuza ait verilerdeki değişkenliğin ilginç kurallarını gözden saklayabileceği vurgulanmıştır.

H.M. Teager ve S.M. Teager (1990), yaptıkları deneylerin sonuçlarına dayanarak, ses bölgesindeki momentum dalgalarının pasif ve akustik olmadığı sonucuna varmışlardır. Ses bölgesindeki hava akışının ölçülmesi, akustik empedansın veya basınç-akış oranının bozulduğunu göstermektedir. Ses bölgesinin herhangi bir enine kesitindeki basıncın, sabit olduğu ve aynı enine kesitteki son derece farklı hava akışlarından değişik bir özellik göstermediği iddia edilmektedir.

Lin ve Fant (1990), telaffuza ait konuşma sentezi için yeni bir algoritma geliştirmişlerdir. Algoritma iki ana bölümden oluşmaktadır: i) ses bölgesinin, frekans düzleminde ayrıntılı bir şekilde modellenmesi, ii) frekans düzleminde zaman düzlemine geçişi sağlayan bir veri kümesi. Ses bölgesi akustiklerini simüle etmek için bilgisayar modeli geliştirilmiştir. Model, ses sisteminin bütün bilinen önemli elemanlarını içermektedir ve akustik kaynak ile dudak/burun çıktısı arasındaki transfer fonksiyonunu hesaplamaktadır.

Göttigen Üniversitesinden doktorlar, mühendisler, fizikçiler, dil uzmanları ve konuşma terapistleri ile işbirliği içinde olan Carstens Medizinelektronik şirketi (1991), ilerlemiş mikroelektronik bilimini kullanarak dil hareketlerinin gözlemlenebilmesi için bir sistem

geliřtirmiřtir. Sistem, endüktif uzaklık ölçüm prensibine dayanmaktadır. Bu ölçümün, biyolojik sistemler üzerinde bir etkisi olmadığı ve çalışılan organ için tamamen zararsız olduğu iddia edilmektedir. Gerçek zamanda telaffuza ait beş elemanın (dil ucu ve tabanı, dudaklar, alt çene, yumuşak damak) aynı anda gözlemlenmesine olanak verilmektedir.

BÖLÜM 3

SES BÖLGESİNİN DURUM UZAY MODELİ

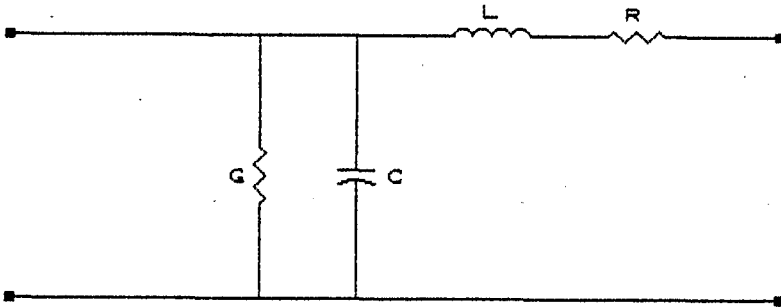
3.1 Giriş

Ses bölgesinin tüplerden oluşan gösteriminde tüplerin ilk önce kayıpsız olduğu kabul edilmiş; daha sonra kayıpların olduğu yani gerçek tüplerden oluşan gösterim incelenmiştir. Ses bölgesinin eşdeğer devresinde gırtlak empedansı sonsuz; ısıma empedansı sıfır kabul edilmiştir. Durum_uzay modelinin elde edilmesinde doğal durum değişkenleri kullanılmıştır.

Ses bölgesinin 10 tüpten oluştuğu ve 17 cm uzunluğa sahip olduğu kabul edilmiştir. Fakat 10 tüpten oluşan ses bölgesinin durum_uzay modeli çok uzun ve büyük ifadelerle sahip olduğundan ve istenilen sayıdaki tüplerden oluşan ses bölgesi modellerine geçiş kolay olduğundan, bu bölümde sadece 4 tüp için model elde edilmiş ve 10 tübe geçiş şekli anlatılmıştır. 10 tüp için elde edilen denklem seti lineer olmadığından ve çözüm vektörü enine kesit alanlarının başlangıç değerlerine bağlı olduğundan sonuç alınamamış, analizler 4 tüp için yapılmıştır.

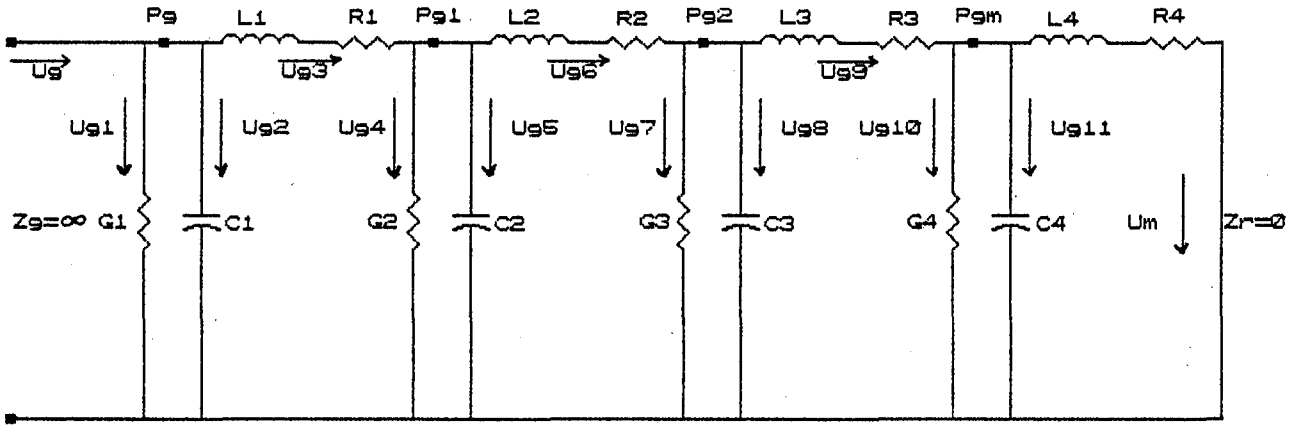
3.2 Modelin Oluşumu

Ses bölgesinin bir bölümü için eşdeğer devre şekil 3.1'de gösterilmiştir (Fant, 1970).



Şekil 3.1 Ses bölgesinin bir bölümünün elektriksel eşdeğer devresi.

Dört tüpten oluşan ses bölgesinin elektriksel eşdeğer devresi şekil 3.2'de verilmiştir.



Şekil 3.2 Dört tüpten oluşan ses bölgesinin elektriksel eşdeğer devresi

Şekil 3.2'de

P: Ses basıncı

U: Akustik hacim hızı

Z: Empedans

R: Rezistans

G: Kondüktans

L: Endüktans

C: Kondansatör

Z_g: Gırtlak empedansı

Z_r: Dudak yayılma empedansıdır.

Yukarıdaki devrede P_g, U_{g3}, P_{g1}, U_{g6}, P_{g2}, U_{g9}, P_{gm} ve U_m durum değişkenleri olarak seçilmiştir. Gerekli işlemler yapıldıktan sonra devrenin durum_uzay gösterimi,

$$\begin{array}{l}
 \text{d} \\
 \text{dt}
 \end{array}
 \begin{bmatrix}
 P_g \\
 U_{g3} \\
 P_{g1} \\
 U_{g6} \\
 P_{g2} \\
 U_{g9} \\
 P_{gm} \\
 U_m
 \end{bmatrix}
 =
 \begin{bmatrix}
 -G_1/C_1 & -1/C_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1/L_1 & -R_1/L_1 & -1/L_1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1/C_2 & -G_2/C_2 & -1/C_2 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1/L_2 & -R_2/L_2 & -1/L_2 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1/C_3 & -G_3/C_3 & -1/C_3 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1/L_3 & -R_3/L_3 & -1/L_3 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1/C_4 & -G_4/C_4 & -1/C_4 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1/L_4 & -R_4/L_4
 \end{bmatrix}
 \begin{bmatrix}
 P_g \\
 U_{g3} \\
 P_{g1} \\
 U_{g6} \\
 P_{g2} \\
 U_{g9} \\
 P_{gm} \\
 U_m
 \end{bmatrix}
 +
 \begin{bmatrix}
 1/C_1 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{bmatrix}
 \quad U_g(3.1)$$

olarak ifade edilebilir. Yukarıdaki eşitlik genel olarak,

$$\dot{X} = A X + B U \quad (3.2)$$

şeklinde yazılabilir. Burada,

X:Durum vektörü

U:Giriş vektörü

A:Durum matrisi

B:Giriş matrisi (vektörü) dir.

Durum matrisindeki rezistanslar, kondüktanslar, enduktanslar ve kondansatörler aşağıdaki gibi ifade edilebilir (Flanagan, 1972).

$$R = \frac{S l}{A^2} \sqrt{w \rho \mu / 2} \quad L = \rho l / A$$

$$G = S \frac{(\eta - 1) l}{\rho c^2} \sqrt{\lambda w / 2 C_p} \quad C = A l / \rho c^2 \quad (3.3)$$

Burada,

l:Tüp uzunluğu(cm)

A:Tüp enine kesit alanı(cm²)

S:Tüp çevresi(cm)

ρ :Hava yoğunluğu=1.14 10⁻³ gr/cm³

c:Ses hızı=3.5 10⁴ cm/sn

μ :Yapışkanlık katsayısı=1.86 10⁻⁴ din.sn/cm²

w:Rezonans frekansı(rad/sec)

η :Adiabatik sabiti=1.4

λ :Isı iletim katsayısı=0.055 10⁻³ kal/cm.sn.derece(0°C)

C_p:Sabit basınçta havanın özgül ısısı=0.24 kal/gr.derece(0°C,1atm)

dir.

3.3 Model Parametrelerinin Bulunması

3.3.1 Kayıpsız durum

ilk olarak tüplerin kayıpsız olduğu varsayılmıştır. Bu durumda

eşitlik (3.1)'de verilen durum_uzay gösterimindeki G ve R elemanları sıfır olmaktadır. Böylece kayıpsız durum için devrenin durum_uzay gösterimi,

$$d \frac{d}{dt} \begin{bmatrix} P_g \\ U_{g3} \\ P_{g1} \\ U_{g6} \\ P_{g2} \\ U_{g9} \\ P_{gm} \\ U_m \end{bmatrix} = \begin{bmatrix} 0 & -1/C_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/L_1 & 0 & -1/L_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/C_2 & 0 & -1/C_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/L_2 & 0 & -1/L_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/C_3 & 0 & -1/C_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/L_3 & 0 & -1/L_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/C_4 & 0 & -1/C_4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/L_4 & 0 \end{bmatrix} \begin{bmatrix} P_g \\ U_{g3} \\ P_{g1} \\ U_{g6} \\ P_{g2} \\ U_{g9} \\ P_{gm} \\ U_m \end{bmatrix} + \begin{bmatrix} 1/C_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3.4)$$

olarak ifade edilebilir.

Eşitlik (3.4)'deki A matrisinin karakteristik denkleminin kökleri sanal eşleniktir. Dolayısıyla karakteristik denklem,

$$|\lambda I - A| = (\lambda^2 + \lambda_1^2)(\lambda^2 + \lambda_3^2)(\lambda^2 + \lambda_5^2)(\lambda^2 + \lambda_7^2) \quad (3.5)$$

olarak yazılabilir. Burada, λ_i ($i=1,3,5,7$) karakteristik denklemin köklerinin mutlak değeridir. Bu değerler aynı zamanda, yalnız kutuplara sahip bir transfer fonksiyonunun köklerine eşittir ve sesin formant frekanslarını vermektedir.

A matrisi, eşitlik (3.5)'de yerine konarak gerekli işlemler yapılarak, aynı dereceden λ 'lara sahip katsayılar birbirine eşitlenirse aşağıdaki denklemler elde edilebilir.

$$\begin{aligned} & \frac{c^2}{l_4^2} + \frac{A_3^2 c^2}{l_3 l_4 A_4} + \frac{c^2}{l_3^2} + \frac{A_2^2 c^2}{l_2 l_3 A_3} + \frac{c^2}{l_2^2} + \frac{A_1^2 c^2}{l_1 l_2 A_2} + \frac{c^2}{l_1^2} = \lambda_1^2 + \lambda_3^2 + \lambda_5^2 + \lambda_7^2 \quad (3.6) \\ & \frac{c^4}{l_3^2 l_4^2} + \frac{c^4 A_2}{l_2 l_3 l_4^2 A_3} + \frac{c^4 A_2}{l_2 l_3^2 l_4 A_4} + \frac{c^4}{l_2^2 l_4^2} + \frac{c^4 A_3}{l_2^2 l_3 l_4 A_4} + \frac{c^4}{l_2^2 l_3^2} + \\ & \frac{c^4 A_1}{l_1 l_2 l_4^2 A_2} + \frac{c^4 A_1 A_3}{l_1 l_2 l_3 l_4 A_2 A_4} + \frac{c^4 A_1}{l_1 l_2 l_3^2 A_2} + \frac{c^4 A_1}{l_1 l_2^2 l_3 A_3} + \frac{c^4}{l_1^2 l_4^2} + \frac{c^4 A_3}{l_1^2 l_3 l_4 A_4} + \end{aligned}$$

$$\frac{c^4}{l_1^2 l_3^2} + \frac{c^4 A_2}{l_1^2 l_2 l_3 A_3} + \frac{c^4}{l_1^2 l_2^2} = \lambda_1^2 \lambda_3^2 + \lambda_5^2 \lambda_7^2 + (\lambda_1^2 + \lambda_3^2)(\lambda_5^2 + \lambda_7^2) \quad (3.7)$$

$$\begin{aligned} & \frac{c^6}{l_2^2 l_3^2 l_4^2} + \frac{c^6 A_1}{l_1 l_2 l_3^2 l_4^2 A_2} + \frac{c^6 A_1}{l_1 l_2^2 l_3 l_4^2 A_3} + \frac{c^6 A_1}{l_1 l_2^2 l_3^2 l_4 A_4} + \frac{c^6}{l_1^2 l_3^2 l_4^2} + \\ & \frac{c^6 A_2}{l_1^2 l_2 l_3 l_4^2 A_3} + \frac{c^6 A_2}{l_1^2 l_2 l_3^2 l_4 A_4} + \frac{c^6}{l_1^2 l_2^2 l_4^2} + \frac{c^6 A_3}{l_1^2 l_2^2 l_3 l_4 A_4} + \frac{c^6}{l_1^2 l_2^2 l_3^2} = \\ & \lambda_1^2 \lambda_3^2 (\lambda_5^2 + \lambda_7^2) + \lambda_5^2 \lambda_7^2 (\lambda_1^2 + \lambda_3^2) \end{aligned} \quad (3.8)$$

$$\frac{c^8}{l_1^2 l_2^2 l_3^2 l_4^2} = \lambda_1^2 \lambda_3^2 \lambda_5^2 \lambda_7^2 \quad (3.9)$$

Eşitlikler (3.6-3.9), L ve C elemanlarının eşitlik (3.3)'de verilen karşılıklarının A matrisinde yerine konması ve gerekli basitleştirmelerin yapılması ile elde edilmiştir. Eşitliklerin sağ tarafındaki formant frekansları çeşitli yöntemler ile elde edilmektedir. Bu yöntemlerden biri, ses verilerinden elde edilen "Fast Fourier Transform (FFT)" spektrumlarını incelemektir (Papamichalis, 1987). Spektrumlardaki tepe noktaları formant frekanslarını vermektedir. Fakat bu spektrumlarda "pitch" frekansı karışıklıklara neden olduğu için FFT spektrumlarının düzgünleştirilmesi gerekir. İkinci yöntem ise, ses verilerinden lineer tahmin parametrelerini (LPC) bularak oluşturulan transfer fonksiyonunun genliğinin frekans spektrumunu elde etmektir (Markel and Gray, 1976). Bu şekilde elde edilen frekans spektrumları, FFT spektrumlarına göre daha düzgün olduğundan formant frekansları hakkında daha doğru bilgi vermektedir. Dolayısıyla durum_uzay modelinden elde edilen denklemlerin çözümünde kullanılan λ 'lar ikinci yöntemden elde edilmiştir. Dolayısıyla λ 'lar bilinmektedir. Ses bölgesini oluşturan her bir bölümün (tübün) uzunluğunun eşit olduğu kabul edilirse l'lerde bilinmektedir. Çünkü ses bölgesinin uzunluğu 17 cm kabul edilmiştir, fakat bu değer istenildiğinde programlarda değiştirilebilir. Böylece eşitlikler (3.6-3.9), dört bilinmeyenli (A_i 'ler) dört denklem şekline dönüşmüş olur. Bu denklem seti çözüldüğünde ses bölgesinin enine kesit alanları bulunabilir. Bu durum ile ilgili programlar Ek-3'de verilmiştir.

On tüpten oluşan ses bölgesi için de yukarıda sıralanan işlemler yapılmış ve elde edilen denklem seti (dokuz bilinmeyenli dokuz denklem), çözücü programlar ile birlikte Ek-3'de verilmiştir.

3.3.2 Kayıplı durum

Bu durumda eşitlik (3.1)'de verilen durum_uzay gösterimi kullanılmıştır. Yani kayıp elemanları G ve R gözönüne alınmıştır. A matrisinin karakteristik denkleminin kökleri, gerçek kısmı negatif olan kompleks eşlenik köklerdir. Böylece iki tüp için karakteristik denklem¹,

$$|\lambda I - A| = (\lambda^2 + 2a_1 \lambda + a_1^2 + b_1^2)(\lambda^2 + 2a_2 \lambda + a_2^2 + b_2^2) \quad (3.10)$$

olarak yazılabilir. Burada, a_i ve b_i 'ler sırasıyla, karakteristik denklemin köklerinin gerçek ve sanal kısımlarıdır. Kayıpsız durumda kökler formant frekanslarına eşit olmakla beraber, bu durumda kökler için bir bilgi edinilememektedir. Eğer devre elemanları arasında,

$$\frac{R_1}{L_1} = \frac{R_2}{L_2} = \frac{G_1}{C_1} = \frac{G_2}{C_2} = d \quad (3.11)$$

koşulu sağlanabilirse sanal eşlenik köklerin d kadar sola kaydığı bilinmektedir (Budak, 1974). Eşitlik (3.11)'de verilen ifadeleri eşitlik (3.1)'de yerlerine koyarak ve eşitlik (3.10)'daki a_i 'ler yerine d koyarak gerekli işlemler ve basitleştirmeler yapıp, aynı dereceye sahip katsayılar eşitlenirse elde edilen denklemlerin, ya eşitlikler (3.6-3.9)'da verilenler ile aynı olduğu yada taraf tarafa toplanmış biçimleri olduğu gözlenmiştir. Buradan, eğer eşitlik (3.11)'de verilen koşul sağlanmak suretiyle kayıp elemanları ilave edilirse ses bölgesinin enine kesit alanlarının değişmeyeceği sonucu çıkmaktadır.

Eğer eşitlik (3.3)'de verilen L, C, R ve G ifadelerinde sabitler yerine konarak basitleştirilirse.

$$\frac{R_1}{L_1} = \sqrt{W_1/A_1} \quad (3.12)$$

¹Dört tüp için ifadeler çok uzun olduğundan ve dört tübe ait ifadeler Ek-4'de verildiğinden burada iki tüp incelenmiştir.

ve

$$\frac{G_1}{C_1} = 0.44 \sqrt{W_1/A_1} \quad (3.13)$$

eşitlikleri elde edilir. Buradan,

$$\frac{R_1}{L_1} = 2 \frac{G_1}{C_1} \quad (3.14)$$

kabullenmesini yaparak, eşitlik (3.1)'de verilen A matrisinden elde edilecek 4*4'lük A matrisinde (iki tüp için) yerine konursa ve bu matrisde eşitlik (3.10)'da yerine konarak gerekli işlemler yapıp, aynı dereceden λ 'lara sahip katsayılar birbirine eşitlenirse aşağıdaki denklemler elde edilebilir.

$$1.5\sqrt{W_1/A_1} + 1.5\sqrt{W_2/A_2} = 2(a_1 + a_2) \quad (3.15)$$

$$0.5(W_1/A_1) + 0.5(W_2/A_2) + 2.25\sqrt{W_1W_2/A_1A_2} + c^2/l_2^2 + c^2/l_1^2 + c^2A_1^2/l_1l_2A_2 = a_1^2 + a_2^2 + b_1^2 + b_2^2 + 4a_1a_2 \quad (3.16)$$

$$0.75\sqrt{W_1/A_1(W_2/A_2)} + 0.75\sqrt{W_2/A_2(W_1/A_1)} + 1.5\sqrt{W_1/A_1(c^2/l_2^2)} + [\sqrt{W_2/A_2} + 0.5\sqrt{W_1/A_1}] (c^2A_1^2/l_1l_2A_2) + 1.5\sqrt{W_2/A_2(c^2/l_1^2)} = 2(a_1a_2^2 + a_1b_2^2 + a_1^2a_2 + a_2b_1^2) \quad (3.17)$$

$$0.25(W_1W_2/A_1A_2) + 0.5(c^2W_1/l_2^2A_1) + 0.5\sqrt{W_1W_2/A_1A_2}(c^2A_1^2/l_1l_2A_2) + 0.5(c^2W_2/l_1^2A_2) + c^4/l_1^2l_2^2 = a_1^2a_2^2 + a_1^2b_2^2 + a_2^2b_1^2 + b_1^2b_2^2 \quad (3.18)$$

Eşitlikler (3.15-3.18)'de l_i 'ler ve c bilinmektedir; W_i 'ler ise formant frekanslarına eşittir ve bunlarda çeşitli yöntemler ile bulunabilmektedir. Burada, b_i 'lerin kayıpsız durumdaki köklere yani formant frekanslarına eşit olduğu kabul edilirse eşitlikler (3.15-3.18), dört bilinmeyenli (A_i, a_i) dört denklem şekline dönüşmüş olur. Eğer dört tüp için gerekli ifadeler türetilmiş olsa idi sekiz bilinmeyenli sekiz denklem elde edilecekti.

Dört tüpten oluşan ses bölgesi için de yukarıda sıralanan işlemler yapılmış ve elde edilen denklem seti (sekiz bilinmeyen sekiz denklem), çözücü programlar ile birlikte Ek-4'de verilmiştir.

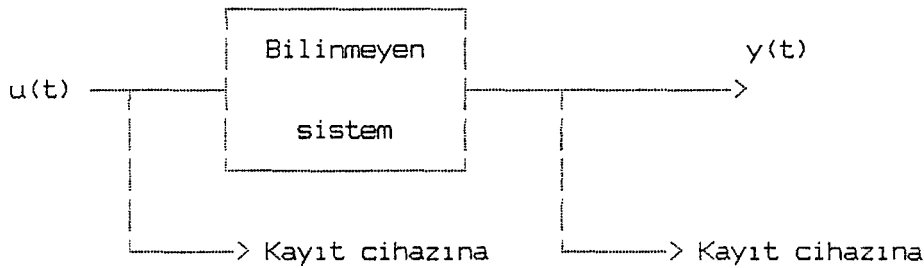
BÖLÜM 4

SES BÖLGESİNİN BELİRLENMESİ

4.1. Giriş

Fizik bilimi ve mühendislik alanlarında, sistem davranışı hakkında analitik tahminler yapmak için bazı fiziksel sistemlerin matematiksel bir modeli geliştirilir. Kontrol uygulamalarında, kontrol edilmek istenen fiziksel bir sistemin modellenmesi ile sık sık ilgilenilir. Böylece o sistemdeki karışıklıkların ve kontrol çabalarının etkisi önceden kestirilebilir.

Genellikle sistem modeli, fizik kurallarının uygulanması ve dinamik denklemlerin elde edilmesi ile oluşturulur. Bazen de model, bilinmeyen kimyasal reaksiyonlardan, fiziksel süreçlerdeki bilinmeyen sınır koşullarından veya sürecin çok karmaşık olmasından dolayı fiziksel kurallardan elde edilemez. Böyle durumlarda, bir sistem modeli oluşturmak için deneysel yöntemlere başvurulur. Bu yol ile belirlenecek bir sisteme ait blok diyagramı şekil 4.1'de verilmiştir (Jacquot, 1981).



Şekil 4.1. Belirlenecek sisteme ait blok diyagramı

Sistem girdisinin ($u(t)$) ve çıktısının ($y(t)$) ölçümleri alındıktan sonra sistemin dinamik yapısı, çıktı ve girdi arasındaki ilişkilerden belirlenir. Girdi, rastgele uyartım, impuls uyartım, birim basamak uyartım ve sinüsoidal uyartım gibi çeşitli şekillerde olabilir. Çıktı/girdi oranı sistem hakkında bilgi verir ve sisteme ait transfer fonksiyonu olarak bilinir (Jacquot, 1981).

4.2 Sistem Belirleme Prosedürü

4.2.1 Üç temel esas

Eldeki verilerden bir model oluşturulması üç temel esasa bağlıdır (Ljung, 1987):

i) Veri kaydı: Belirlenecek sisteme ait girdi-çıkıtı verileri özellikle dizayn edilmiş bir deney esnasında kaydedilir. Bu çalışma da, gırtlak ve dudaktan alınan sinyaller belli zaman aralıklarında ölçülerek bilgisayara kaydedildi.

ii) Modeller seti : Modellerin toplanması ile model adaylarının bir seti elde edilir ve bunların içinden uygun bir tanesi seçilir. Burada sistem belirleme prosedürünün seçimi en zor ve en önemli safhadır. Daha sonra bazı bilinmeyen fiziksel parametrelere sahip bir model, temel fiziksel kurallardan ve diğer iyi bilinen ilişkilerden oluşturulur.

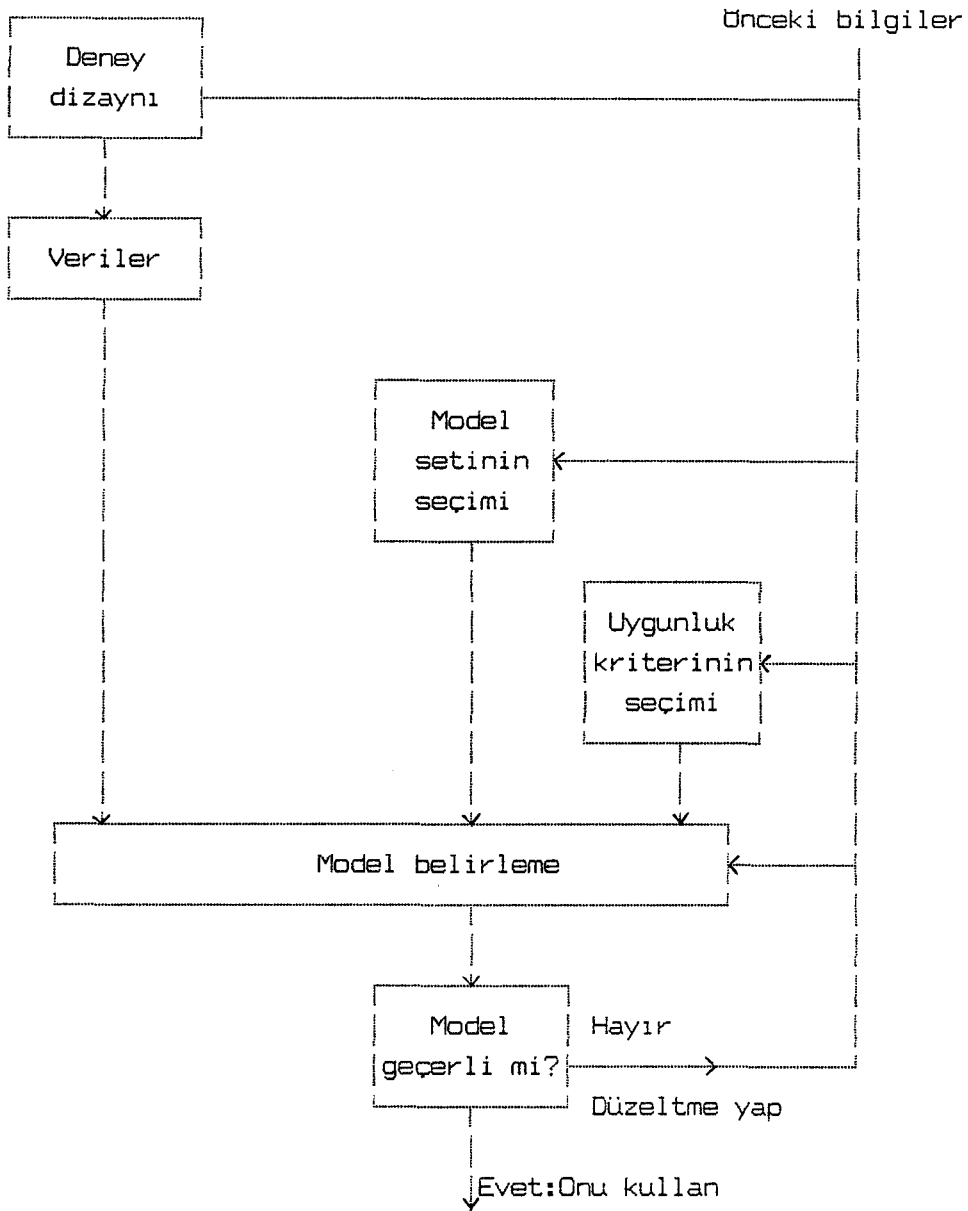
iii) Veriler ışığında set içinden en iyi modelin belirlenmesi : Bu sistem belirleme yöntemidir. Modelin geçerliliği, ölçülen verilerin yeniden elde edilmesindeki modelin performansına bağlıdır.

4.2.2. Modelin geçerliliği

Yukarda bahsedilen üç esası göz önüne aldıktan sonra set içinden en iyi olan bir model seçilir ve bu modele, çeşitli kriterler uygulanır. Böylece seçilen modelin yeterli derecede iyi olup olmadığına karar verilir. Herhangi bir model, hiçbir zaman sistemin doğru bir tanımı olarak kabul edilemez (Ljung, 1987).

4.2.3 Sistem belirleme döngüsü

Sistem belirleme prosedürünün mantıksal bir akışı vardır. İlk olarak veriler toplanır, ardından bir model seti seçilir ve son olarak da bu set içinden en iyi model alınır. İlk seçilen model çeşitli nedenlerden dolayı yeterli olmayabilir. Önemli olan bu nedenleri iyi irdeleyip prosedürün çeşitli basamaklarını tekrar gözden geçirerek düzeltmektir. Sistem belirleme döngüsü şekil 4.2'de verilmiştir (Ljung, 1987).



Şekil 4.2 Sistem belirleme döngüsü.

4.3 Sistem Belirleme Yöntemleri

Sistem belirlemede kullanılan değişik yöntemler vardır. Bunlardan birkaçı şöyle sıralanabilir: Kendiliğinden geriye doğru ilerleyen ortalama (Autoregressive moving average=ARMA) yöntemi, durum_uzay (state_space) yöntemi, en küçük kareler (least-square) yöntemi.

Durum_uzay yöntemi önceki bölümde incelenmişti. Bu bölümde en küçük kareler yöntemi incelenecek ve sabit parametrelere sahip

sistemlerin belirlenmesi ile ilgilenilecektir. Parametreler, eşitlik (4.1)'de verilen lineer matris ilişkisi vasıtası ile ölçümlere bağlı olacak ve bir parametre vektörü (\underline{a}) oluşturacaktır (Jacquot, 1981).

$$\underline{y} = \underline{A} \underline{a} + \underline{v} \quad (4.1)$$

Burada, \underline{y} , $n \times 1$ 'lik ölçüm vektörü; \underline{a} , $p \times 1$ 'lik tahmin edilecek parametre vektörü; \underline{v} , $n \times 1$ 'lik alınan verilerdeki gürültü veya hata vektörü ve \underline{A} , $n \times p$ 'lik veri matrisidir.

Tahmin edilecek model,

$$\hat{\underline{y}} = \underline{A} \underline{a} \quad (4.2)$$

şeklinde ifade edilebilir. Eşitlik (4.1)'deki hata vektörünün minimize edilmesi modelin uygunluğunu garantiler. Hataların karelerinin toplamını minimize etmek, eşitlik (4.3)'de verilen skalar fonksiyonu minimize etmek ile aynı anlama gelmektedir (Jacquot, 1981).

$$J = 1/2 \underline{v}^T \underline{v} = 1/2 \sum v_i^2 \quad (4.3)$$

Burada, $1/2$ faktörü uygunluk için konmuştur. Hatanın $\underline{y} - \hat{\underline{y}}$ olduğu yani gerçek değer ile tahmin değeri arasındaki fark olduğu biliniyor; böylece

$$J = 1/2 (\underline{y} - \underline{A} \underline{a})^T (\underline{y} - \underline{A} \underline{a}) \quad (4.4)$$

olarak yazılabilir. J 'yi \underline{a} 'ya göre minimize etmek için \underline{a} vektörüne göre J fonksiyonunun türevinin alınması gerekir ve

$$\frac{\delta J}{\delta \underline{a}} = (\underline{y} - \underline{A} \underline{a})^T (-\underline{A}) = \underline{0} \quad (4.5)$$

ifadesi elde edilir. Eşitlik (4.5)'in transpozesi alınırsa,

$$\underline{A}^T (\underline{y} - \underline{A} \underline{a}) = \underline{0} \quad (4.6)$$

ve

$$\underline{A}^T \underline{y} - \underline{A}^T \underline{A} \underline{a} \quad (4.7)$$

eşitlikleri elde edilir. Eşitlik (4.7)'den \underline{a} vektörü çekilirse tahmin edilecek parametre vektörü $\hat{\underline{a}}$ bulunmuş olur ve böylece

$$\hat{\underline{a}} = (\underline{A}^T \underline{A})^{-1} \underline{A}^T \underline{y} \quad (4.8)$$

ifadesi elde edilir. Burada, $\underline{A}^T \underline{A}$ matrisi, $p \times p$ 'lik bir kare matristir. Eğer $p > n$ ise, $\underline{A}^T \underline{A}$ 'nın rankı p 'den küçük olur. Bu durumda matris "singular" olur ve tersi alınamaz. Alınan veri sayısının, tahmin edilecek parametre sayısından fazla olması gerekmektedir (Jacquot, 1981).

4.4 En Küçük Kareler Yöntemi ile Transfer Fonksiyonunun Belirlenmesi

Bir kesikli_zaman sisteminin girdi ($u(k)$) ve çıktı ($y(k)$) dizilerinin elde edildiği ve saklandığı varsayılarak bu sisteme ait transfer fonksiyonu belirlenebilir. Bu transfer fonksiyonu z düzleminde çıkarılır ve genel olarak,

$$G(z) = \frac{Y(z)}{U(z)} = \frac{\beta_1 z^{-1} + \beta_2 z^{-2} + \dots + \beta_m z^{-m}}{1 + \alpha_1 z^{-1} + \alpha_2 z^{-2} + \dots + \alpha_n z^{-n}} \quad (4.9)$$

şeklinde ifade edilebilir (Morikawa ve Fujisaki, 1984). Eşitlik (4.9)'un fark denklemi ise,

$$y(k) = -\alpha_1 y(k-1) - \dots - \alpha_n y(k-n) + \beta_1 u(k-1) + \dots + \beta_m u(k-m) \quad (4.10)$$

olarak yazılabilir. $Y(k)$ ve $u(k)$ 'nin, k 'nin negatif değerleri için sıfır olduğu kabul edilirse eşitlik (4.10) matris şekline dönüştürülebilir.

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} u(0) & 0 & 0 & \dots & 0 & -y(0) & 0 & \dots & 0 \\ u(1) & u(0) & 0 & \dots & 0 & -y(1) & -y(0) & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u(N-1) & u(N-2) & \dots & u(N-m) & -y(N-1) & \dots & -y(N-n) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_m \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} + \begin{bmatrix} V(1) \\ V(2) \\ \vdots \\ \vdots \\ V(N) \end{bmatrix} \quad (4.11)$$

Burada N , çıktıya ve girdiye ait örnek sayısıdır. Böylece eşitlik (4.1)'deki y vektörü ve A matrisi eşitlik (4.11)'de tanımlanmış olur. Bunların eşitlik (4.8)'de yerine konması ile α_i ve β_i 'lerden oluşan parametre vektörü a elde edilir. Belli bir zaman aralığında örnek sayısının fazla olması yani küçük zaman dilimlerinde örnekler alınması daha iyi bir parametre vektörü verir (Jacquot, 1981).

4.4.1 Transfer fonksiyonunun derecesinin belirlenmesi

Ses bölgesinin transfer fonksiyonunun kutup ve sıfırlarının sayısının doğru tahmini, onların frekanslarındaki geçici değişmelerin doğru izlenebilmesi için gerekli koşuldur. Transfer fonksiyonunun pay ve paydasının derecesi, ses bölgesinin yapısına bağlıdır ve konuşmaya göre değişir (Morikawa, 1990).

Pay ve paydanın derecesinin belirlenmesi için değişik yöntemler vardır. Bu bölümde üç yöntemden bahsedilecektir.

4.4.1.1. En küçük kareler yöntemi

Bu yöntemde göre, hataların karelerinin toplamının belli bir değerin altına düştüğü andaki dereceler optimum derecelerdir. Sınır değerin belirlenmesi ilerdeki bir bölümde açıklanacaktır.

Eşitlik (4.11)'de verilen hata vektörünün elemanlarının karelerinin toplamı (SSE),

$$SSE = \sum_{i=1}^N v^2(i) \quad (4.12)$$

eşitliği ile bulunur. Eşitlik (4.9)'da verilen transfer fonksiyonunda m ve n dereceleri 1'den 10'a kadar arttırıldı. Burada $m \leq n$ şartının sağlanması gerektiğinden m 'nin her artışında n , m 'den başlayıp 10'a kadar arttırıldı. m ve n 'nin her yeni değeri için SSE hesaplanıp sınır değerin altında olup olmadığına bakıldı ve böylece optimum dereceler belirlendi.

4.4.1.2. "Signal to residue ratio" yöntemi

ARMA modeline göre oluşturulan transfer fonksiyonunun derecelerinin belirlenmesinde "signal to residue ratio" (SRR) yöntemi

kullanılmaktadır (Morikawa, 1990). SRR değerinin maksimum olduğu andaki m ve n değerleri optimum derecelerdir. SRR'in eşitliği

$$SRR(n,m) = 10 \log_{10} \frac{b_N}{r_N} \quad (4.13)$$

olarak ifade edilmektedir. Burada r_N ve b_N sırasıyla, SEARMA (simultaneous estimation of ARMA) yöntemiyle hesaplanan N'inci örneğe ait "residue"ların ve çıktı (ses) sinyallerinin varyanslarıdır ve aşağıda verilen eşitliklere sahiptirler (Morikawa, 1990).

$$b_i = \frac{i-1}{i} b_{i-1} + \frac{1}{i} s_i^2 \quad i = 1, \dots, n \quad (4.14)$$

$$r_i = \frac{i-1}{i} r_{i-1} + \frac{1}{i} w_i^2 \quad i = 1, \dots, m \quad (4.15)$$

$$w_i = s_i - \hat{S}_{i-1}^T \hat{\Phi}_{i-1} \quad (4.16)$$

Eşitlik (4.16)'da verilen w_i , "residue" sinyali; s_i , ses sinyali; \hat{S}_{i-1}^T ve $\hat{\Phi}_{i-1}^T$ aşağıdaki ifadelerle sahip olan tahmin vektörleridir.

$$\hat{S}_{i-1}^T = [s_{i-n}, \dots, s_{i-1}, \hat{u}_{i-m}, \dots, \hat{u}_{i-1}] \quad (4.17)$$

$$\hat{\Phi}_{i-1}^T = [-\hat{\alpha}_n, \dots, -\hat{\alpha}_1, \hat{\beta}_m, \dots, \hat{\beta}_1]_{i-1} \quad (4.18)$$

Eşitlik (4.18)'deki $i-1$ indisi, $i-1$ 'inci örnekleme noktasındaki vektörü göstermektedir. Tahmin vektörleri $\hat{\Phi}_i$ ve \hat{U}_i , SEARMA yöntemi vasıtasıyla aşağıda verilen eşitlikler olarak belirlenmiştir (Morikawa ve Fujisaki, 1982).

$$\hat{\Phi}_{i+1} = \hat{\Phi}_i + K_i (s_{i+1} - \hat{S}_i^T \hat{\Phi}_i) \quad (4.19)$$

$$K_i = P_i \hat{S}_i (\hat{S}_i^T P_i \hat{S}_i + 1)^{-1} \quad (4.20)$$

$$P_i = P_{i-1} - P_{i-1} \hat{S}_{i-1} (\hat{S}_{i-1}^T P_{i-1} \hat{S}_{i-1} + 1)^{-1} \hat{S}_{i-1}^T P_{i-1} \quad (4.21)$$

$$\hat{U}_i = s_i - \hat{S}_{i-1}^T \hat{\Phi}_i \quad (4.22)$$

4.4.1.3 "Akaike information criterion" yöntemi

Bu yöntem de transfer fonksiyonunun derecesini belirlemede kullanılmaktadır ve

$$AIC(n,m) = N \log r_N + 2(n+m) \quad (4.23)$$

eşitliği ile ifade edilmektedir (Morikawa, 1990). Burada N, yine örnek sayısı ve r_N 'de "residue" varyansıdır. Eşitlik (4.23)'deki ifadenin minimum olduğu andaki n ve m dereceleri optimum değerlerdir.

4.4.1.4 Optimum derecelerin bulunması

Transfer fonksiyonunun parametre sayısı veya derecesi arttıkça hataların karelerinin toplamı (SSE) azalacaktır. Parametrelerin sayısı n_1 'den n_2 'ye arttığı zaman SSE'nin önemli ölçüde azalıp azalmadığını test etmek için aşağıdaki test ifadesi kullanılabilir (Aström and Eykhoff, 1971).

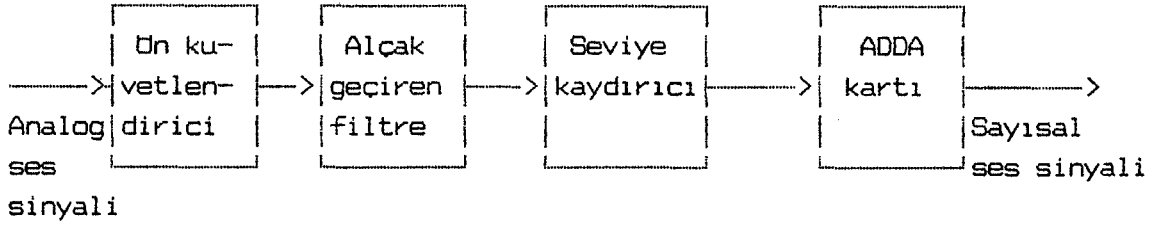
$$t = \frac{V_1 - V_2}{V_2} \frac{N - n_2}{n_2 - n_1} \quad (4.24)$$

Burada V_i , n_i ($i=1,2$) parametreye sahip bir model için SSE'nin minimum değeridir. $N > 100$ ve %5'lik bir risk seviyesinde eğer $t \geq 3$ ise SSE önemli ölçüde azalmış demektir. (Aström and Eykhoff, 1971). Böylece $t < 3$ olduğu andaki n ve m değerleri optimum derecelerdir.

4.5 Bilgisayara Ses Verilerinin Alınması

Ses sinyali analog bir sinyaldir ve bilgisayara kaydedilmesi için sayısala çevrilmesi gerekir. Yani küçük zaman aralıklarında analog sinyale ait veriler alınmalıdır. Bu aşamaları gösteren blok diyagramı şekil 4.3'de verilmiştir.

Şekil 4.3'de, analog ses sinyali ön kuvvetlendirici devresinde kuvvetlenerek alçak geçiren (LP) filtreye girer, buradan yüksek frekanslı gürültülerden arınmış olarak çıkar ve belli bir d.a. (doğru akım) seviyesi kazandırılmak için seviye kaydırıcı devresine gönderilir. Son aşama olarak ADDA (Analog_Digital, Digital-Analog) kartında sayısala çevrilir. Filtre çıkışındaki sinyale d.a. seviye kazandırılmasının nedeni ADDA kartının 0-9V arasındaki sinyali örnekleyebilmesidir. Şekil 4.3'de görülen her bir bloğun açık şeması ve ADDA kartının özellikleri Ek-5'de verilmiştir.



Şekil 4.3 Analog ses sinyalinin sayısal olarak elde edilmesi

4.6 Ses Bölgesinin Enine Kesit Alanlarının Bulunması

4.6.1 Kutup katsayılarını kullanarak alanların bulunması

Bu yöntemde elde edilen transfer fonksiyonuna tüm_kutup transfer fonksiyonu da denilebilir ve genel olarak kesikli düzlemde,

$$G(z) = \frac{Y(z)}{u(z)} = \frac{\beta_0}{1 + \alpha_1 z^{-1} + \dots + \alpha_n z^{-n}} \quad (4.25)$$

şekilinde ifade edilebilir. Bu eşitliğin fark denklemi ise,

$$y(k) = -\alpha_1 y(k-1) - \dots - \alpha_n y(k-n) + \beta_0 u(k) \quad (4.26)$$

olarak yazılabilir. $Y(k)$ ve $u(k)$ 'nin, k 'nin negatif değerleri için sıfır olduğu kabul edilirse eşitlik (4.26), matris şekline dönüştürülebilir.

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} u(1) & -y(0) & 0 & \dots & \dots & \dots & \dots & 0 \\ u(2) & -y(1) & -y(0) & \dots & \dots & \dots & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u(N) & -y(N-1) & \dots & \dots & \dots & \dots & -y(N-n) & \dots \end{bmatrix} \begin{bmatrix} \beta_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} + \begin{bmatrix} V(1) \\ V(2) \\ \vdots \\ V(N) \end{bmatrix} \quad (4.27)$$

Eşitlik (4.1) ile (4.27) karşılaştırıldığında gerekli vektör ve matrisler tanımlanabilir. β_0 ve α_i 'lerden oluşan parametre vektörü eşitlik (4.8)'den elde edilebilir.

4.6.1.1 Transfer fonksiyonunun derecesinin belirlenmesi

Ses bölgesindeki akustik dalgaların, hem gırtlaktan dudağa doğru,

hem de dudaktan gırtlığa doğru yayıldığı bilinmektedir. Bu dalgaların birbirlerine olan etkilerinden dolayı, bir dalganın ses bölgesinin bir bölümünden geçmesi için gerekli zamanın iki katı, "T" olarak tanımlanmaktadır ve

$$T = \frac{2l}{c} \quad (4.28)$$

olarak ifade edilmektedir (Markel and Gray, 1976). Buradaki l, bir bölümün uzunluğu ve c, ses hızı olarak daha önce tanımlanmıştır. Dolayısıyla ses sinyalinin örnekleme periyodu $t=nT$ olmaktadır. Eşitlik (4.28)'den yararlanarak, tüm_kutup transfer fonksiyonuna sahip bir ses bölgesinde, örnekleme frekansı f_s , tüp sayısı M, ses bölgesinin uzunluğu L ve sesin hızı c arasındaki ilişki,

$$f_s = \frac{M c}{2 L} \quad (4.29)$$

eşitliği ile belirlenebilir. Nedeni daha önce açıklandığı gibi bu çalışmada örnekleme frekansı 10 kHz olarak seçilmiştir. Ses bölgesinin uzunluğu da yaklaşık olarak 17 cm alınırsa eşitlik (4.29)'den tüp sayısının 10 olması gerektiği sonucu çıkmaktadır.

4.6.1.2 Kullanılacak verilerin sayısını ve cinsini belirleme

Enine kesit alanları, sadece ses verilerinden elde edileceği zaman verilere "Hamming window" ve "preamphasis" uygulanması gerektiği bilinmektedir (Markel and Gray, 1976). Fakat, hem gırtlak hem de dudaktan alınan verileri kullanarak enine kesit alanları bulunmak istendiğinde verilerin nasıl kullanılacağı bilinmemektedir. Bu çalışmada kullanılan verilerin sayısı, gırtlaktan elde edilen sinyalin minimum noktasını ve buna karşı gelen dudak sinyalindeki noktayı bularak, bu noktalardan itibaren 256 adet olarak belirlenmiştir. Öncelikle, 128 , 256, 512 ve iki periyodluk veri alınmış ve bu verileri kullanarak elde edilen fft spektrumları ve transfer fonksiyonlarının genliklerinin frekans spektrumları karşılaştırılmıştır. Eşitlik (4.25)' den de görüleceği gibi, verileri kullanarak elde edilen fft spektrumları,

$$|G| = \log^{-1} (\log|Y| - \log|U|) \quad (4.30)$$

eşitliğinden bulunan $|G|$ 'nin frekansa göre çizdirilmesi ile oluşmuştur. Transfer fonksiyonlarının fft spektrumları ise,

$$|G| = \left| \frac{\beta_0}{1 + \alpha_1 e^{-j\omega T} + \dots + \alpha_n e^{-jn\omega T}} \right| \quad (4.31)$$

eşitliğinden elde edilmiştir. Sonuçta, eşitlikler (4.30) ve (4.31)' dan elde edilen fft spektrumlarının, özellikle 256 adet veri için çok uygun olduğu görülmüştür. Çeşitli veri sayısı için spektrumlar çizilirken, bilgisayara kaydedilen orijinal veriler kullanılmıştır. Yani verilere "Hamming window" veya "preemphasis" uygulanmamıştır. "A" harfi için eşitlik (4.30)' dan elde edilen fft spektrumu şekil 4.4'de, eşitlik (4.31)' dan elde edilen fft spektrumu şekil 4.5'de verilmiştir.

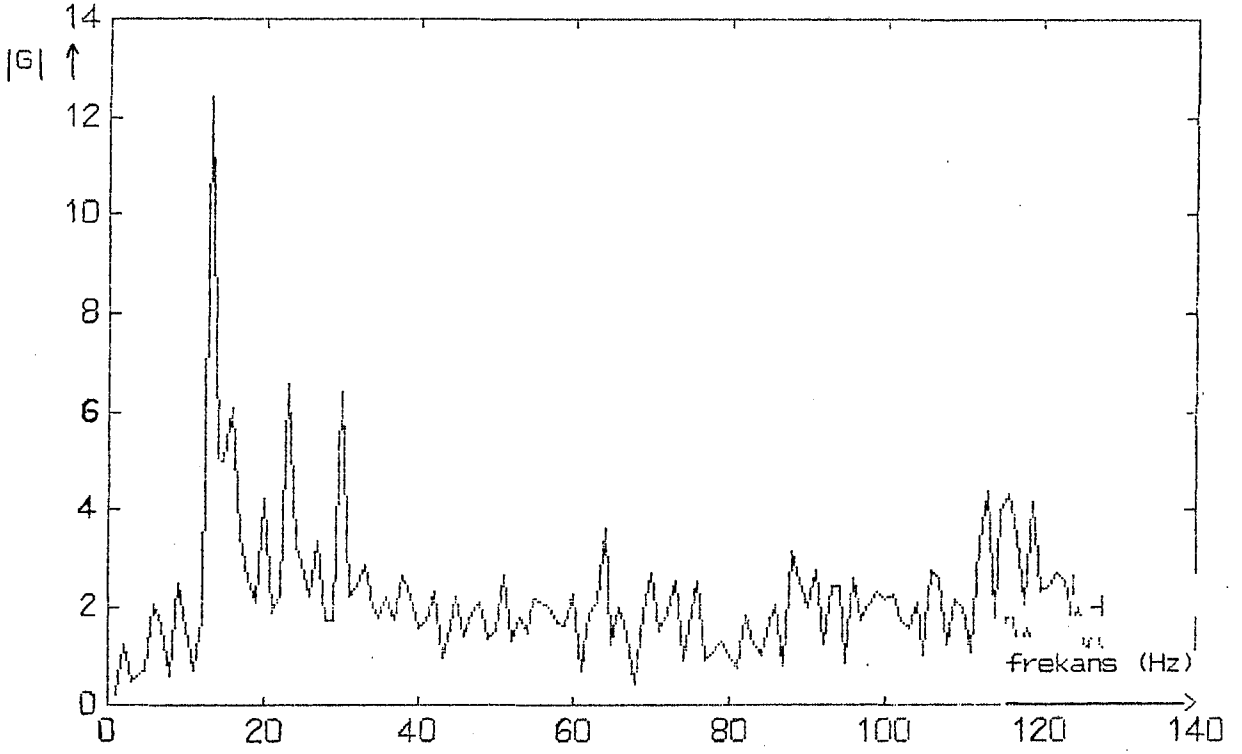
Verilere "Hamming window" ve "preemphasis" uygulandığı zaman gerçeğe yakın enine kesit alanları elde edilemedi. Dolayısıyla, öncelikle "preemphasis" uygulandı ve daha gerçekçi alanların elde edildiği gözlenerek "preemphasis" in gerekli olduğuna karar verildi. Yani dudaktan alınan veriler $1-Z^{-1}$ şeklinde bir filtreden geçirilerek dudak yayılma karakteristiğinin etkisi göz önüne alınmış oldu. Dudaktan alınan 256 veriye "preemphasis" uygulayarak eşitlik (4.30)'dan elde edilen fft spektrumu şekil 4.6'da, eşitlik (4.31)'den elde edilen fft spektrumu şekil 4.7'da verilmiştir.

Şekil 4.6'dan görüldüğü gibi yüksek frekanslı bileşenlerin genlikleri artmaktadır, yani bir türev etkisi görülmektedir. Bunun ise "preemphasis" in doğal bir sonucu olduğu bilinmektedir.

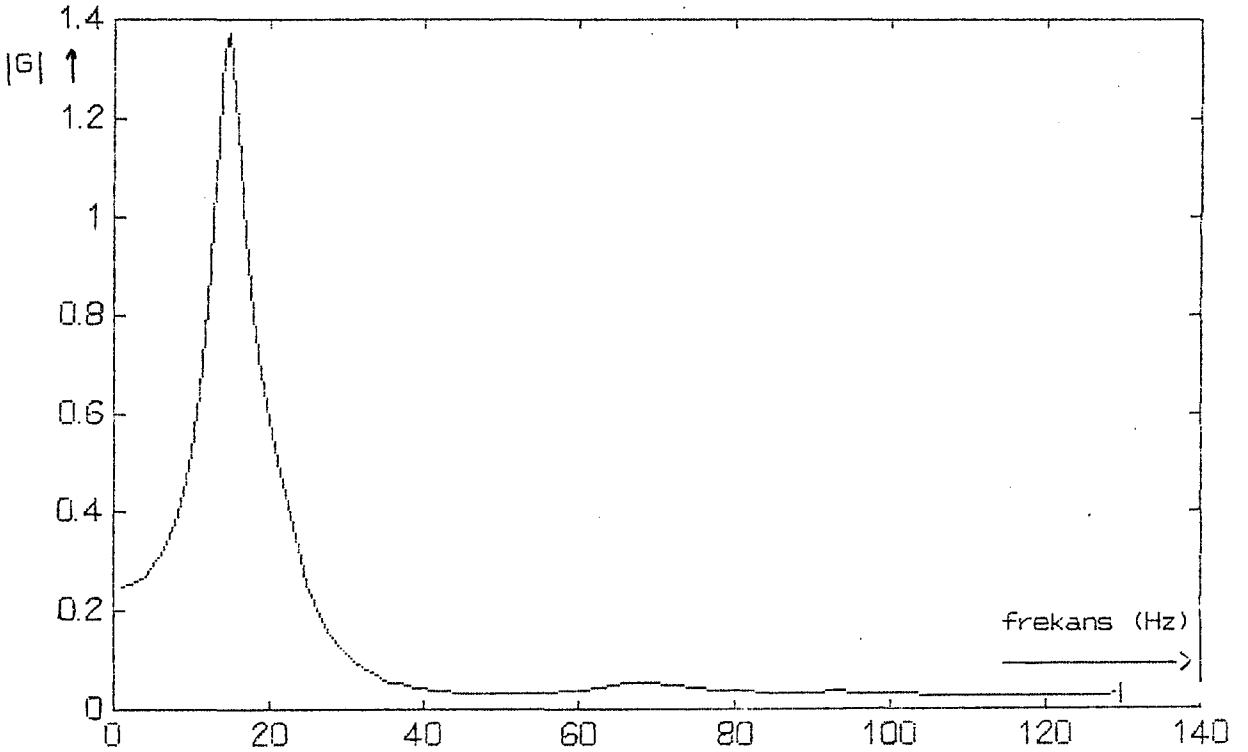
Dolayısıyla bu çalışmada, ses bölgesinin enine kesit alanları bulunurken, gırtlaktan alınan veriler olduğu gibi, dudaktan alınan veriler ise "preemphasis" uygulayarak kullanıldı.

4.6.1.3. Kutup katsayıları ile yansıma katsayıları arasındaki ilişki

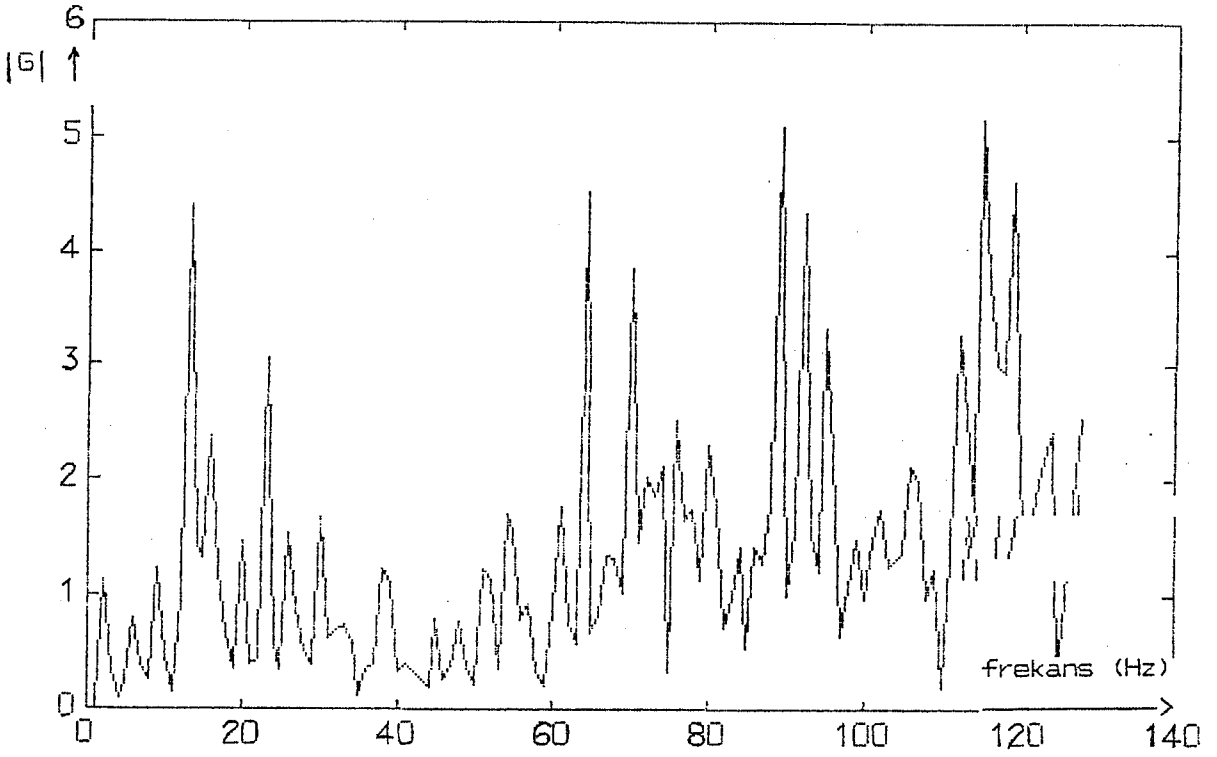
Yansıma katsayılarının (k_m) , kutup katsayılarından (α_m) bulunmasını sağlayan ifade $m=M, M-1, \dots, 1$ ve $i=0, 1, \dots, m-1$ için



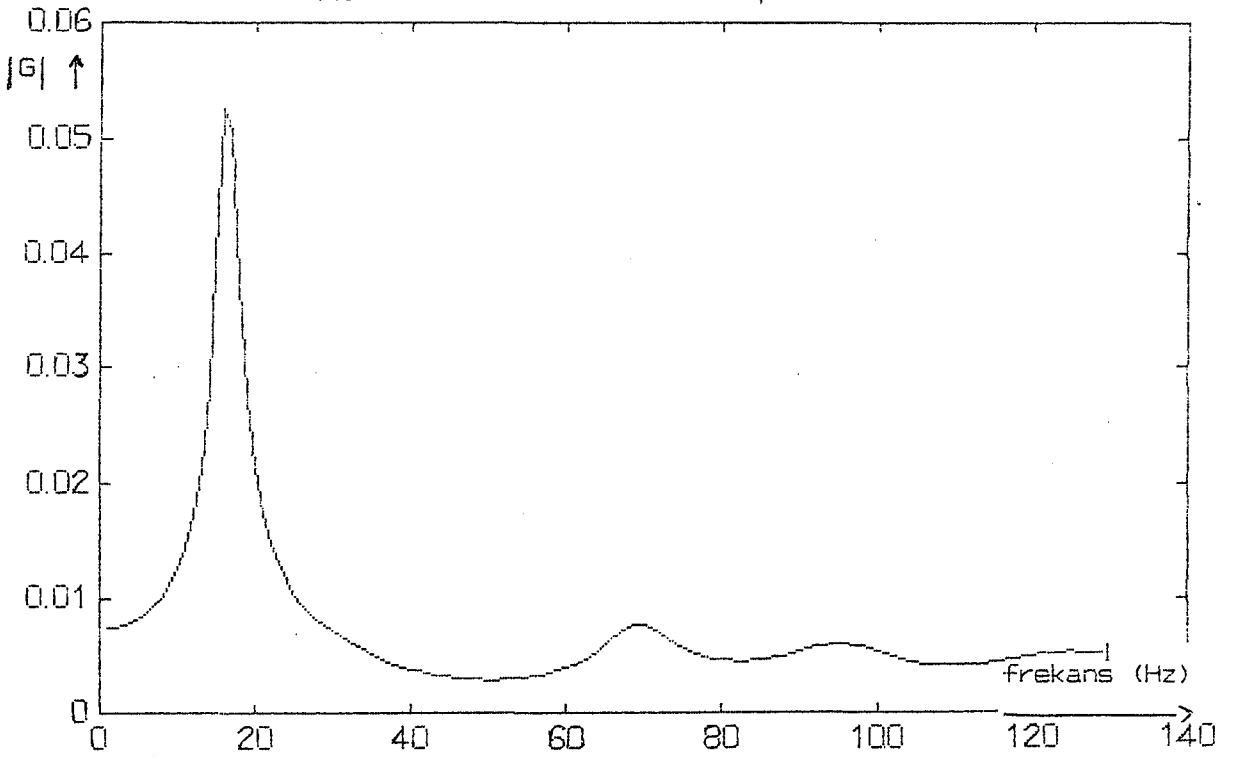
Sekil 4.4. 256 veri için eşitlik (4.30)'dan elde edilen fft spektrumu.



Sekil 4.5. 256 veri için eşitlik (4.31)'dan elde edilen fft spektrumu.



Sekil 4.6. Dudak sinyaline "preemphasis" uygulayarak eşitlik (4.30)'dan elde edilen fft spektrumu.



Sekil 4.7. Dudak sinyaline "preemphasis" uygulayarak eşitlik (4.31)'dan elde edilen fft spektrumu.

$$\alpha_{m-1,i} = \frac{\alpha_{m,i} - k_m \alpha_{m,m-i}}{1 - k_m^2} ; \quad k_m = a_{mm} \quad (4.32)$$

olarak yazılabilir (Markel and Gray, 1976). Elde edilen transfer fonksiyonunun kararlı olması için yansıma katsayılarının mutlak değerinin 1'den küçük olması gerekir. Dolayısıyla yansıma katsayıları bulunurken $|k_m| < 1$ koşulunun sağlanıp sağlanmadığına dikkat edilmelidir.

4.6.1.4. Yansıma katsayıları ile enine kesit alanları arasındaki ilişki

Tahmin edilen ses bölgesi şeklinin kesikli alan fonksiyonu,

$$A_{m-1} = \frac{1+k_m}{1-k_m} A_m \quad m=M, M-1, \dots, 1 \quad (4.33)$$

ifadesini kullanarak elde edilebilir (Markel and Gray, 1976). Burada A_m 'ler enine kesit alanları göstermektedir. Fakat A_M , gırtlaktan önceki suni bölümün alanını ifade etmekte ve genellikle 1 cm^2 olarak sabit alınmaktadır. Kutup katsayılarını kullanarak ses bölgesinin enine kesit alanlarını bulan "Matlab" dilindeki bilgisayar programı Ek.6'da verilmiştir.

4.6.2. Kutup ve sıfırları kullanarak alanların bulunması

Ses bölgesinin alan fonksiyonu, eşit uzunluklara ve sabit alanlara sahip bölümlerin kaskad şeklinde birleştirilmesi ile yaklaşık olarak elde edilebilir (Mermelstein, 1967). Herbir bölümün orta noktasındaki enine kesit alanlar,

$$A_i = \exp \sum_{n=1}^r a_n \frac{n\pi(i-1/2)h}{L} , \quad i=1,2,\dots,L/h \quad (4.34)$$

ifadesi kullanarak alan fonksiyonunun Fourier açılımının katsayılarından (a_n) bulunabilir (Mermelstein, 1967). Burada, L, ses bölgesinin uzunluğu; h, herbir bölümün uzunluğu ve r, pay ve paydanın derecelerinin toplam sayısıdır. Herbir bölüm, R, B, C, D parametrelerine sahip bir T eşdeğer devresi ile gösterilebilir. Bu

parametrelerin ifadeleri ve i'nci bölüm için elde edilen matris aşağıda verilmiştir (Mermelstein, 1967).

$$K_i(w) = \begin{bmatrix} R_i(w) & B_i(w) \\ C_i(w) & D_i(w) \end{bmatrix} \quad (4.35)$$

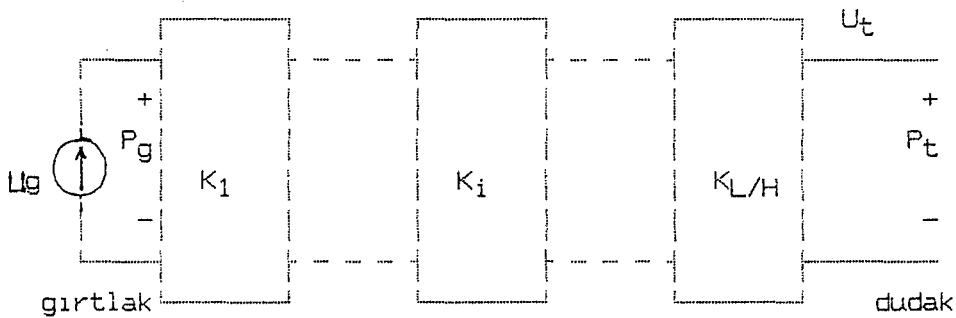
$$R_i(w) = \text{Cosh}(jwh/c) = \text{Cos} \frac{wh}{c},$$

$$B_i(w) = \frac{\rho c}{A_i} \text{Sinh} j \frac{wh}{c} = j \frac{\rho c}{A_i} \text{Sin} \frac{wh}{c},$$

$$C_i(w) = \frac{A_i}{\rho c} \text{Sinh} j \frac{wh}{c} = j \frac{A_i}{\rho c} \text{Sin} \frac{wh}{c},$$

$$D_i(w) = \text{Cosh}(jwh/c) = \text{Cos} \frac{wh}{c},$$

Buradaki, w , c , ρ ve A , eşitlik (3.3)'de verilenler ile aynı anlama gelmektedir. Ses bölgesinin bölümler halinde gösterimi şekil 4.8'de verilmiştir.



Şekil 4.8 Ses bölgesinin genel gösterimi.

Herhangi bir frekans değeri (ω) için dudaklardaki basınç (P_t) ve hacim hızı (U_t), gırtlaktaki benzer değişkenlerden (P_g, U_g),

$$\begin{bmatrix} P_t \\ U_t \end{bmatrix} = \frac{L/h}{\pi} \sum_{i=1}^{\infty} K_i(\omega) \begin{bmatrix} P_g \\ U_g \end{bmatrix} \quad (4.36)$$

ifadesi ile bulunabilir (Mermelstein, 1967). Kaynak sınır koşulları gırtlaktaki admitansın (Y_g) sıfır olmasını gerektirdiğinden ($P_g, U_g=0$) dudaktaki admitansın kutup ve sıfırları sırasıyla, $P_t=0$ ve $U_t=0$ eşitliklerinden elde edilebilir (Mermelstein, 1967). Eğer kutup ve sıfırlar transfer fonksiyonunun yani beklenen kutup ve sıfırlardan ± 1 'den daha fazla uzaklaşmış ise katsayılar (a_n) değiştirilmek suretiyle yukarıda sıralanan işlemler yapılarak kutup ve sıfırlar tekrar bulunur. Herbir katsayı bir kutup veya sıfıra bağlıdır. Elde edilen kutup ve sıfırlar istenilen aralıkta kalıncaya kadar bu işlemler tekrarlanır (Mermelstein, 1967). Katsayıların, kutup ve sıfırları dikkate alınarak nasıl ve ne kadar değiştirilmesi gerektiği ilgili yayında anlatılmıştır. Sonuçta bulunan a_i 'lerden eşitlik (4.34) kullanılarak ses bölgesinin enine kesit alanları bulunabilir. Yalnız burada, transfer fonksiyonunun pay ve paydasının derecesinin eşit olması gerekmektedir.

Bu çalışmada, gırtlaktan ve dudaktan alınan sinyallerin örneklenmesi sonucu elde edilen verilerden belirlenen transfer fonksiyonu gırtlaktaki titreşimlerin ve dudaktaki ses dalgalarının yarattığı basınçların bir oranıdır ve genel ifadesi eşitlik (4.9)'da verilmiştir. Dolayısıyla bu transfer fonksiyonunun yani basınçlarının oranının, dudaktaki basıncın hacim hızına oranı şekiline dönüştürülmesi yani dudak empedansının bulunması gerekir. Bunun için de,

$$Z_{in}(z) = Z_0 \frac{A(z)}{A^r(z)} \quad (4.37)$$

ifadesi kullanılabilir (Markel and Gray, 1976). Burada Z_{in} , dudak empedansı, $Z_0 = \rho c / A_0$, sıfırinci bölümün (dudak) karakteristik empedansı, $A^r(z)$, eşitlik (4.9)'da verilen transfer fonksiyonunun negatif yansıma katsayılarına sahip olan eşleniğidir. Böylece,

basınçların oranı şeklinde elde edilen transfer fonksiyonundan dudak empedansının elde edilmesi ve yukarıda anlatılan işlemlerin yapılması gerekir. Sonuçta "Fourier" açılımının katsayıları (a_n) ve eşitlik (4.34)'den de enine kesit alanları bulunabilir. Bu bölümde anlatılan enine kesit alanlarının elde edilme prosedürü ile ilgili bilgisayar programları "Matlab" dilinde yazılmış ve Ek.7'de verilmiştir. Sonuçlar ise bölüm 5'de karşılaştırmalı olarak verilmiştir.

Enine kesit alanlarının bulunmasında diğer bir yol ise, eşitlik (4.36)'dan

$$\begin{bmatrix} U_t \\ \\ U_g \end{bmatrix} = \begin{bmatrix} \frac{T(w)}{Q(w)} & \frac{Q(w) S(w) - P(w) T(w)}{Q(w)} \\ 1 & \frac{P(w)}{Q(w)} \end{bmatrix} \begin{bmatrix} P_t \\ \\ P_g \end{bmatrix} \quad (4.38)$$

ifadesinin türetilmesine ve sınır koşullarının ($P_t=1$, $P_g=0$) kullanılması ile $U_t=0$ ve $U_g=0$ eşitliklerinden transfer fonksiyonunun kutup ve sıfırlarının bulunmasına bağlıdır. İstenen kutup ve sıfırlar elde edilinceye kadar bu bölümün başında sıralanan işlemlerin tekrarlanması ile açılım katsayıları (a_n) ve enine kesit alanları (A_i) bulunabilir. Eşitlik (4.38) elde edilirken eşitlik (4.36)'dan da yararlanılarak aşağıdaki kabullerme yapılmıştır.

$$\prod_{i=1}^{L/h} K_i(w) = \begin{bmatrix} P(w) & Q(w) \\ S(w) & T(w) \end{bmatrix} \quad (4.39)$$

Eğer eşitlik (4.9)'da verilen transfer fonksiyonunun paydasının derecesi ses bölgesinin tüp sayısına (M) eşit alınırsa, $2M$ tane katsayıdan bulunan enine kesit alanlarından yansıma katsayılarının ve kutupların elde edilmesi mümkündür. İstenilen kutuplar elde edilinceye

kadar katsayılar (a_n) değiştirilerek işlemler tekrarlanır. Çünkü sıfırlara karşılık gelen katsayıların değişimi de kutupları etkilemektedir. Sonuçta bulunan katsayılardan eşitlik (4.34) kullanılarak ses bölgesinin enine kesit alanları elde edilebilir. Bu prosedür ve bir üst paragrafta anlatılan prosedürün "Matlab" dilinde yazılmış programları EK.8'de , karşılaştırmalı sonuçlar ise bölüm 5'de verilmiştir.

4.6.3. Enine kesit alanlarının bilgisayarda karşılaştırmalı çizimi.

Bir önceki bölümde elde edilen enine kesit alanlarının, ilk önce kesikli olarak bilgisayarda çizimi için "C" dilinde program yazıldı. Bu program aynı zamanda, farklı harflere ait enine kesit alanlarının alt alta ve üst üste çakışık olarak aynı anda bilgisayarda gösterimini sağlamaktadır. Daha sonra kesikli alan fonksiyonu, Fourier açılımının kullanılması ile düzleştirilerek eğri şekline dönüştürülmüştür. Kesikli ve eğri şeklindeki enine kesit alanlarının yine alt alta ve üst üste çakışık olarak aynı anda bilgisayarda gösterimi için programa ilave yapılmıştır. Ayrıca, farklı harflere eğri şeklindeki enine kesit alanlarının yine yukarıda belirtilen gösterimin bilgisayarda elde edilmesi bu program ile gerçekleştirilmiştir. Bu gösterimler bölüm 5'de karşılaştırmalı olarak verildiğinden ayrıca burada verilmemiştir. Enine kesit alan fonksiyonunun karşılaştırmalı çizimi için yazılan "C" dilindeki program Ek.9'da verilmiştir.

4.6.4 Enine kesit alanlarının insan kafası içine yerleştirilmesi

Bu bölüm için yapılan çalışmalarda ilk önce insan kafasının yandan görünüşü, "C" dilinde yazılan programlar vasıtasıyla bilgisayar ekranına çizildi. Daha sonra, üst damak sabit olduğundan, üst damağı temsil eden dört doğrunun denkleminde enine kesit alanları çıkarmak suretiyle dilin aldığı şeklin ekranda gösterimi gerçekleştirildi. Böylece sekiz ünlüden herhangi biri söylenirken, gırtlaktan dudaklara kadar uzanan ses bölgesinin yapısının gözleme ve inceleme olanağı sağlanmıştır. Bunun da işitme engellilere yardımcı olacağı beklenmektedir. Ses bölgesinin yapısının ekranda gösterimini sağlayan "C" dilindeki program, bir önceki bölümde anlatılan çizimleri gerçekleştiren program ile birlikte yazılmış ve Ek.9'da verilmiştir.

BÖLÜM 5

TÜM MODELLER ÜZERİNDE YAPILAN ÇALIŞMALAR VE KARŞILAŞTIRMALAR

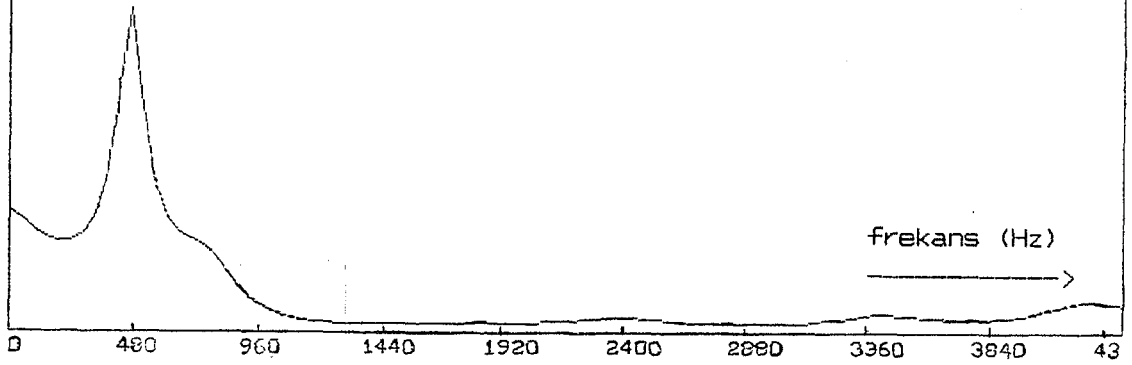
On tüpten oluşan kayıpsız ses bölgesi için elde edilen lineer olmayan denklem setini çözen paket programı kullanılacak şekilde Fortran dilinde bilgisayar programı yazıldı. Lineer olmayan denklem setinin çözümünde enine kesit alanları için başlangıç değerlerinin verilmesi gerekmektedir. Rastgele seçilen enine kesit alanlarından bulunan formant frekansları yerine konularak ve seçilen değerlere yakın başlangıç değerleri girilerek seçilen değerlerin bulunduğu ve programın çalıştığı gözlemlendi. Sekiz ünlü için alınan örneklenmiş ses sinyallerinden elde edilen LPC katsayılarından bulunan formant frekansları denklemlerde yerine konuldu. Fakat enine kesit alanları bilinmediğinden ve denklem setinin çözümü için başlangıç değerleri gerektiğinden, çok sayıda farklı başlangıç değerleri verilmesine rağmen istenilen çözüm elde edilememiştir. Program çözüme ulaştığına dair bilgi vermekle beraber, ya değerlerin bir kısmı negatif çıkmakta ya da mantıksız değerler elde edilmektedir. Bu da birden fazla çözüm olduğunu göstermektedir. Bu yöntem ile çözüme gidilememesinin diğer bir nedeni de LPC katsayılarının sayısı çok az değiştirilse bile formant frekanslarının oldukça fazla değişmesidir. Bu durum "a" ve "ö" harfleri için şekiller 5.1-5.4'de gösterilmiştir. Dolayısıyla, dört tüpten oluşan ses bölgesine göre denklem seti elde edildi ve üzerinde çalışmalar yapıldı. Ses bölgesinin gırtlak tarafındaki ilk tübün enine kesit alanı 1 cm^2 olarak alındı. Diğer üç tüp için çok sayıda, farklı başlangıç değerleri girildikten sonra özellikle tüm kutup modelinden elde edilen sonuçlara yakın değerler bulundu. Bu da kayıpsız durum için durum uzay modelinin, tüm kutup modeline eşdeğer olduğunu göstermektedir. Sekiz ünlü için elde edilen değerler çizelgeler 5.1-5.8'de verilmiştir.

Kayıplı durum için önce on tüp daha sonra dört tüp için denklem seti elde edildi. Fakat, durum_uzay modelindeki durum matrisinin özdeğerlerinin sanal kısımları formant frekanslarına eşit alınmakla beraber gerçek kısımları bilinmemektedir. Dolayısıyla denklem setindeki bilinmeyenler (gerçek kısımlar ve enine kesit alanlar) için çeşitli başlangıç değerleri girilmesine rağmen daha önce ve yukarıda anlatılan nedenlerden dolayı mantıklı çözümler elde edilememiştir.

Tüm kutup modeli üzerinde yapılan çalışmalarda öncelikle, girdi olarak "impulse" zinciri kullanılmıştır. Bugüne kadar yapılan çalışmalarda ses tellerinin titreşim kaynağı olarak impals zinciri

$|G|$

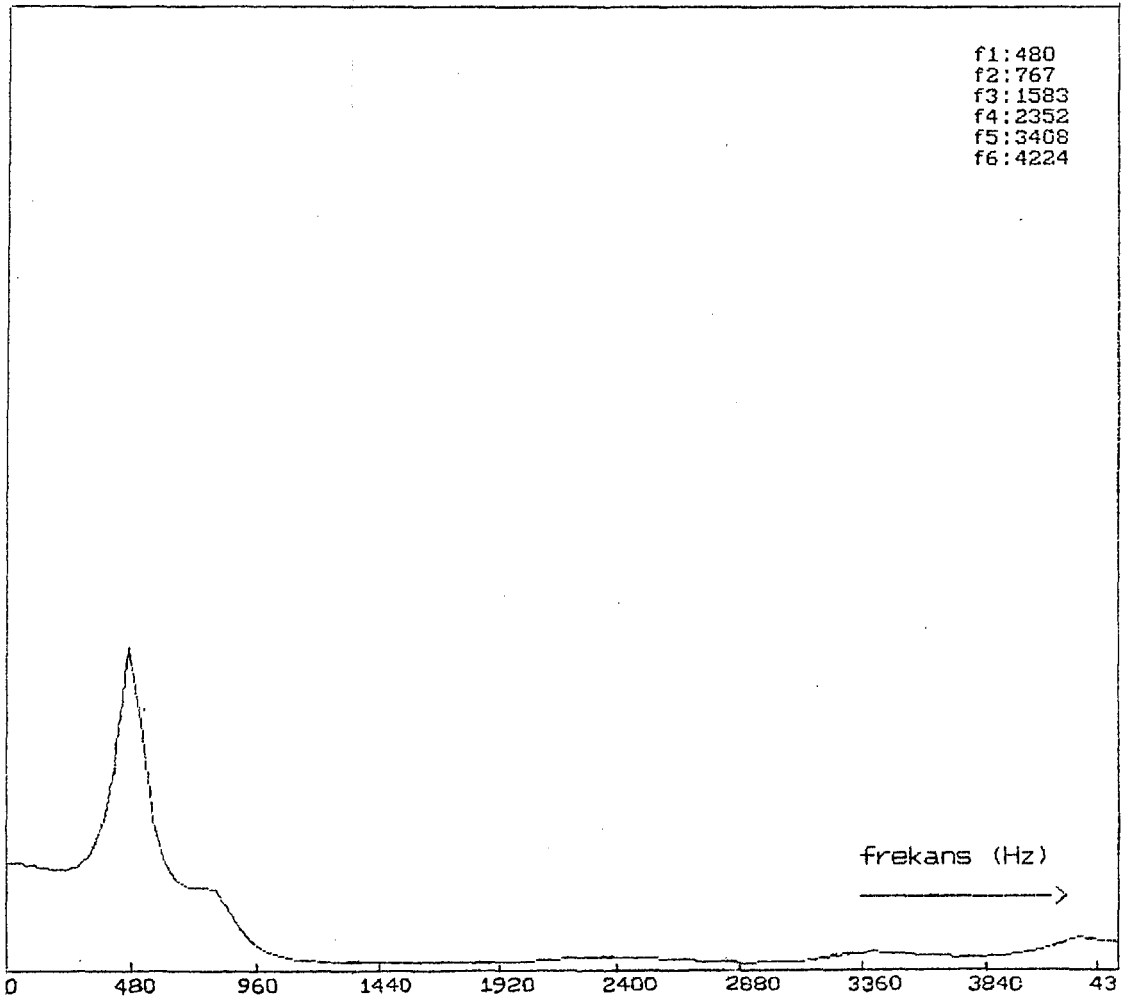
f1:480
f2:2400
f3:3408
f4:4224



Şekil 5.1. "A" harfi için 18 adet LPC katsayısından elde edilen frekans spektrumu ve formant frekansları.

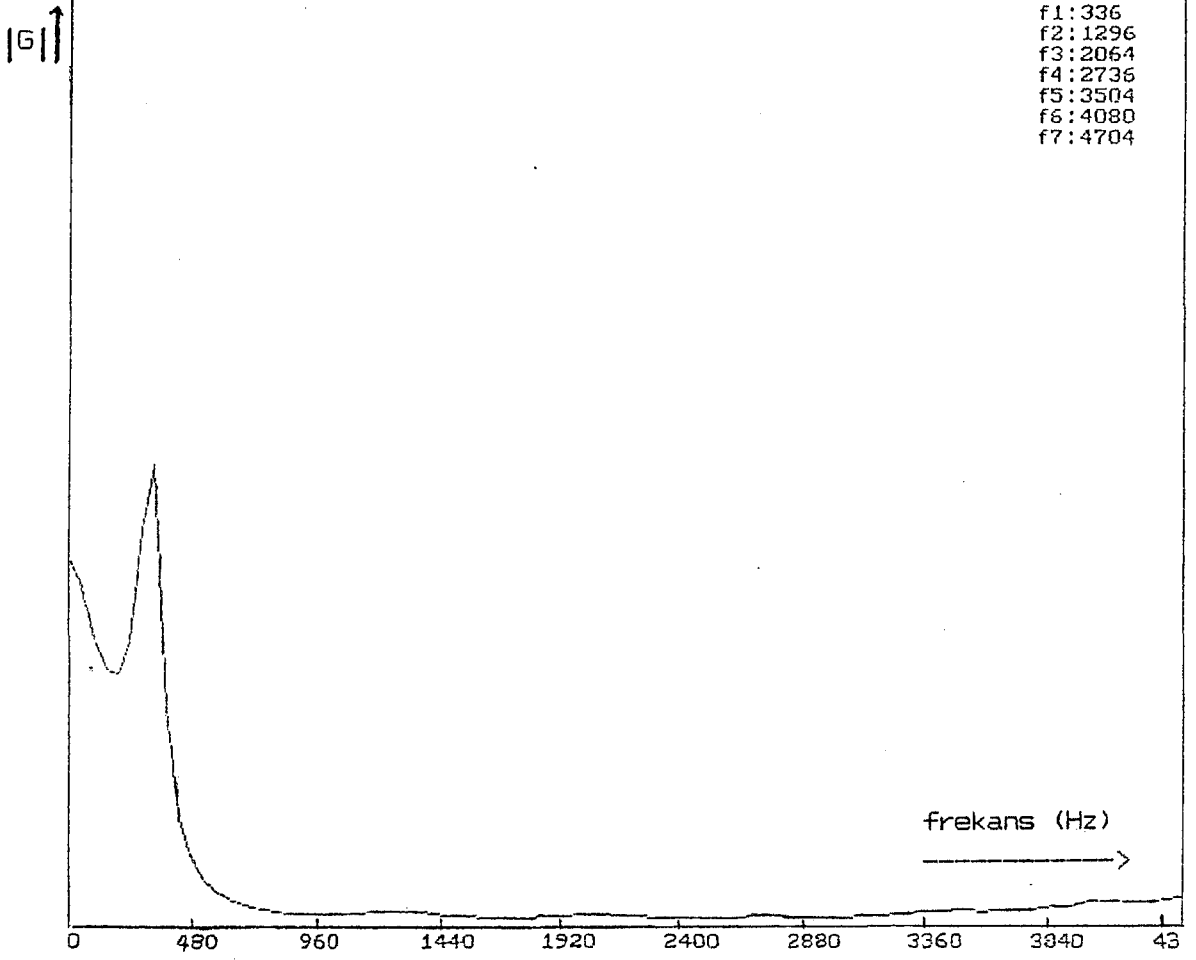
 $|G|$

f1:480
f2:767
f3:1583
f4:2352
f5:3408
f6:4224



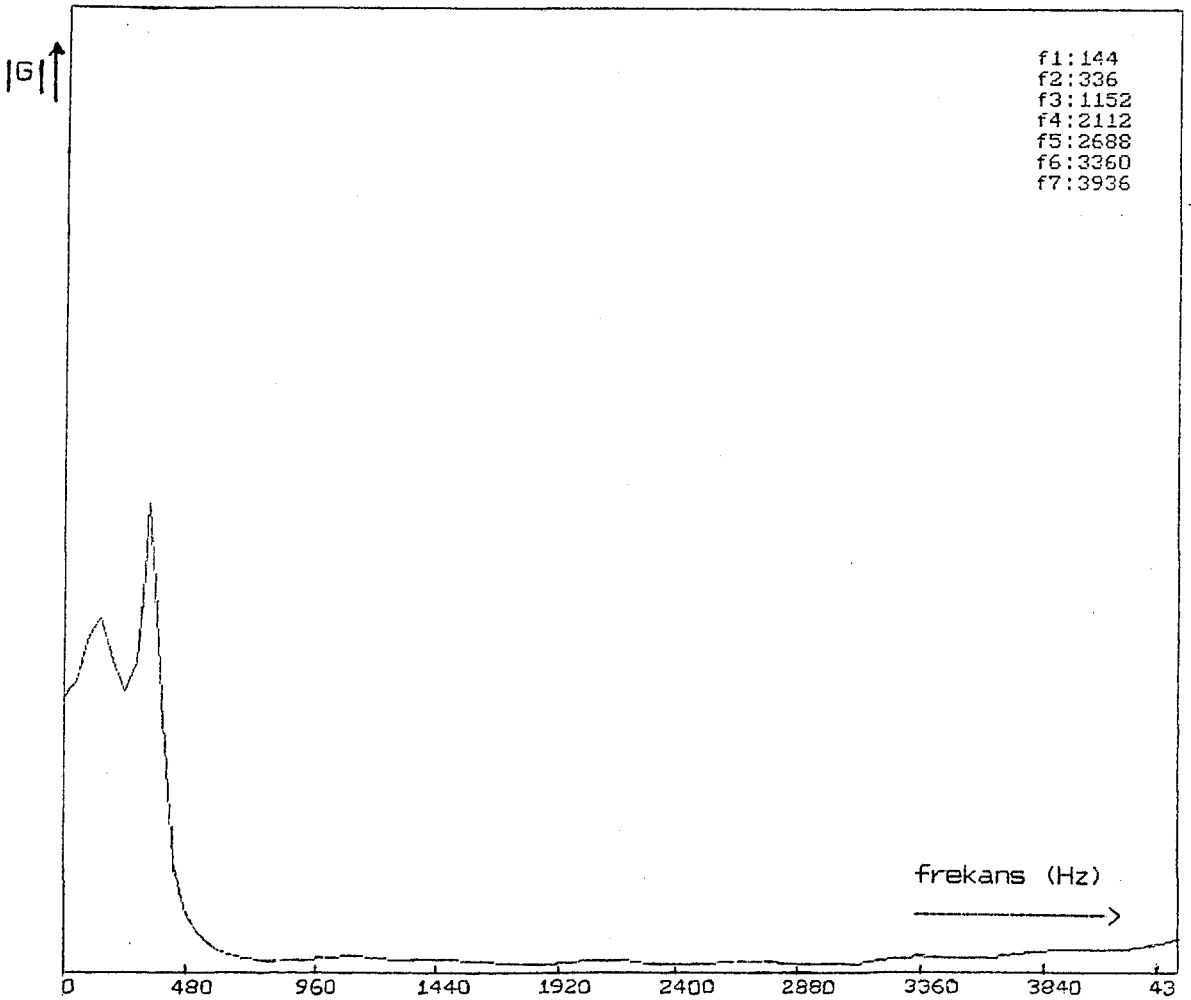
Şekil 5.2. "A" harfi için 20 adet LPC katsayısından elde edilen frekans spektrumu ve formant frekansları.

f1:336
f2:1296
f3:2064
f4:2736
f5:3504
f6:4080
f7:4704



Şekil 5.3. "O" harfi için 16 adet LPC katsayılarından elde edilen frekans spektrumu ve formant frekansları.

f1:144
f2:336
f3:1152
f4:2112
f5:2688
f6:3360
f7:3936



Şekil 5.4. "O" harfi için 20 adet LPC katsayılarında elde edilen frekans spektrumu ve formant frekansları.

kullanılmakla beraber bu impuls'ların nerede veya hangi noktada uygulandığı belirtilmemektedir. Yani başlangıç ($t=0$) noktasından mı, yoksa ses sinyalinin ilk tepe noktasına ulaştığı yerden mi periyodik olarak uygulandığı belli değildir. Tüm kutup modeline uygulanan impuls girdinin hem başlangıç noktasından hem de ses sinyalinin ilk tepe noktasından periyodik olarak uygulandığı kabul edilerek çalışmalar yapılmış ve $t=0$ 'dan itibaren uygulanan impuls için biraz daha iyi sonuçların elde edildiği gözlenmiştir. Ayrıca impuls'ların genliği de belli olmadığından hem "1", hem de ölçülen girdinin tepe değerine eşit genlikteki impuls zinciri kullanılarak enine kesit alanları bulundu ve değerler çizelgeler 5.1-5.8'de verildi.

Alternatif bir girdi için, gırtlaktaki ses tellerinin titreştiği yere piezo kristalin konulması suretiyle ses bölgesine uygulanan gerçek sinyal ölçülmeye çalışıldı. Fakat piezo kristalin karakteristiği bilinmediğinden, kristalin gerçek sinyale olan etkisi belirlenememiştir. Piezo kristalin, gerçek sinyalin seviyesini kaydırıp kaydırmadığı belli olmadığından, girdi olarak hem ölçülen sinyali kullanarak, hem sinyale negatif kısmı kadar d.a. seviye kazandırarak, hem de sinyalin negatif kısmını atarak analizler yapılmış ve sonuçların farklı olmadığı gözlenmiştir. Dolayısıyla tüm kutup ve kutup-sıfır modellerinde girdi olarak piezo kristalinden ölçülen sinyal olduğu gibi kullanılmıştır.

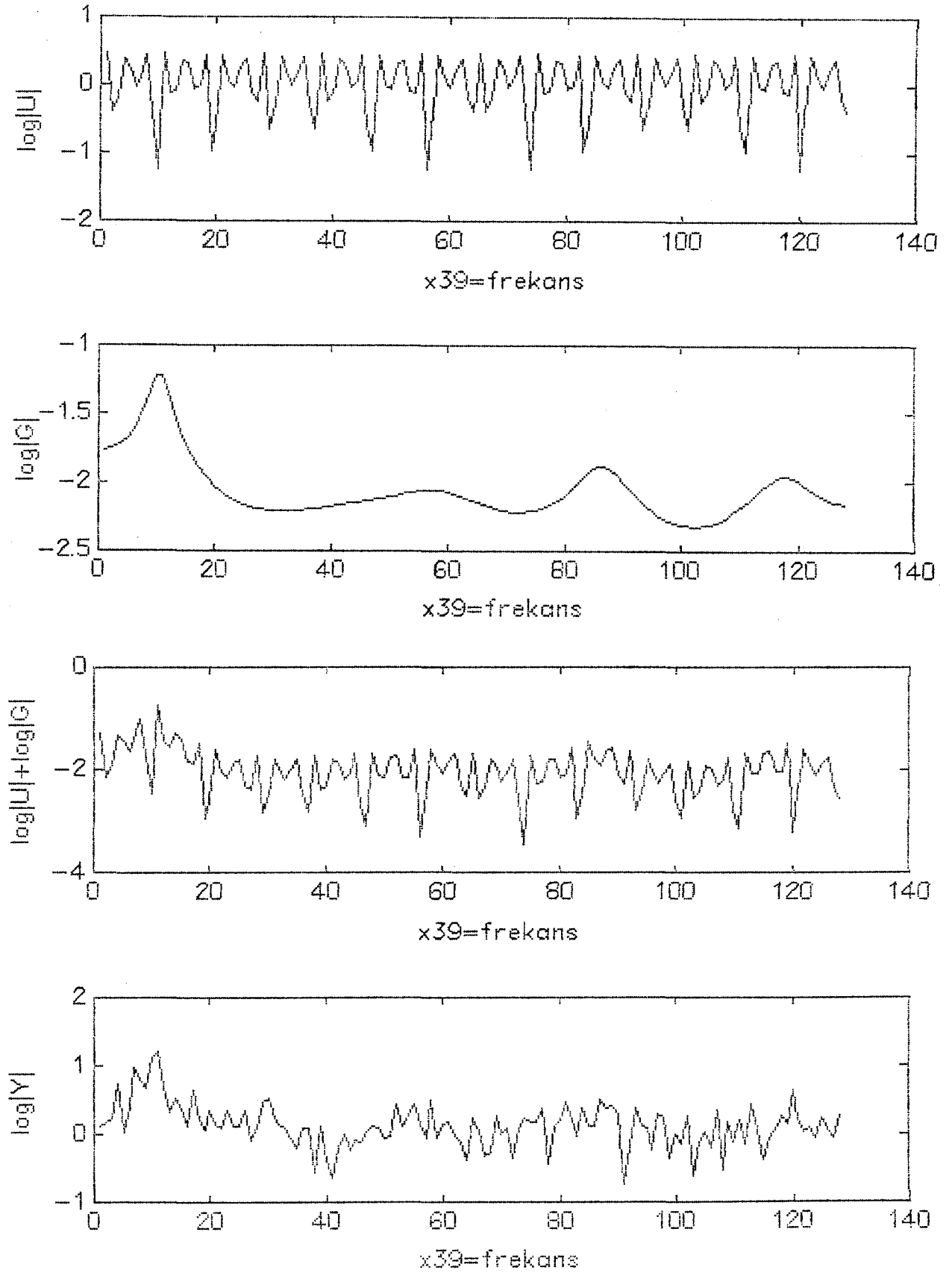
Daha önceki çalışmalarda tüm kutup modelini kullanarak oto-korelasyon ve LPC katsayılarından enine kesit alanları bulunurken transfer fonksiyonunun payı "1" olarak alınıyordu. Halbuki sistem belirleme yöntemi ile transfer fonksiyonu elde edilirken pay kısmında bir kazanç faktörünün olması gerçeğe daha uygundur. Gırtlaktan ölçülen sinyalin yukarıda anlatıldığı gibi farklı şekillerde kullanılmasının sonuçları değiştirmemesi kazanç faktörüne bağlanabilir. Çünkü, farklı girdi sinyalleri alınarak yapılan her analizin sonucunda enine kesit alanlar birbirine yakın olmakla beraber kazanç faktörü oldukça farklıdır. Sonuçta farklı girdi sinyalleri, kazanç faktörünü değiştirir, fakat enine kesit alanları pek etkilemez denilebilir. impuls ve ölçülen girdi için elde edilen enine kesit alanlarının pek farklı çıkmaması da kazanç faktöründen kaynaklanabilir. Gerçekte kazanç faktörünün pozitif olması gerekmektedir. Çünkü negatif kazanç, sistem üzerinde bazı faktörlerin ters etkisi olduğunu göstermektedir. Çizelgeler 5.1-5.8'den görüleceği gibi tüm kutup modelinde ölçülen girdi sinyalinin kullanılması, diğer girdi sinyallerine göre kazanç faktörünü pozitif yapmaktadır. Bu da, gırtlaktan ölçülen sinyalin kullanılmasının daha doğru ve güvenilir sonuçlar verdiğini göstermektedir. Kazanç açısından bakılacak olursa, ölçülen girdinin

tepe değerine eşit genlikli impuls zincirinin kullanılması da, birim impuls zincirinden daha iyi sonuçlar vermektedir. Sonuçların yakın çıkmasının diğer bir nedeni de $t=0$ 'da uygulandığı kabul edilen impuls girdinin gerçek girdinin iyi bir yaklaşımı olabileceği fikrini vermektedir.

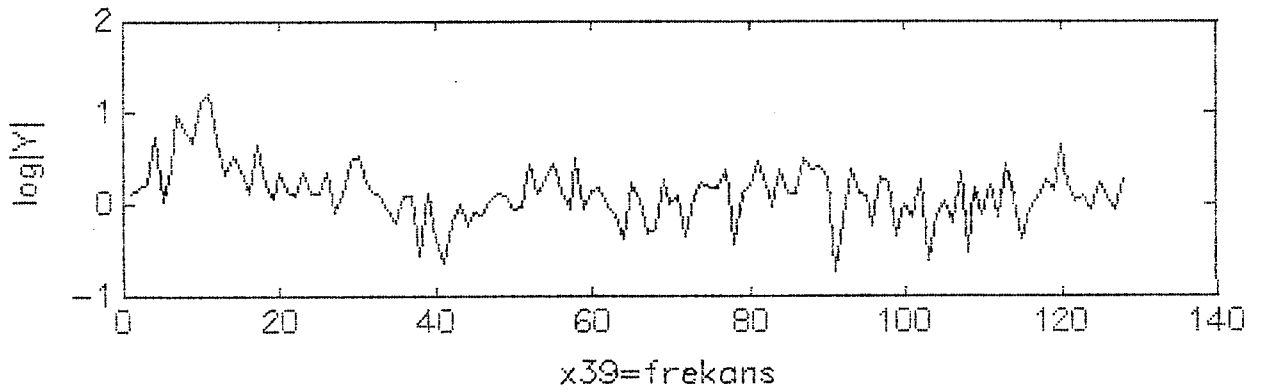
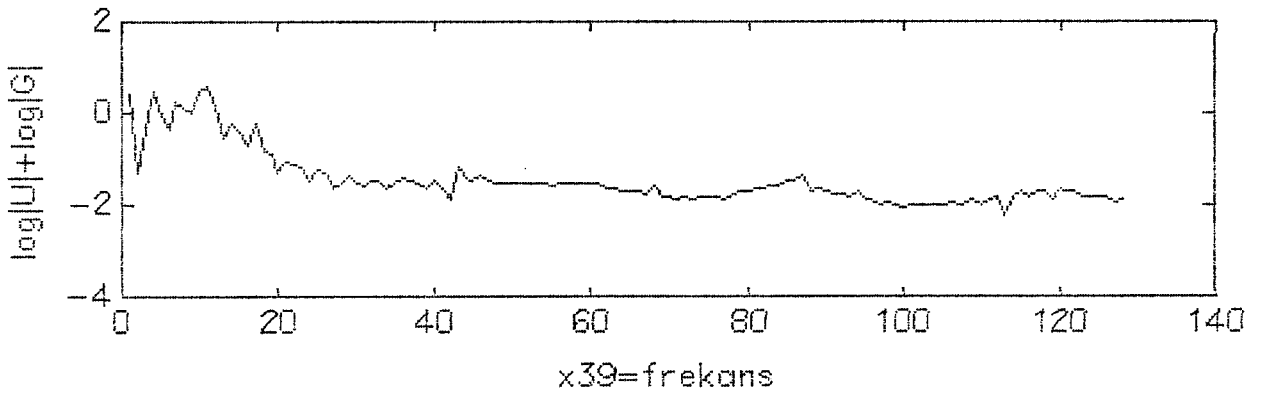
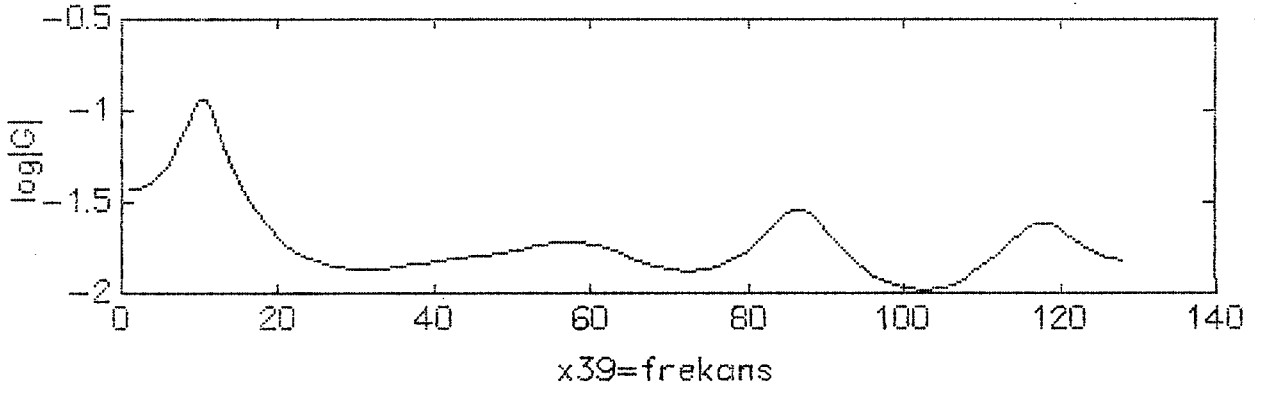
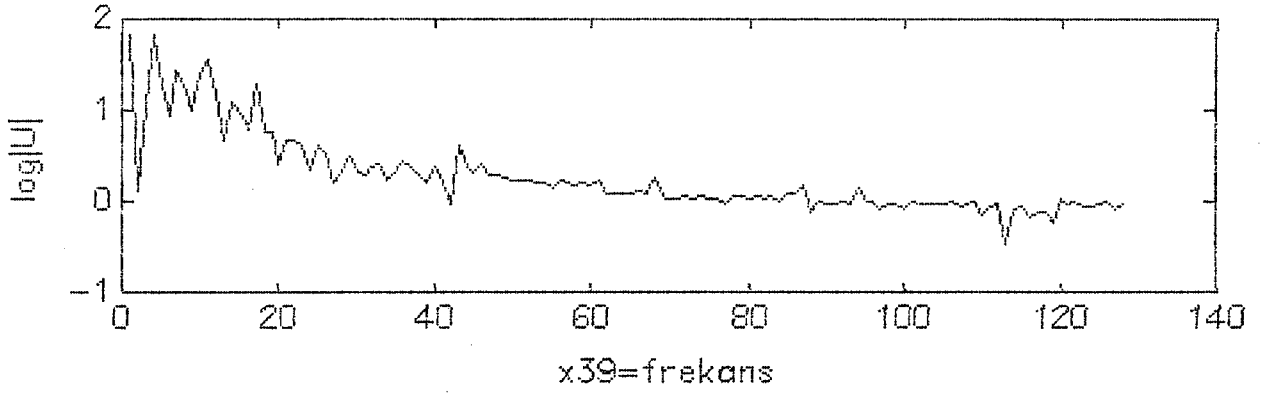
Eşitlikler (4.30) ve (4.31)'den yararlanarak elde edilen çıktının "FFT"sinin logaritması ($\log|Y|$) ile, analizlerde kullanılan çıktının "FFT"sinin logaritması impuls ve ölçülen girdi için karşılaştırılmıştır. Tüm kutup modelinde girdi olarak, impuls kullanıldığında hataların karelerinin toplamı (SSE) 643.49, ölçülen değerler kullanıldığında ise 348.4 olmaktadır. Bu durumlarla ilgili logaritmik değişimler şekiller 5.5 ve 5.6'da gösterilmiştir. Kutup_sıfır modelinde, girdi olarak impuls kullanıldığında SSE, 294.96, ölçülen değerler kullanıldığında ise 39.35 olmaktadır. Bu durumlar ile ilgili logaritmik değişimler şekiller 5.7 ve 5.8'de gösterilmiştir. Hem şekillerden hem de hataların karelerinin toplamından görüleceği gibi girdi olarak ölçülen değerler kullanıldığında elde edilen transfer fonksiyonu impuls girdiye göre daha doğru olmaktadır. Özellikle kutup_sıfır modelinde bu durum daha belirgin olarak görülmektedir.

Kutup_sıfır modeli için ilk olarak eşitlik (4.37) kullanılarak dudak admitansı elde edildi. Admitans fonksiyonunun optimum dereceleri, alt bölüm (4.4.1)'de değinilen kriterlerin uygulanması ile belirlendi ve, bu derecelerdeki kutup ve sıfırlar beklenen değerler olarak alındı. Bu değerler, Mermelstein (1967)'nin geliştirdiği ve eşitlikler (4.34-4.36) ile ifade edilen yöntemden bulunan kutup ve sıfırları ile karşılaştırıldı. Aradaki farka göre enine kesit alan fonksiyonunun Fourier açılımının katsayıları ± 0.05 adımlar ile değiştirilerek farkın, beklenen değerlerin %1'ne kadar indirilmesine çalışıldı. Diğer kutup_sıfır modelinde ise sistem belirleme yöntemi ile elde edilen ses bölgesinin transfer fonksiyonunun kutup ve sıfırları beklenen değerler olarak alındı. Bu değerler eşitlik (4.38)'den elde edilen kutup ve sıfırları ile karşılaştırıldı. Yukarıda anlatılan işlemler tekrar edilerek farkın, beklenen değerlerin %1'nin indirilmesine çalışıldı.

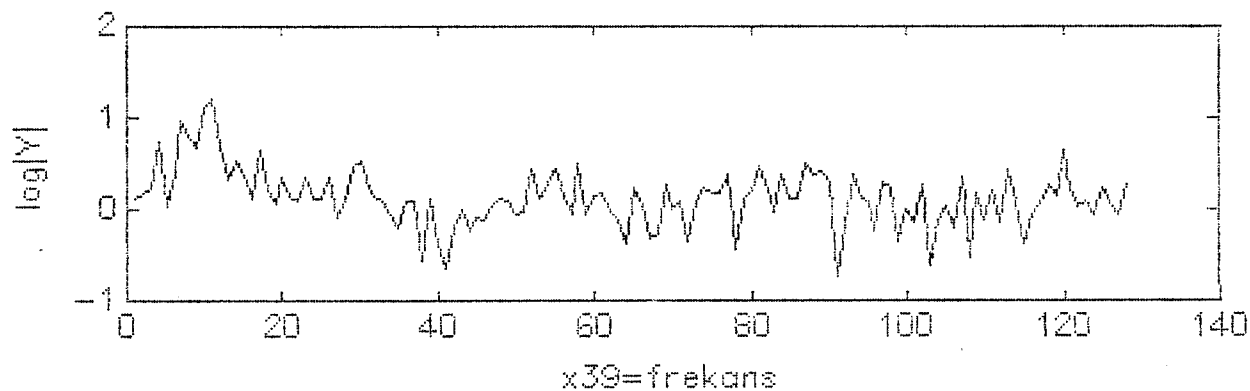
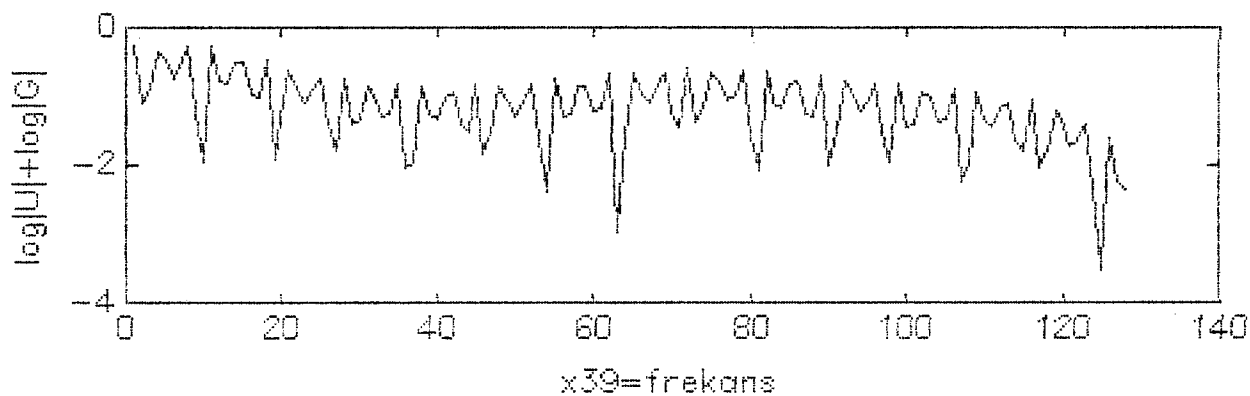
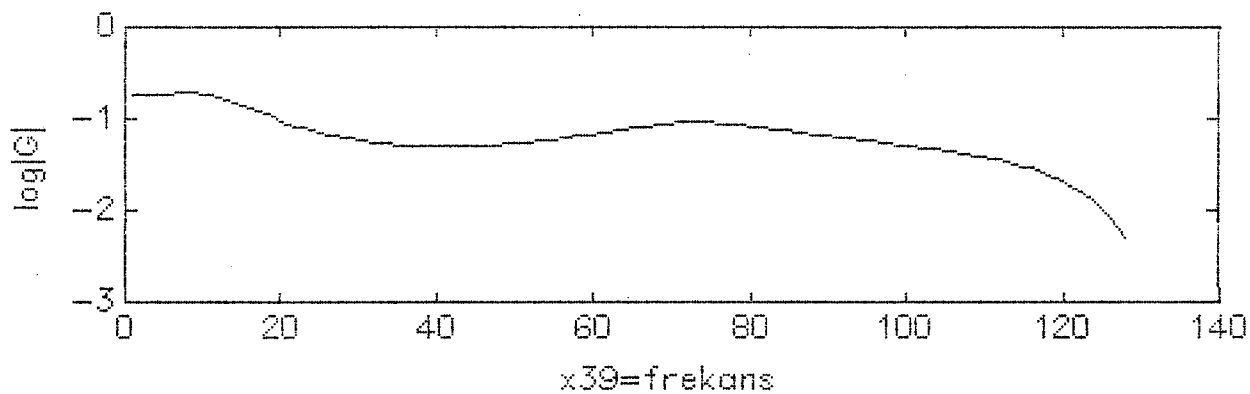
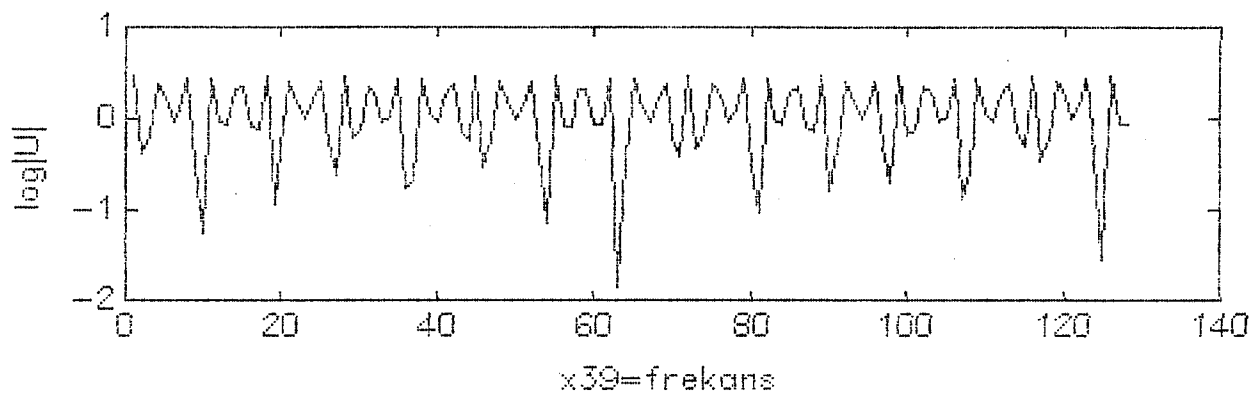
Fourier açılımının katsayılarının bir kutba veya sıfıra bağlı olması beklenirken, yani katsayıların sayısı kadar kutup ve sıfır elde edilmesi gerekirken (Mermelstein, 1967) öyle olmadığı gözlemlendi. Karşılaştırma yapabilmek için her iki yöntemden elde edilen kutup ve sıfırların sayılarının eşit olması gerektiğinden, farklı olduğu durumlarda transfer fonksiyonunun derecesi değiştirilerek kutup ve sıfırlar yeniden bulundu. Ayrıca, elde edilen kutup ve sıfırlar



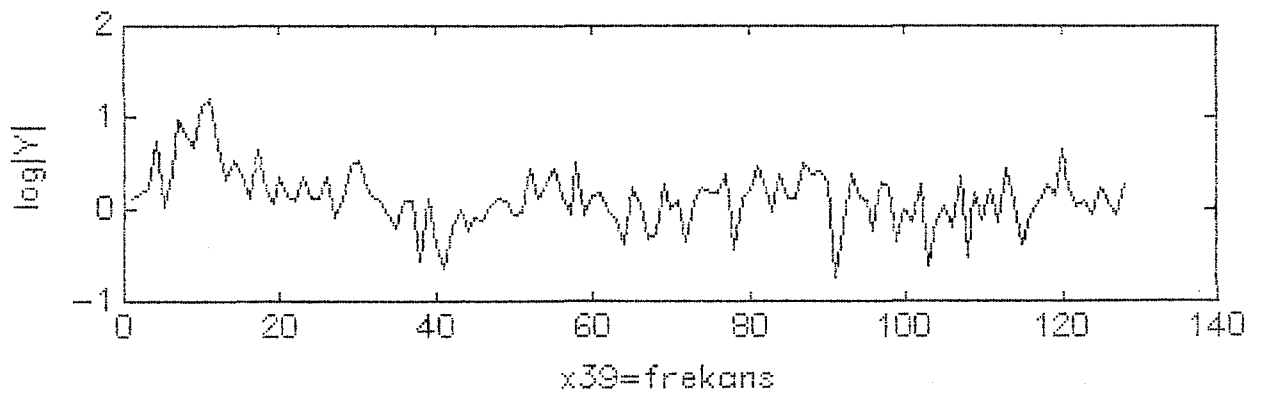
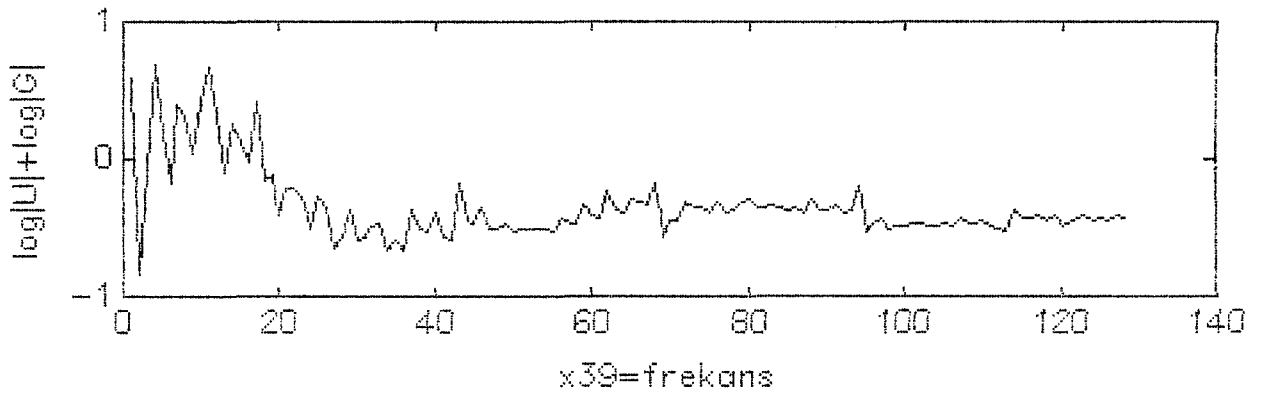
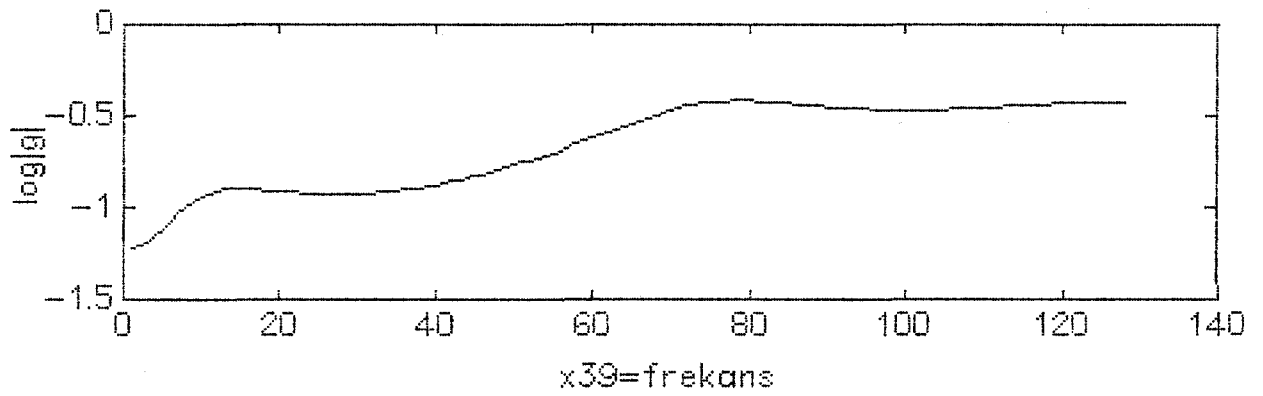
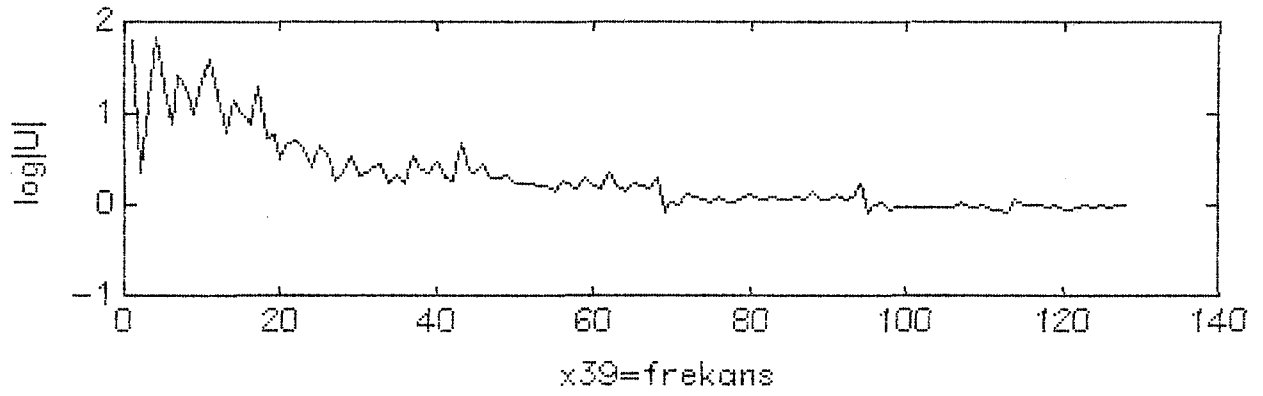
Sekil 5.5. Tüm kutup modelinde impuls girdi için elde edilen deęişimler.



Sekil 5.6. Tüm kutup modelinden ölçülen girdi için elde edilen değişimler.



Sekil 5.7. Kutup sıfır modelinden impuls girdi için elde edilen deęişimler.



Sekil 5.8. Kutup sıfır modelinden ölçülen girdi için elde edilen değişimler.

gerçek, sanal ve kompleks olabildiğinden sadece sanal kısımları alındı ve küçükten büyüğe doğru dizilerek işleme konuldu. Çünkü köklerin gerçek kısımları kayıpları göstermektedir. Dolayısıyla ses bölgesinin kayıpsız olduğu kabul edilerek, kutup-sıfır modeli üzerinde sekiz ünlü için yapılan çalışmaların sonuçları çizelgeler 5.1-5.8'de verilmiştir. Özellikle "e", "i", "u" "ı" ve "o" harflarındaki daralma noktası, yani dilin aldığı pozisyon oldukça belirgindir. Tahmin edildiği gibi kutup-sıfır modeli diğer modellere göre daha iyi sonuçlar vermiştir. Beklenen kutup ve sıfırlardan farklı değerler elde edildiğinde, alan fonksiyonunun katsayılarının ne şekilde değiştirilmesi gerektiği bilinmediğinden, her bir kutup ve sıfır için iki olasılık olmak üzere toplam dört olasılık denendi. Kutup değerini arttırmak için ona bağlı katsayının azaltılması gerektiği, sıfır değeri için de bunun ters olması gerektiği sonucuna varıldı. Katsayıların değiştirilme işlemi 100 iterasyona tabi tutulmasına rağmen elde edilen bütün kutup ve sıfırlar, beklenen bütün kutup ve sıfırların %1'ne indirilemedi. Dolayısıyla verilen değerler 100 iterasyon sonucunda elde edilen değerlerdir. Eğer daha fazla iterasyon kullanılarak beklenen kutup ve sıfırlara istenilen sınır dahilinde yaklaşılsa daha iyi sonuçlar elde edilebilir. Ayrıca kutup-sıfır modelinde, daha önceki sonuçlara dayanılarak sadece ölçülen girdi kullanılmıştır.

Sekiz ünlü harf için ses bölgesinin transfer fonksiyonunun kutup katsayılarından elde edilen kesikli ve sürekli alan fonksiyonları ayrı ayrı ve üst üste olmak üzere şekiller 5.9-5.16'da gösterilmiştir. Ayrıca 20 kişiden alınan sekiz ünlü için elde edilen enine kesit alanlarından her bir ünlü için ortalama değerler çizdirilmiş ve standard sapmaları ile beraber Ek-10'da verilmiştir.

İki farklı kişiden alınan dört aynı ünlü için ses bölgesinin transfer fonksiyonunun kutup katsayılarından elde edilen sürekli alan fonksiyonları ayrı ayrı ve üst üste olmak üzere şekiller 5.17-5.20'de gösterilmiştir.

Sekiz ünlü için ses bölgesinin aldığı şeklin insan kafası içindeki gösterimi şekiller 5.21-5.28'de verilmiştir.

Ses bölgesinin şeklinin silindirik bir tüp olduğu kabul edildiğinden enine kesit alanlarından yarıçaplar bulunarak şekiller çizdirildi.

Çizelge 5.1. "A" harfi için çeşitli modellerden elde edilen enine kesit alanlar (cm²).

Model	Durum	Enine Kesit Alanlar											Kazanç
Tüm Kutup	†	1	1.44	1.64	1.38	1.58	2.55	5.99	12.31	17.72	15.06	2.25	0.015
	††	1	1.42	1.6	1.34	1.51	2.41	5.62	11.43	16.19	13.55	2.05	-0.025
	†††	1	1.42	1.6	1.33	1.51	2.41	5.62	11.43	16.19	13.55	2.05	0.044
Kutup	e		0.65	0.47	2.23	5.39	9.18	20.15	17	8.78	3.13	0.52	
Sıfır	ee		0.92	1.22	1.2	0.86	1.64	3.28	4.9	2.66	0.68	0.26	
Durum Uzay	Kayıpsız (4 tüp)	1		2.26			5.97			3.28			

Çizelge 5.2. "E" harfi için çeşitli modellerden elde edilen enine kesit alanlar (cm²).

Model	Durum	Enine Kesit Alanlar											Kazanç
Tüm Kutup	†	1	2.31	4.42	6.75	10.6	11.54	10.69	11	8.71	8.36	2.67	0.021
	††	1	2.28	4.33	6.65	10.64	11.83	11.17	11.57	9.09	8.55	2.58	-0.067
	†††	1	2.28	4.32	6.64	10.61	11.79	11.13	11.52	9.06	8.52	2.58	0.074
Kutup	e		0.54	1.53	2.6	1.44	0.76	0.65	1.27	2.11	1.34	0.7	
Sıfır	ee		0.15	0.67	1.5	5.64	12.34	7.5	5.21	1.34	2.36	0.36	
Durum Uzay	Kayıpsız (4 tüp)	1		5.2			2.8			1.4			

† Ölçülen girdi.

†† t=0'da birim impals.

††† t=0'da, ölçülen girdi sinyalinin tepe değerine eşit genlikteki impals.

e Esitlik (4.37)'den elde edilen model.

ee Esitlik (4.38)'den elde edilen model.

Çizelge 5.3. "1" harfi için çeşitli modellerden elde edilen enine kesit alanlar (cm²).

Model	Durum	Enine Kesit Alanlar											Kazanç
Tüm Kutup	†	1	1.18	1.68	2.77	3.12	3.33	2.63	1.49	0.97	0.65	0.29	0.023
	**	1	1.17	1.68	2.8	3.22	3.49	2.77	1.56	0.98	0.63	0.25	0.041
	***	1	1.17	1.68	2.8	3.21	3.49	2.77	1.56	0.98	0.63	0.25	-0.096
Kutup	e		0.42	1.26	3.34	5.38	6.26	5	2.75	0.88	0.3	0.1	
Sıfır	@@		0.26	0.83	1.5	8.5	15.75	12	4.78	1.55	0.8	0.15	
Durum Uzay	Kayıpsız (4 tüp)	1	4.06				5.6			1.43			

Çizelge 5.4. "0" harfi için çeşitli modellerden elde edilen enine kesit alanlar (cm²).

Model	Durum	Enine Kesit Alanlar											Kazanç
Tüm Kutup	†	1	0.95	1.06	1.44	2.29	3.38	5.44	6.7	7.03	3.85	0.59	-0.075
	**	1	0.88	0.89	1.13	1.8	2.85	5.45	8.76	13.04	8.88	0.69	-0.047
	***	1	0.88	0.9	1.15	1.82	2.89	5.49	8.87	13.04	8.88	0.71	-0.016
Kutup	e		0.59	1.28	3.65	7.26	8.98	7.1	2.6	0.6	0.18	0.36	
Sıfır	@@		0.75	1.35	2.2	2.98	5.2	7.8	9.6	4.2	1.3	2.1	
Durum Uzay	Kayıpsız (4 tüp)	1	4.11				7.33			4.98			

† Ölçülen girdi.

** t=0'da birim impals.

*** t=0'da, ölçülen girdi sinyalinin tepe değerine eşit genlikteki impals.

e Eşitlik (4.37)'den elde edilen model.

@@ Eşitlik (4.38)'den elde edilen model.

Çizelge 5.5. "I" harfi için çeşitli modellerden elde edilen enine kesit alanlar (cm²).

Model	Durum	Enine Kesit Alanlar											Kazanç	
Tüm Kutup	*	1	1.48	1.25	2	3.48	4.08	6.08	6.34	4.75	2.4	0.58	0.019	
	**	1	1.47	1.24	2	3.58	4.37	6.88	7.66	5.98	3.01	0.63	-0.009	
	***	1	1.46	1.23	1.99	3.55	4.33	6.83	7.64	5.97	3	0.63	-0.036	
Kutup	e		0.36	0.15	1.23	2.47	3.2	8.57	5.2	1.22	0.15	0.51		
Sıfır	@e		1.1	1.68	2.4	3.7	1	0.26	0.85	1.36	4.74	2.2		
Durum Uzay	Kayıpsız (4 tüp)	1			3.4			6.8			2.4			

Çizelge 5.6. "U" harfi için çeşitli modellerden elde edilen enine kesit alanlar (cm²).

Model	Durum	Enine Kesit Alanlar											Kazanç	
Tüm Kutup	*	1	1.18	1.23	1.7	1.71	1.82	1.93	1.45	0.76	0.46	0.17	-0.058	
	**	1	1.17	1.25	1.8	1.94	2.19	2.54	2.07	1.14	0.73	0.28	0.044	
	***	1	1.17	1.25	1.8	1.94	2.19	2.54	2.07	1.15	0.73	0.28	-0.058	
Kutup	e		1.1	1.55	1.63	1.2	0.95	1.34	2.1	1.45	0.55	0.2		
Sıfır	@e		0.53	1.33	1.16	0.98	1.98	3.36	1.53	1.25	0.85	0.3		
Durum Uzay	Kayıpsız (4 tüp)	1			2.76			1.17			3.28			

* Ölçülen girdi.

** t=0'da birim impals.

*** t=0'da, ölçülen girdi sinyalinin tepe değerine eşit genlikteki impals.

e Eşitlik (4.37)'den elde edilen model.

@e Eşitlik (4.38)'den elde edilen model.

Cizelge 5.7. "Ü" harfi için çeşitli modellerden elde edilen enine kesit alanlar (cm²).

Model	Durum	Enine Kesit Alanlar											Kazanc
Tüm Kutup	‡	1	1.34	2.32	3.25	3.88	5.41	5.73	5.96	4.07	2.39	0.7	0.0028
	**	1	1.34	2.31	3.24	3.91	5.5	5.89	6.15	4.19	2.44	0.7	-0.044
	***	1	1.34	2.31	3.24	3.91	5.5	5.88	6.15	4.2	2.44	0.7	0.0368
Kutup	@		1.1	1.55	1.63	1.2	0.95	1.34	2.1	1.45	0.55	0.2	
Sıfır	@@		0.55	0.32	0.76	2.45	11.45	18.2	8.39	2.8	0.9	0.4	
Durum Uzay	Kayıpsız (4 tüp)	1			3.36			2.5			1.27		

Cizelge 5.8. "Ü" harfi için çeşitli modellerden elde edilen enine kesit alanlar (cm²).

Model	Durum	Enine Kesit Alanlar											Kazanc
Tüm Kutup	‡	1	1.73	2.73	3.78	4.83	4.93	4.72	3.39	1.87	1.04	0.33	0.024
	**	1	1.72	2.8	4.13	5.81	6.61	7.04	5.49	3.1	1.65	0.41	-0.007
	***	1	1.72	2.8	4.12	5.81	6.6	7.04	5.49	3.1	1.65	0.41	0.0137
Kutup	@		0.52	1.2	2.41	1.6	0.76	1.85	1.67	2.58	1.27	0.37	
Sıfır	@@		0.25	0.56	1	4.06	11.95	13.33	7.8	3.28	0.76	0.21	
Durum Uzay	Kayıpsız (4 tüp)	1			3.1			5.7			2		

‡ Ölçülen girdi.

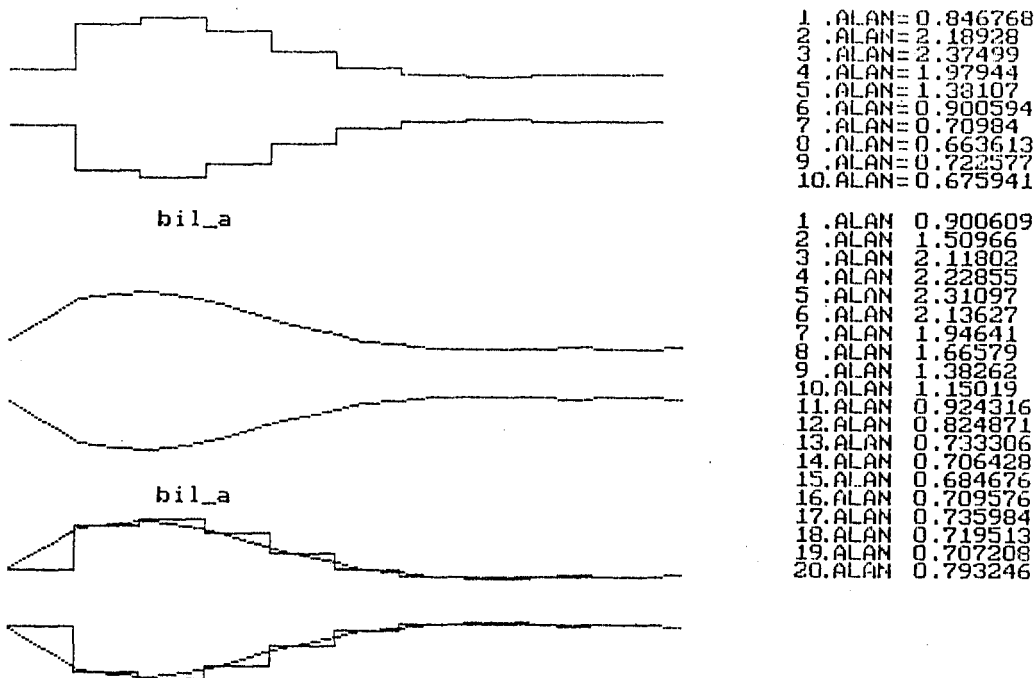
** t=0'da birim impals.

*** t=0'da, ölçülen girdi sinyalinin tepe değerine eşit genlikteki impals.

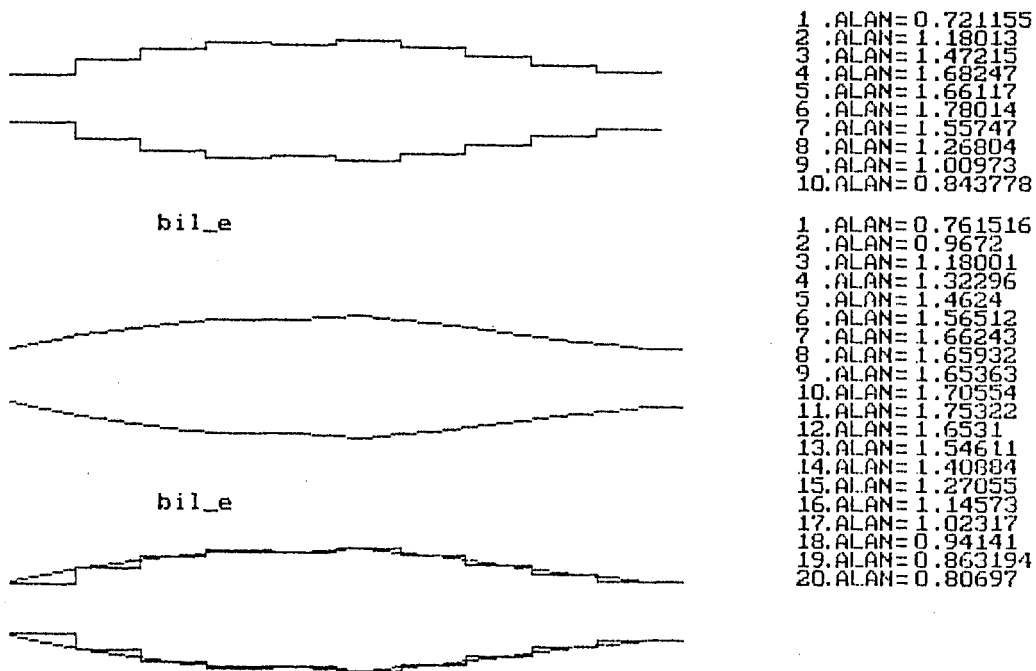
@ Eşitlik (4.36)'dan elde edilen model.

@ Eşitlik (4.37)'dan elde edilen model.

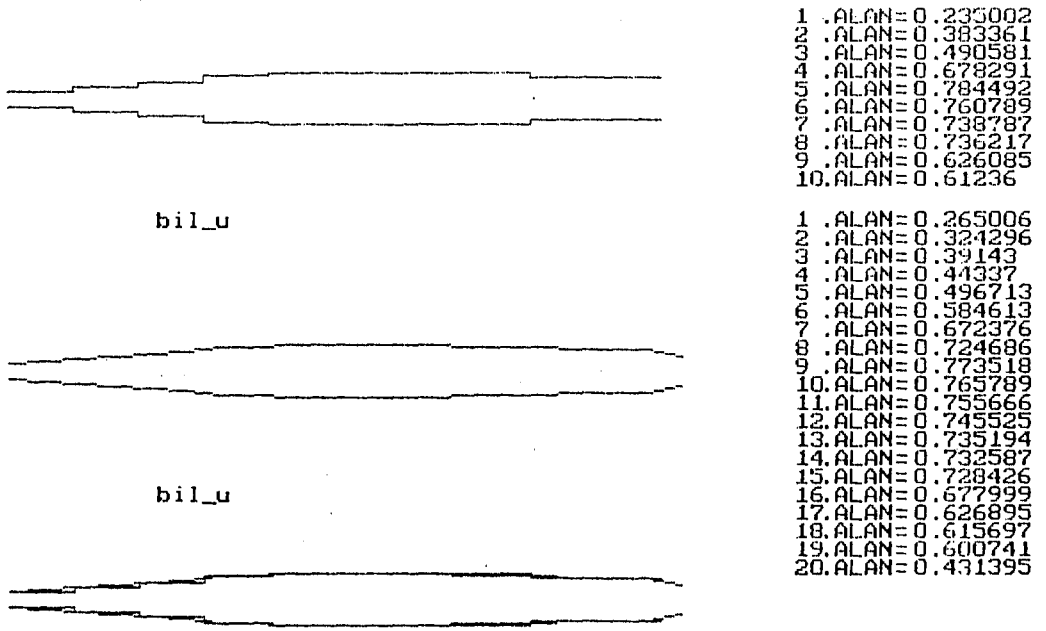
@@ Eşitlik (4.38)'den elde edilen model.



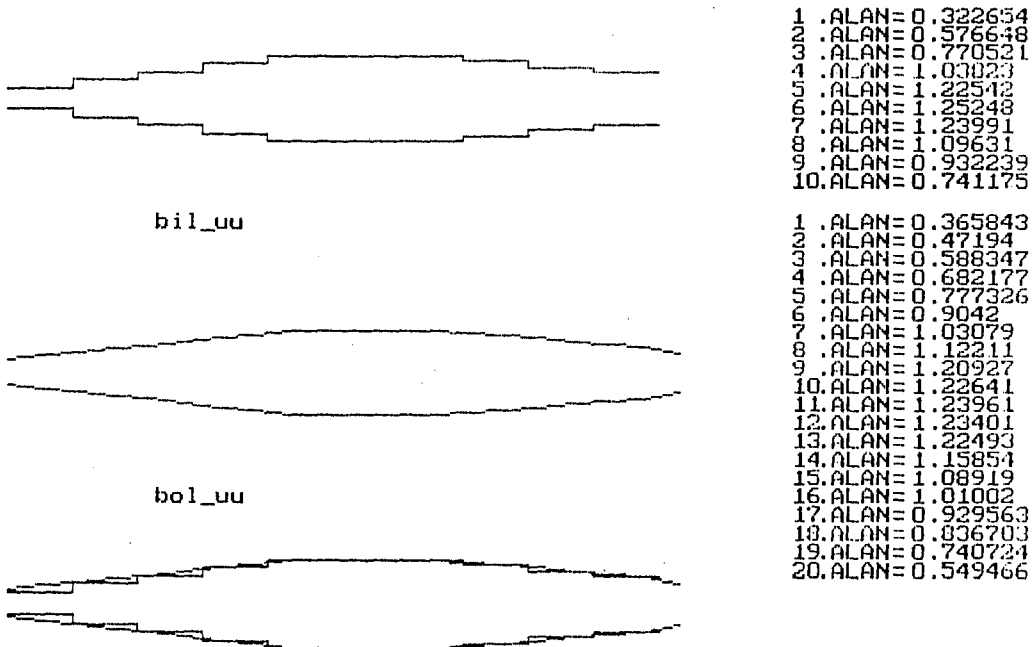
Şekil 5.9. "A" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması



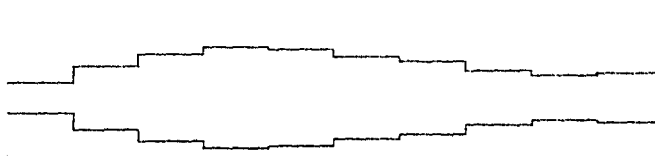
Şekil 5.10. "E" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması



Şekil 5.11. "U" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması



Şekil 5.12. "U" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması



bil_i



bil_i



1 .ALAN=0.42999
 2 .ALAN=0.873165
 3 .ALAN=1.22905
 4 .ALAN=1.42102
 5 .ALAN=1.39072
 6 .ALAN=1.13951
 7 .ALAN=1.05187
 8 .ALAN=0.798336
 9 .ALAN=0.630354
 10 .ALAN=0.6862

1 .ALAN=0.471893
 2 .ALAN=0.668671
 3 .ALAN=0.875106
 4 .ALAN=1.04593
 5 .ALAN=1.21315
 6 .ALAN=1.30829
 7 .ALAN=1.3964
 8 .ALAN=1.38652
 9 .ALAN=1.36882
 10 .ALAN=1.25484
 11 .ALAN=1.13897
 12 .ALAN=1.09176
 13 .ALAN=1.04415
 14 .ALAN=0.925854
 15 .ALAN=0.806768
 16 .ALAN=0.724994
 17 .ALAN=0.648046
 18 .ALAN=0.66745
 19 .ALAN=0.686904
 20 .ALAN=0.575886

Şekil 5.13. "I" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması



bil_ii



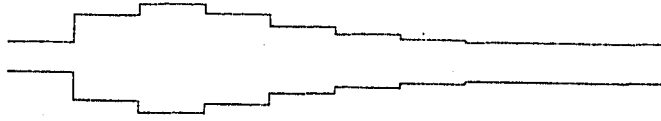
bil_ii



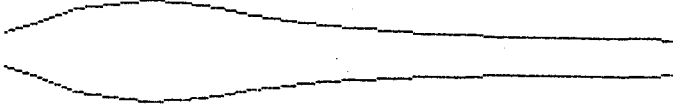
1 .ALAN=0.304894
 2 .ALAN=0.456281
 3 .ALAN=0.554864
 4 .ALAN=0.689693
 5 .ALAN=0.915656
 6 .ALAN=1.02985
 7 .ALAN=0.996943
 8 .ALAN=0.939321
 9 .ALAN=0.73212
 10 .ALAN=0.611861

1 .ALAN=0.334419
 2 .ALAN=0.396125
 3 .ALAN=0.464677
 4 .ALAN=0.512791
 5 .ALAN=0.561653
 6 .ALAN=0.6267
 7 .ALAN=0.693767
 8 .ALAN=0.799794
 9 .ALAN=0.905211
 10 .ALAN=0.961926
 11 .ALAN=1.01419
 12 .ALAN=1.002
 13 .ALAN=0.986296
 14 .ALAN=0.958327
 15 .ALAN=0.927069
 16 .ALAN=0.8311
 17 .ALAN=0.733722
 18 .ALAN=0.67267
 19 .ALAN=0.610494
 20 .ALAN=0.470055

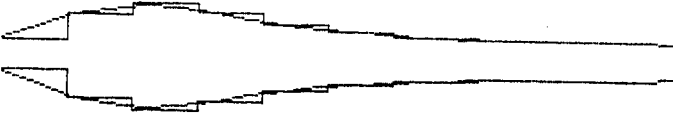
Şekil 5.14. "I" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması



bil_o



bil_o



1 .ALAN= 0.434376
 2 .ALAN= 1.24156
 3 .ALAN= 1.3547
 4 .ALAN= 1.31587
 5 .ALAN= 0.959229
 6 .ALAN= 0.750856
 7 .ALAN= 0.659407
 8 .ALAN= 0.574339
 9 .ALAN= 0.552579
 10 .ALAN= 0.550203

1 .ALAN= 0.480767
 2 .ALAN= 0.843515
 3 .ALAN= 1.21228
 4 .ALAN= 1.36883
 5 .ALAN= 1.50887
 6 .ALAN= 1.40793
 7 .ALAN= 1.29506
 8 .ALAN= 1.12896
 9 .ALAN= 0.962285
 10 .ALAN= 0.859113
 11 .ALAN= 0.759989
 12 .ALAN= 0.712026
 13 .ALAN= 0.666544
 14 .ALAN= 0.624897
 15 .ALAN= 0.584912
 16 .ALAN= 0.572783
 17 .ALAN= 0.562428
 18 .ALAN= 0.561109
 19 .ALAN= 0.559985
 20 .ALAN= 0.513942

Şekil 5.15. "0" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması



bil_oo



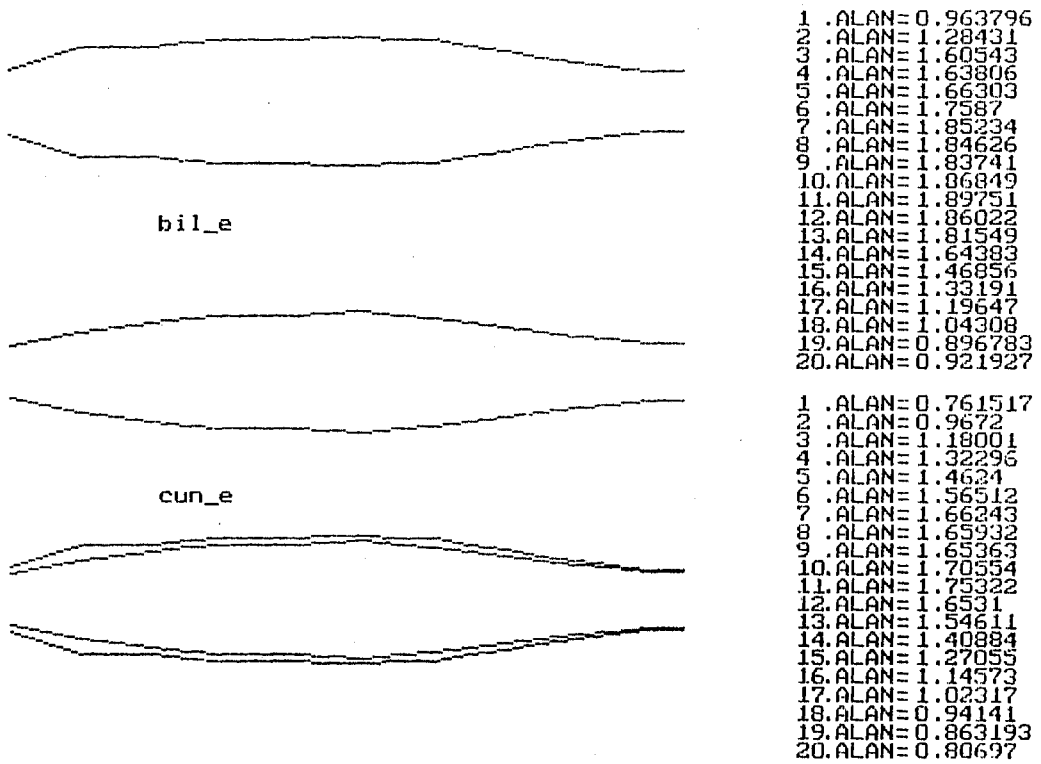
bil_oo



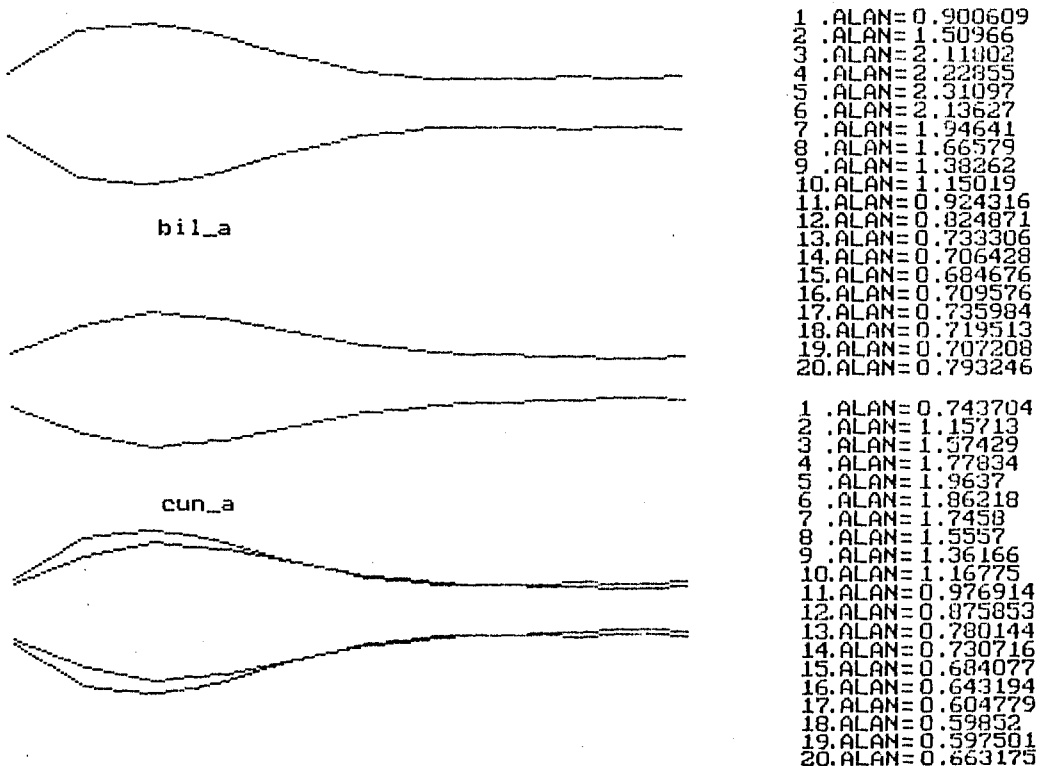
1 .ALAN= 0.447496
 2 .ALAN= 0.677277
 3 .ALAN= 0.924943
 4 .ALAN= 1.15745
 5 .ALAN= 1.21114
 6 .ALAN= 1.147
 7 .ALAN= 1.06314
 8 .ALAN= 0.882904
 9 .ALAN= 0.791699
 10 .ALAN= 0.638181

1 .ALAN= 0.47752
 2 .ALAN= 0.57834
 3 .ALAN= 0.686585
 4 .ALAN= 0.804378
 5 .ALAN= 0.922653
 6 .ALAN= 1.03384
 7 .ALAN= 1.14163
 8 .ALAN= 1.17139
 9 .ALAN= 1.1959
 10 .ALAN= 1.16829
 11 .ALAN= 1.13769
 12 .ALAN= 1.09732
 13 .ALAN= 1.05469
 14 .ALAN= 0.970451
 15 .ALAN= 0.885839
 16 .ALAN= 0.838979
 17 .ALAN= 0.792512
 18 .ALAN= 0.719687
 19 .ALAN= 0.6462
 20 .ALAN= 0.558531

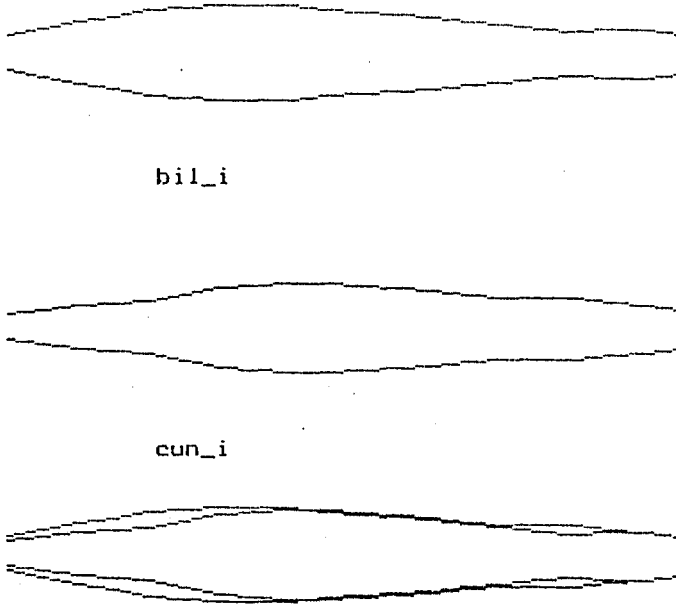
Şekil 5.16. "0" harfi için kesikli ve sürekli alan fonksiyonlarının karşılaştırması



Sekil 5.18. İki farklı kişiden alınan "E" harfinin karşılaştırması



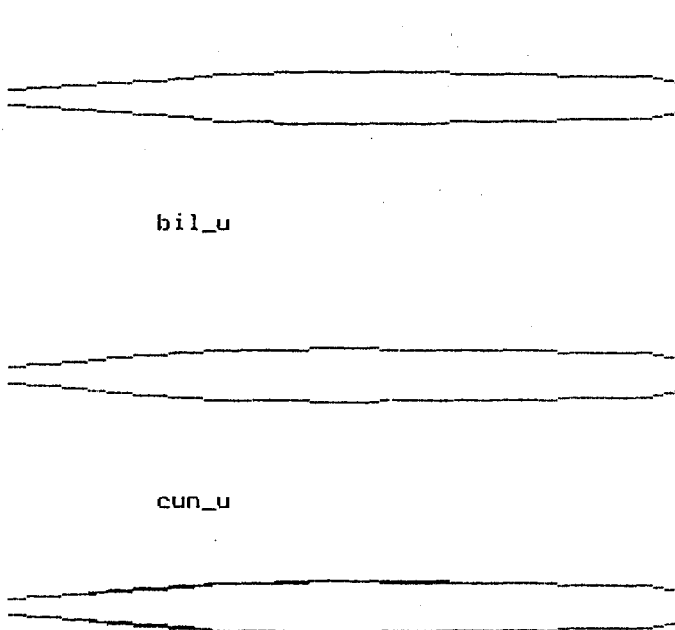
Sekil 5.17. İki farklı kişiden alınan "A" harfinin karşılaştırması



1 .ALAN= 0.471893
 2 .ALAN= 0.668671
 3 .ALAN= 0.875106
 4 .ALAN= 1.04593
 5 .ALAN= 1.21315
 6 .ALAN= 1.30829
 7 .ALAN= 1.3964
 8 .ALAN= 1.38652
 9 .ALAN= 1.36882
 10 .ALAN= 1.25484
 11 .ALAN= 1.13897
 12 .ALAN= 1.09176
 13 .ALAN= 1.04415
 14 .ALAN= 0.925854
 15 .ALAN= 0.806768
 16 .ALAN= 0.724994
 17 .ALAN= 0.648046
 18 .ALAN= 0.66745
 19 .ALAN= 0.686904
 20 .ALAN= 0.575886

1 .ALAN= 0.369587
 2 .ALAN= 0.483288
 3 .ALAN= 0.606891
 4 .ALAN= 0.697336
 5 .ALAN= 0.790622
 6 .ALAN= 0.974549
 7 .ALAN= 1.15732
 8 .ALAN= 1.22789
 9 .ALAN= 1.29081
 10 .ALAN= 1.26875
 11 .ALAN= 1.24091
 12 .ALAN= 1.16333
 13 .ALAN= 1.08308
 14 .ALAN= 0.992095
 15 .ALAN= 0.902913
 16 .ALAN= 0.885644
 17 .ALAN= 0.868847
 18 .ALAN= 0.799081
 19 .ALAN= 0.724541
 20 .ALAN= 0.543548

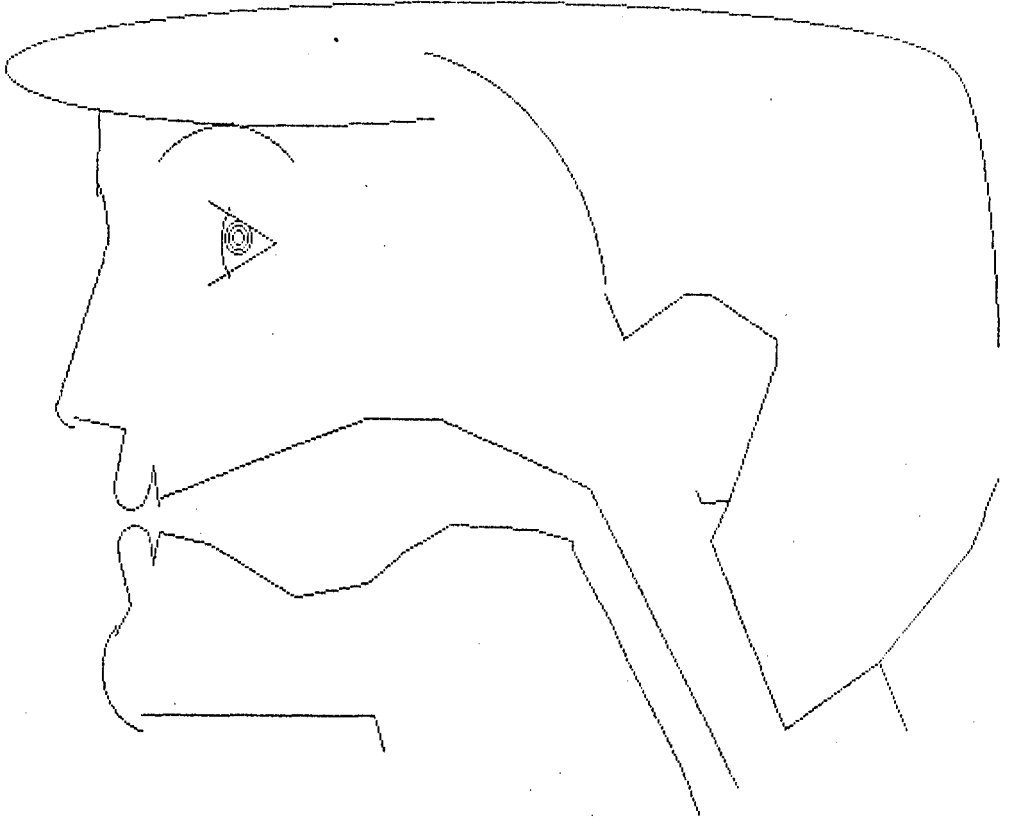
Sekil 5.19. İki farklı kişiden alınan "i" harfinin karşılaştırması



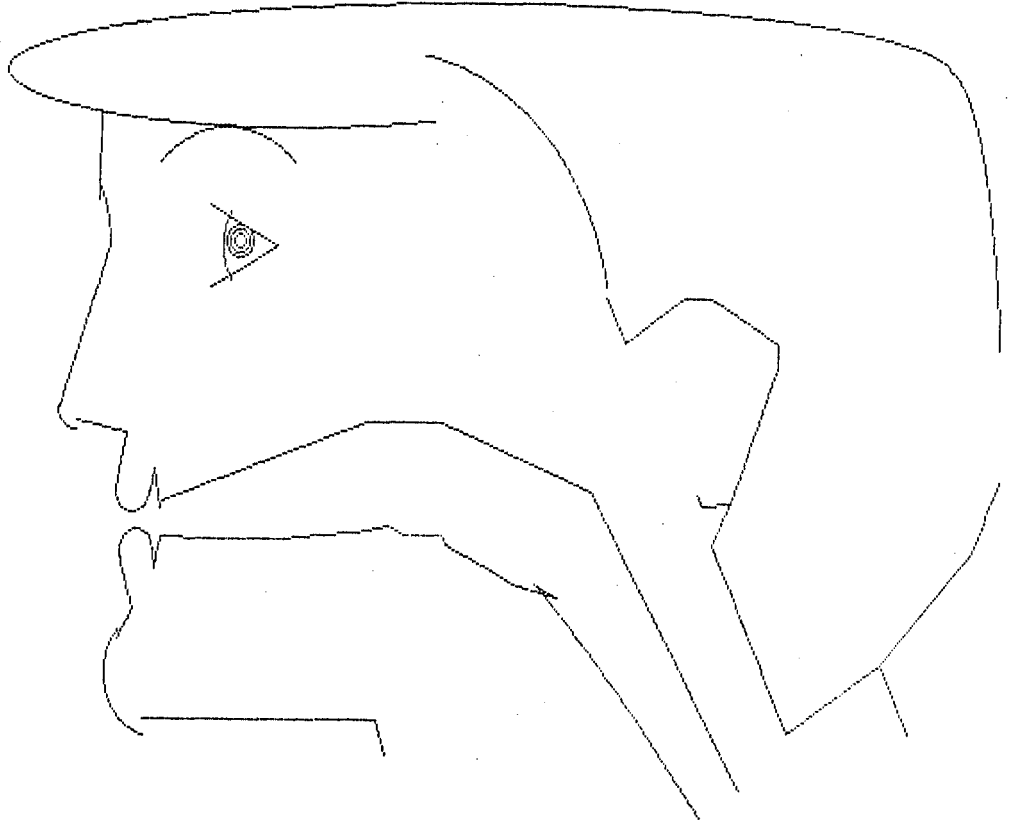
1 .ALAN= 0.265006
 2 .ALAN= 0.324296
 3 .ALAN= 0.39143
 4 .ALAN= 0.44337
 5 .ALAN= 0.496713
 6 .ALAN= 0.584613
 7 .ALAN= 0.672376
 8 .ALAN= 0.724686
 9 .ALAN= 0.773518
 10 .ALAN= 0.763789
 11 .ALAN= 0.755666
 12 .ALAN= 0.745525
 13 .ALAN= 0.735194
 14 .ALAN= 0.732587
 15 .ALAN= 0.728426
 16 .ALAN= 0.677999
 17 .ALAN= 0.626895
 18 .ALAN= 0.615697
 19 .ALAN= 0.600741
 20 .ALAN= 0.431395

1 .ALAN= 0.248373
 2 .ALAN= 0.319139
 3 .ALAN= 0.398333
 4 .ALAN= 0.477097
 5 .ALAN= 0.553434
 6 .ALAN= 0.618047
 7 .ALAN= 0.673847
 8 .ALAN= 0.706675
 9 .ALAN= 0.732363
 10 .ALAN= 0.73965
 11 .ALAN= 0.745311
 12 .ALAN= 0.732937
 13 .ALAN= 0.719813
 14 .ALAN= 0.712855
 15 .ALAN= 0.704879
 16 .ALAN= 0.668321
 17 .ALAN= 0.631157
 18 .ALAN= 0.618445
 19 .ALAN= 0.601259
 20 .ALAN= 0.423097

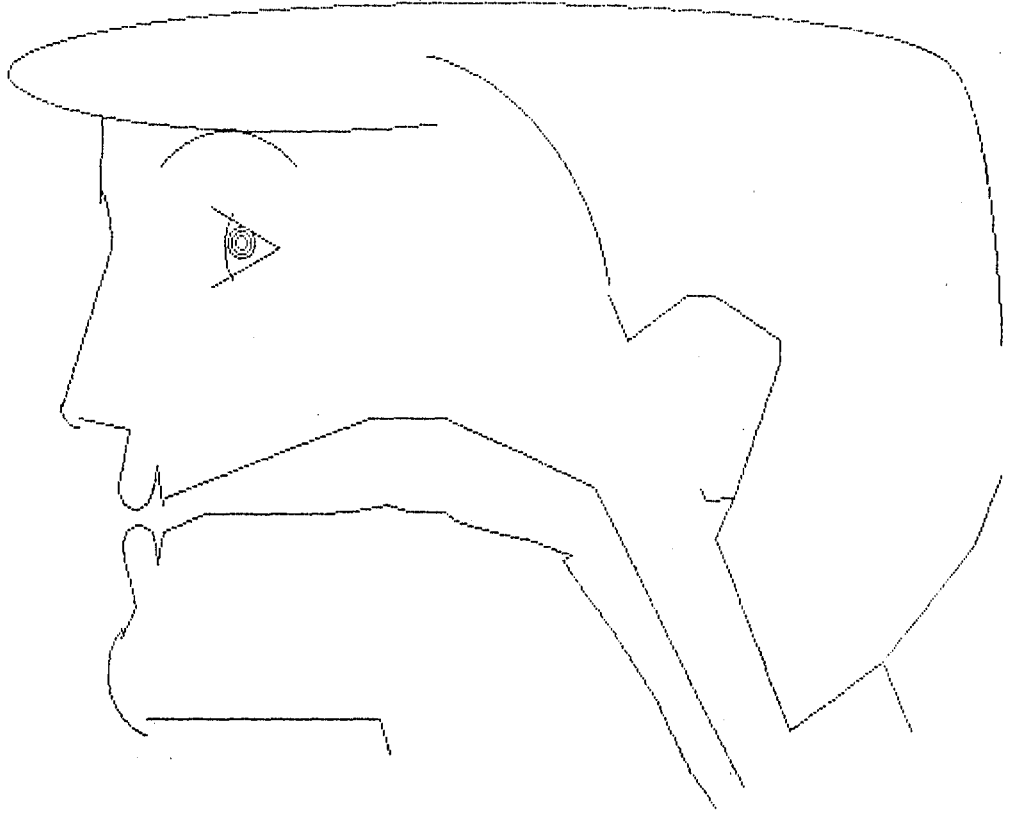
Sekil 5.20. İki farklı kişiden alınan "u" harfinin karşılaştırması



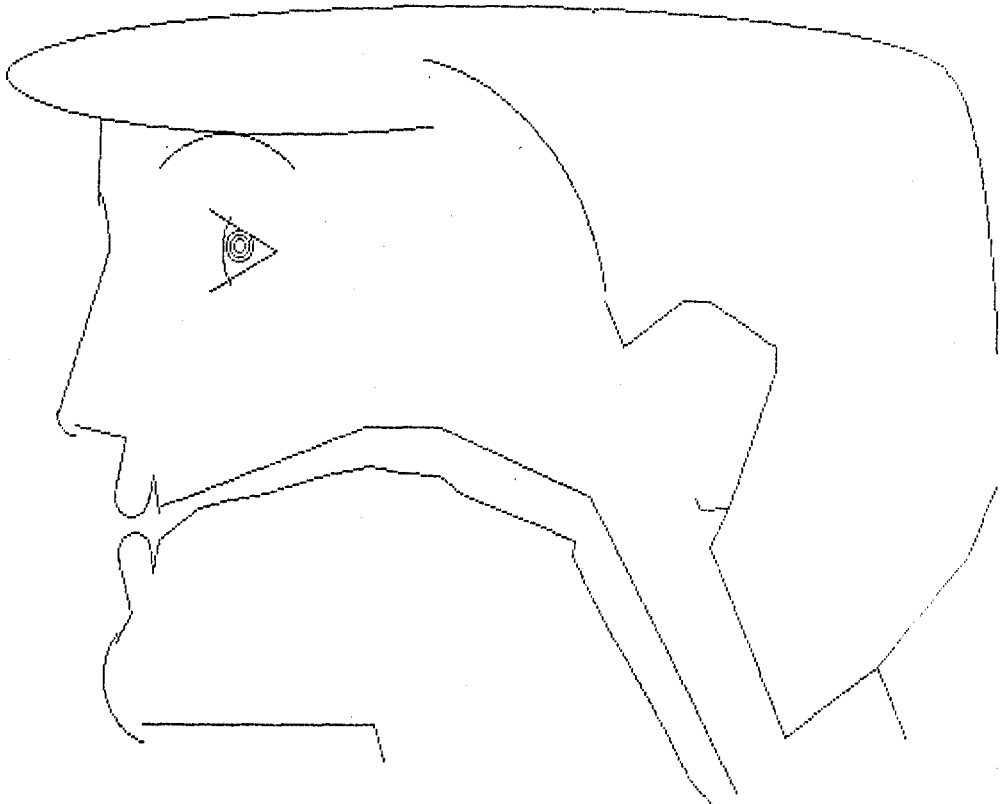
Şekil 5.21."A" harfi için ses bölgesinin gösterimi



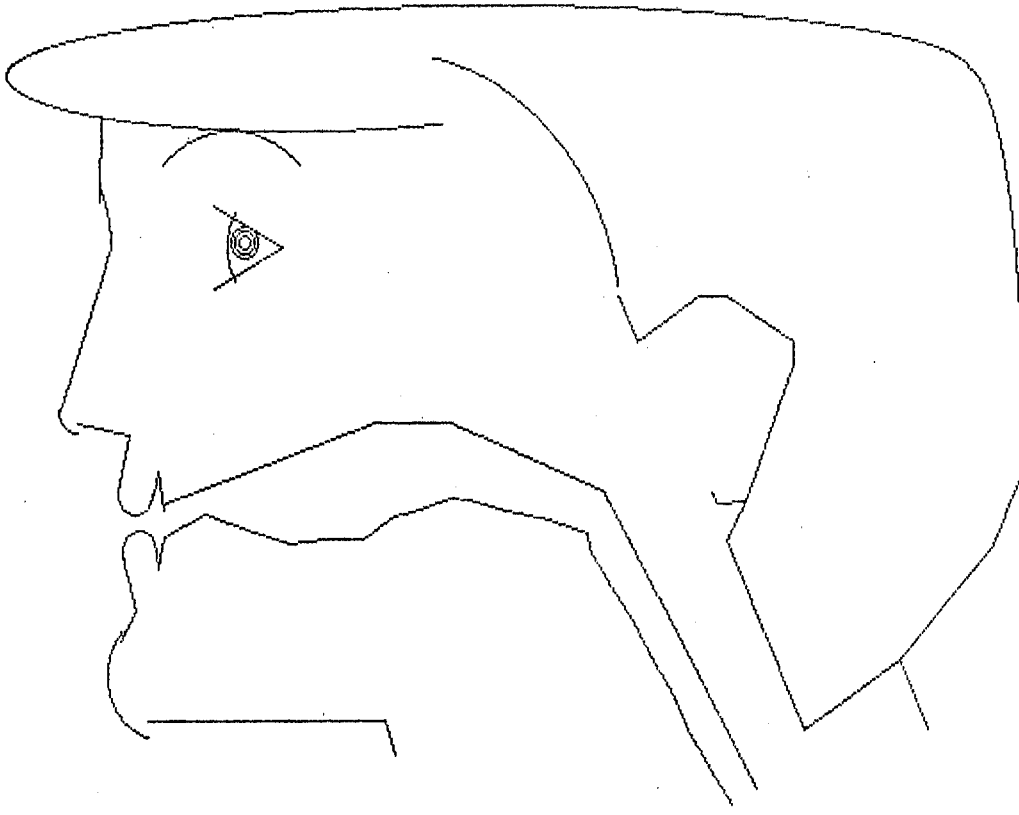
Şekil 5.22."E" harfi için ses bölgesinin gösterimi



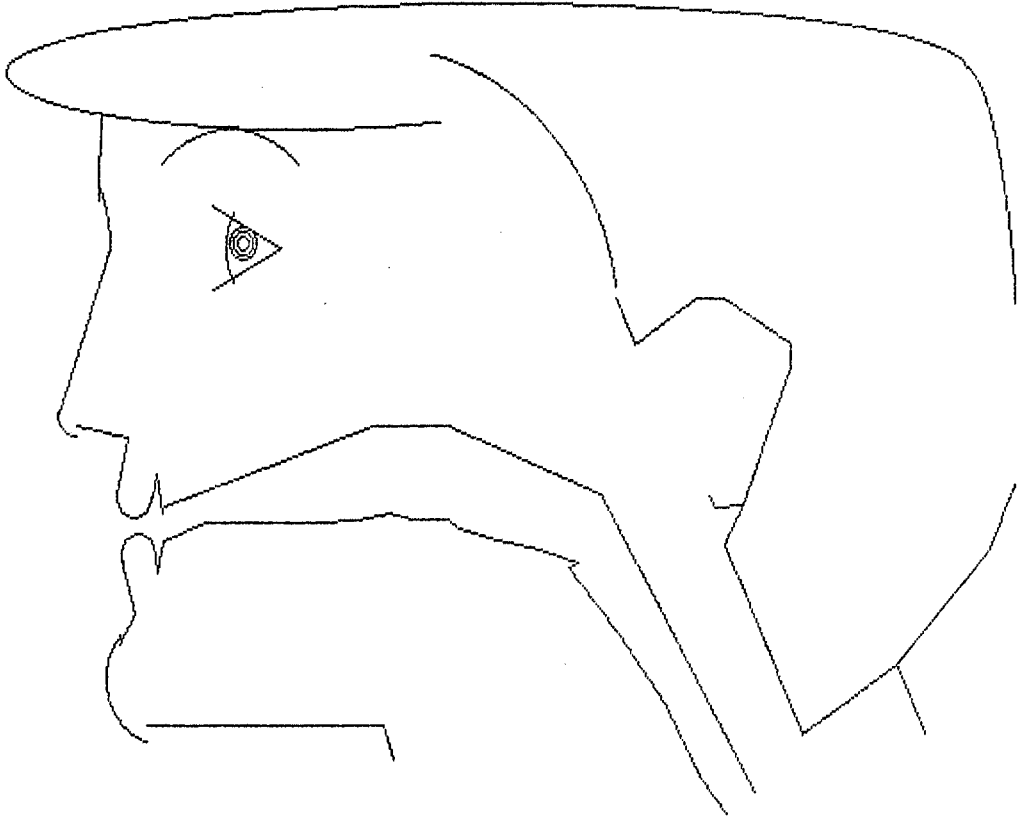
Şekil 5.23."I" harfi için ses bölgesinin gösterimi.



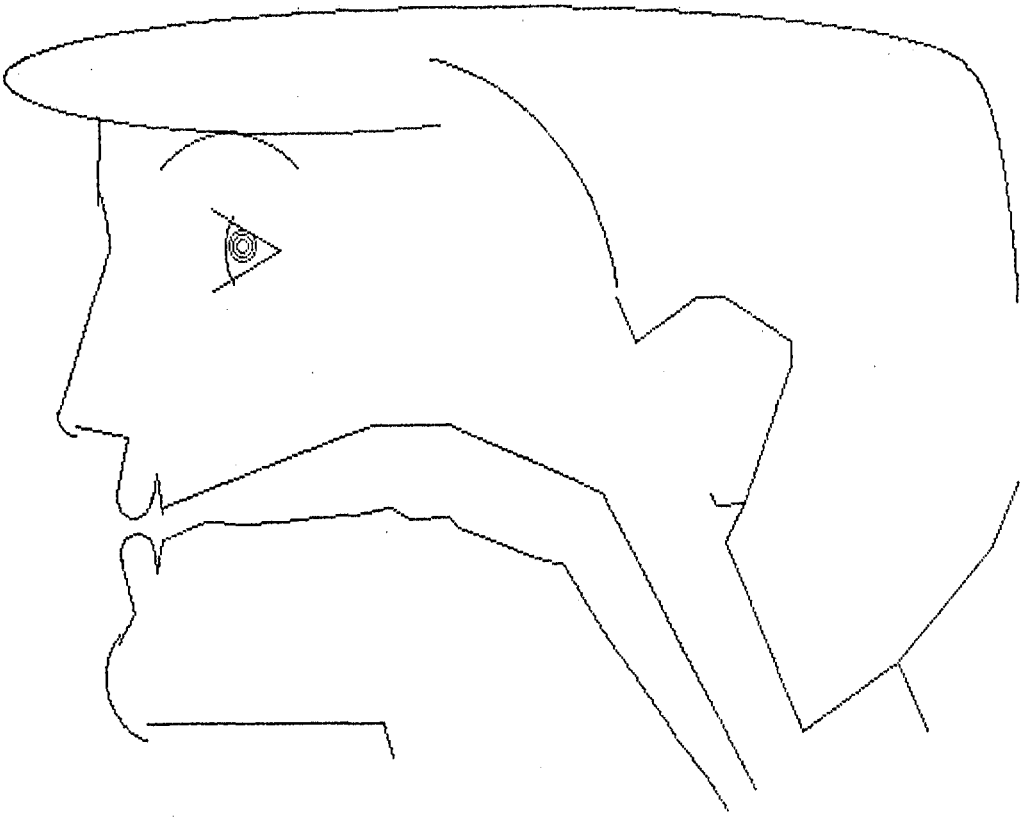
Şekil 5.24."U" harfi için ses bölgesinin gösterimi.



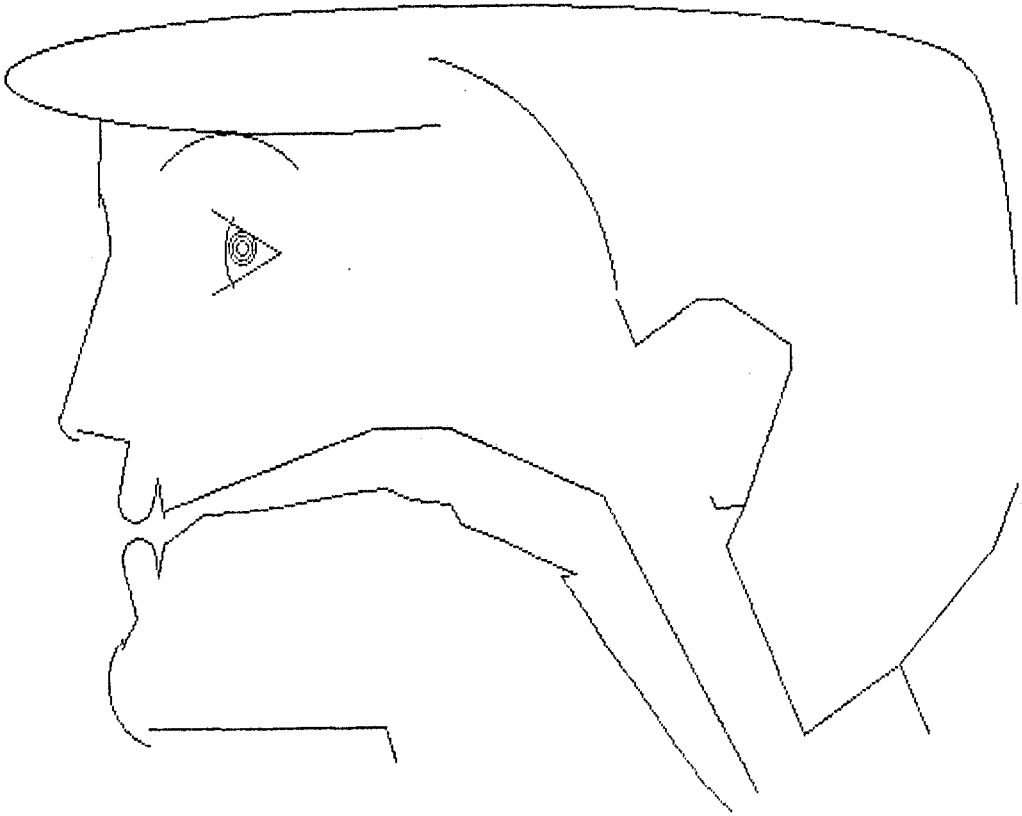
Sekil 5.25."O" harfi için ses bölgesinin gösterimi



Sekil 5.26."I" harfi için ses bölgesinin gösterimi



Şekil 5.27."d" harfi için ses bölgesinin gösterimi



Şekil 5.28."ü" harfi için ses bölgesinin gösterimi

BÖLÜM 6

SONUÇLAR VE ÖNERİLER

Bu çalışmada kullanılan ve, gırtlaktan ve dudaktan alınan sinyalleri örnekleyen ADDA kartı, aynı anda iki sinyalin 50 μ s'de örneklemesine olanak vermemiştir. Alt bölüm 4.5.5'de açıklandığı gibi en iyi değerler iki sinyalin ancak 100 μ s'de örneklenmesi ile elde edilmiştir. Fakat ses sinyalinin osilasyon frekansı girdi sinyaline göre yüksek olduğundan 4 defa ses sinyalinden 1 defa girdi sinyalinden değer alınacak şekilde iki sinyal örneklenmiştir. Arada kalan boşluklar ise interpolasyon ile doldurulmuştur. Dolayısıyla bu şekilde elde edilen örneklenmiş değerler gerçek değerlerden bir miktar sapma göstermektedir. Bu da sonuçlar üzerinde etkili olabilir. Eğer girdi ve çıktı sinyalleri daha hassas olarak aynı anda 50 μ s'de örneklenebilirse sonuçlar daha iyi olabilir. Ayrıca ses sinyali ile gırtlak sinyali arasında, ses bölgesinin uzunluğundan dolayı 500 μ s'lik bir faz kayması olmaktadır. Bu faz kaymasının sistemin doğal bir özelliği olduğu düşünülerek, örneklenen değerler kaydırılmayıp olduğu gibi alınmıştır.

Tüm kutup modeli üzerinde yapılan çalışmalar göstermiştir ki, çeşitli girdiler için enine kesit alanları pek değişmemektedir. Fakat ölçülen girdi kullanıldığında kazancın pozitif olması, impuls girdi yerine ölçülen girdi kullanılmasının daha geçerli olacağını göstermektedir. Ayrıca ölçülen girdinin gerçek girdiye uygunluğu daha ayrıntılı bir çalışma ile incelenerek gerçek girdi elde edilebilirse, tüm kutup modeli daha iyi sonuçlar verebilir. Yapılan çalışmalar sonucu, impuls girdinin yerinin önemli olmadığı, fakat genliğinin ölçülen girdiye göre belirlenmesinin daha uygun olacağı görülmektedir.

Kutup-sıfır modeli, özellikle dilin daralma yerini daha iyi göstermesi bakımından avantajlı olmakla beraber, enine kesit alanları uzun bir iterasyon sonucu elde edildiğinden kullanıcıya kısa sürede bilgi vermemesi bakımından da dezavantaja sahiptir. Fakat bu model sonucu elde edilen değerler insan kafası içine yerleştirilerek dosyalanırsa istenildiği zaman kullanıcıya göstermek suretiyle yararlı olabilir. Ayrıca, beklenen değerlere ulaşıldığında programın bitmesini sağlanacak şekilde iterasyon sayısı artırılırsa daha iyi sonuçlar elde edilir.

Sonuçta, kullanıcıya hemen bir fikir verebilmek için, ölçülen girdinin kullanılarak tüm kutup modelinin seçilmesi daha uygundur. Girdinin piezo kristal ile ölçülmesi zahmetinden kurtulmak istenirse

$t=0$ 'da birim impals girdi uygulamak suretiyle de tüm kutup modelinden elde edilen sonuçlar kullanıcıya verilebilir. Bu şekildeki girdi, kutup-sıfır modeline de uygulanmak suretiyle elde edilen sonuçlar karşılaştırılırsa, farklı girdilerin sonuçlara olan etkileri daha belirgin olarak görülebilir.

Durum-uzay modeli ve tüm kutup modeli üzerindeki yapılan çalışmalarda gırtlaktaki birinci bölümün enine kesit alanı 1 cm^2 olarak alınmıştır. Kutup-sıfır modelinde ise böyle bir kabullenme gerekmediğinden bu modelden elde edilen sonuçlar diğer kaynaklarda verilenlere daha yakın olmaktadır.

Ses bölgesinin şeklinin silindirik bir tüp olduğu kabul edildiğinden elde edilen enine kesit alanlarından yarıçaplar bulunarak insan kafası içine çizdirildi. Gerçekte ses bölgesinin enine kesiti alındığında eliptik şekiller elde edilmektedir. Elips şeklindeki enine kesit alanlar için analizler yapılırsa dilin daralma yeri daha belirgin bir hal alabilir ve bu da gerçeğe ne kadar yaklaşıldığını gösterir.

Sekiz ünlü için elde edilen enine kesit alanları, ve kaynaklarda verilen yabancı dildeki ünlüler ile karşılaştırıldığından, bu çalışmanın bundan sonra Türkçe'deki ünlüler için yapılacak çalışmalara ışık tutacağı beklenmektedir. Aslında Türkçe'deki ünlüler için de, Maeda (1982) tarafından yabancı dil için yapıldığı gibi X-ışınları kullanılarak ses bölgesinin belirlenmesi bu konudaki çalışmalara destek olacaktır.

Bu çalışmanın sonucunda, ünsüzler, ünlü-ünsüzler, ünsüz-ünlüler hatta hece ve kelimeler üzerinde böyle çalışmaların yapılması gerektiği anlaşılmıştır.

KAYNAKLAR DİZİNİ

- Aktop, G., 1988, Türkçedeki sert sessiz harflerin incelenmesi, Bitirme ödevi, Anadolu Üniversitesi Mühendislik Mimarlık Fakültesi, 70 s.
- Aström, K.J. and Eykhoff, P., 1971, System identification_A survey, *Automatica*, 7, 123-162.
- Budak, A., 1974, Passive and active network analysis and synthesis, Houghton Mifflin Company, Boston, 733 p.
- Carre, R. and Mrayati, M., 1989, Articulatory acoustic phonetic relations_Regions and modes, Proceedings of a NATO Advanced Study Institute, W.J. Hardcastle and A. Marchal (Eds.) Kluwer Academic Publishers, Chapter for "Speech Production and Speech Modelling".
- Carstens Medizinelektronik GmbH, 1991, Visualize, analyse and document results from the Articulograph AG100, Göttingen, 18 p.
- Fallside, F. and Woods, W.A., 1985, Computer speech processing, Prentice_Hall International Ltd, 505 p.
- Fant, G., 1970, Acoustic theory of speech production, Mouton and Co. N.V., Publishers, The Hague, 328 p.
- Flanagan, J.L., 1972, Speech analysis synthesis and perception, Springer_Verlag, 444 p.
- Fry, D.B., 1979, The physics of speech, Cambridge University Press, Cambridge, 148 p.
- Jacquot, R.G., 1981, Modern digital control systems, Marcel Dekker, Inc., New York, 355 p.
- Ladefoged, P., 1962, Elements of acoustic phonetics, University of Chicago Press, 120 p.
- Lin, Q. and Fant, G., 1990, A new algorithm for speech synthesis based on vocal tract modelling, 120_th Meeting of the Acoustical Society of America, 45-52.
- Ljung, L., 1987, System identification, Prentice-Hall, Inc., New Jersey, 519 p.
- Maeda, S., 1979, An articulatory model of the tongue based on a statistical analysis, Meeting of Acoustical Society of America, Cambridge, 67-69.
- Maeda, S., 1982, A digital simulation method of the vocal tract system, *Speech Communication I*, North-Holland Publishing Company, 199-229.
- Maeda, S., 1985, On the generation of sound in stop consonants, 14 emes Journées d'Etudes sur la Parole, Paris, 1-13.
- Maeda, S., 1987, Articulatory-acoustic relationships in unvoiced stops: A simulation study, 11_th ICPHS. Tallinn, USSR, 11-14.
- Makhoul, I.J. and Wolf, J.J., 1972, linear prediction and spectral analysis of speech, Virginia, 236 p.
- Markel, J.D. and Gray, A.H., 1976, Linear prediction of speech, Springer Verlag, New York, 288 p.

KAYNAKLAR DİZİNİ (devam ediyor)

- Mermelstein, P., 1967, Determination of the vocal tract shape from measured formant frequencies, *The Journal of The Acoustical Society of America*, 41, 5, 1283-1294.
- Miyanaga, Y., Miki, N. and Nagai, N., 1986, Adaptive identification of a time varying ARMA speech model, *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-34, 3, 423-433.
- Morikawa, H. and Fujisaki, H., 1982, Adaptive analysis of speech based on a pole-zero representation, *IEEE Transactions on Acoustics, speech and signal processing*, ASSP-30, 1, 77-87.
- Morikawa, H. and Fujisaki, H., 1984, System identification of the speech production process based on a state-space representation, *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-32, 2, 252-262.
- Morikawa, H., 1990, Adaptive estimation of time-varying model order in the ARMA speech analysis, *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38, 7, 1073-1083.
- Papamichalis, P.E., 1987, *Practical approach to speech coding*, Prentice-Hall, Inc., New Jersey, 322 p.
- Perkell, J.S., Testing theories of speech production : Implications of some detailed analysis of variable articulatory data, *Proceedings of a NATO Advanced Study Institute*, W.J. Hardcastle and A. Marchal (Eds.) Kluwer Academic Publishers, Chapter for "Speech Production and Speech Modelling".
- Teager, H.M. and Teager, S.M., 1989, Evidence for nonlinear production mechanism in the vocal tract, *Proceedings of a NATO Advanced Study Institute*, W.J. Hardcastle and A. Marchal (Eds.) Kluwer Academic Publishers, Chapter for "Speech Production and Speech Modelling".
- Wakita, H., 1973, Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms, *IEEE Transactions on Audio Electroacoust.*, AU-21, 417-427.
- Yannakoudakis, E.J. and Hutton, P.J., 1987, *Speech synthesis and recognition systems*, Ellis Harwood Limited, 184 p.

SES YOLUNUN
MATEMATİK MODELİ

Mehmet Bilginer Gülmezoglu

EKLER

- Ek.1. Tavır Nitelikleri.
- Ek.2. Konum Nitelikleri.
- Ek.3. Durum Uzay Modelinin Kayıpsız Durumu İçin Programlar.
- Ek.4. Durum Uzay Modelinin Kayıplı Durumu İçin Programlar.
- Ek.5. Bilgisayara Ses Verilerinin Alınmasında Kullanılan Devreler.
- Ek.6. Tüm Kutup Modelinden Enine Kesit Alanları Bulan Program.
- Ek.7. Eşitlik (4.37)'ye Dayanan Kutup Sıfır Modelinden Enine Kesit Alanları Bulan Program.
- Ek.8. Eşitlik (4.38)'e Dayanan Kutup Sıfır Modelinden Enine Kesit Alanları Bulan Program.
- Ek.9. Enine Kesit Alanları İnsan Kafasına Yerleştiren Program.
- Ek.10.20 Kişiden Alınan 8 Ünlü İçin Elde Edilen Enine Kesit Alanlarının Ortalaması.

EK 1

TAVIR NİTELİKLERİ

Nitelikler	Üretim Karakteristikleri	Ses Dalgası Karakteristikleri
Ses Kaynağı	Ses telleri titreşimlerinin bir hava akımı ile modülasyonu	Frekans ve zamanda hemen hemen periyodik yapıdadır.
Gürültü kaynağı	Dar geçitler içinden dalgalı hava akışı ve keskin engellerden geçiş.	Frekans ve zamanda rastgele yapıdadır.
Geçici kaynak	Bir aşırı basıncın kurulması tarafından sebep olunmuş ses boşlukları tek darbe tahriki.	Formant frekanslarında, kısa bir enerji azalması tarafından oluşmuş dar bandlı spektrumda dikey engelden (spike) dolayı tek süreksizlik.
Sessizlik	Ses yolunda kaynak hareketinin olmaması.	Spektral enerji yoktur.
Ünlü benzeri	Yutak ve ağızdan geçen havanın serbestliği.	Formant modeli tam olarak görülür. Eger kaynak sesli ise F_1 ve F_2 izlenebilir ve kaynak gürültülü ise F_2 ve F_3 izlenebilir.
Yanal (lateral)	Ağızda yanal açılma ile merkezi kapayan kısmının birleşmesi ve ağıza ait kapanmanın bitmesi.	F_2 - F_4 bölgesinde bir yankılanma tarafından sesli harf benzeri modelinin değişikliği ve takip eden yana ait olmayan vokalik bölümde F_1 'in daha yüksek pozisyona değişmesi.
Kapanma (Occlusi-ve)	Yutakta yada ağızda tam kapanma. Kapanmanın bitmesi	Genizden olmayan kapanma sesinin spektrumu, çok düşük frekanslı F_1 tarafından hakimdir. Daha yüksek formantlar genellikle görülmez. Sessiz kapanma enerji olmadığını gösterir. Tek süreksizlik F_1 geçişi yükselir.
Genizsel (Nasal)	Genize bağlı yumuşak damak ile ses alanının hareketsizliği.	Genizden gelen mırıltı spektrum tarafından karakterize edilir. F_2 zayıftır yada yoktur. Yüksek frekanslardaki formantlar daha düşüktür. Ünlü harf spektrumunda üst üste

		gelme bozukluđu olur. F_1 'de eşzamanlı azalma ve kayma olur. F_2 yada F_3 'ün bağıntılı azalması sık sık gürülür.
F,v,s,z gibisürtme sesi çıkarılan sürtümlü (fricative) harfler	Yutak yada ağzın kapanan bir bölümünde hava akımının çok dar bir geçitten geçmesi.	Sürtümlü seslerde F_4 daha belirgindir, fakat F_1 daha az şiddetli ve daha düşüktür. Üretilmiş bir sürtümlü ile gırtlak dışındaki gürültü kaynağı, spektrumda yüksek frekanslı gürültü alanıyla kabul edilir. Gürültü sesi tipik bir sürtümlüdür.
Geçişsel (Transiti-onal)	Ağız içindeki yüksek hızlı telaffuz hareketleri süzülme kısmındaki orta hızdaki telaffuz hareketleri olarak tanımlanabilir.	Bir ses geçiş bölümü (örneğin vokalik kısmın başlangıcı) kapanma kısmını takip eder. Sonunda formant geçişinin asıl bölümü tamamlanır. Mümkün olan süzülme nispeten spektrum değişimleriyle karakterize edilebilir. Süzülme (glide) j,l,r seslerinden meydana gelir.

EK 2

KONUM NİTELİKLERİ

Telaffuz hareketleri	Ses dalgası (speech wave)
Dil önde iken a) Ön damakta b) Orta damakta	$F_2 - F_1$ bandı geçiş F_2 yüksek, F_3 çok yüksek F_2 çok yüksek, F_3 yok
Dil geri çekilmiş iken	$F_2 - F_1$ bandı küçük, F_1 daha yüksek
Ağız açıklığı dar iken	F_1 düşük
Dudaklar nispeten kapalı ve dışarı çıkıntılı (dudağın az açıklığı)	İleri dil kapanması her seferinde F_1 , F_2 ve F_3 ' ün düşmesine tek başına nedendir. Fakat belirli bir dil pozisyonuna bağımlı olarak değişir. F_3 ' de ki etkisi dilin ön damak pozisyonu durumunda belirgindir.
Geriye bükük durumu a) Diş yuvasına ait telaffuz hareketi b) Damağa ait telaffuz hareketi	F_4 düşük ve F_3 yok F_3 düşük ve F_2 yok
Dudaksız kapanmalar	Damaksız (palatal) dil pozisyonunda F_2 yüksektir. Sürtünme (f) sesinin gürültü spektrumu aslında düzdür ve düşük şiddetlidir.
T ve d gibi dişsel ünsüzlerin telaffuzu	Ağırlık merkezi sürtünmeli (f)'den daha büyüktür.
Dişsel yada öndişlere ait telaffuz hareketleri.	F_2 orta ve F_3 yüksektir. Sürtünme yüksek frekanslıdır ve gürültü yüksektir. Düşük frekans limiti öndişlerde dişselden daha düşüktür.
a) Damaksız geriye bükük telaffuz b) Damaksız geriye bükük telaffuz ile dilin aşağı yana yatması.	F_3 düşük. Sürtünme gürültüsünün şiddeti yüksektir. F_2 ve F_3 yüksek. Gürültü merkezi F_3 , F_4 ve F_5 ' de görülür. "Ç" deki gürültü alanı, gerite bükük seslerden daha yüksek frekanslıdır.
Yumuşak damaksız ve gırtlaksız telaffuz hareketleri.	F_2 orta yada düşük. Sürtünme gürültüsünün geniş bir bölümü F_2 tarafından oluşur. F_1 hariç F-modeli net olarak görülebilir.

EK 3

DURUM UZAY MODELİNİN KAYIPSIZ DURUMU İÇİN PROGRAMLAR

(4 tüp için)

```

IMPLICIT REAL*8(A-H,O-Z)
EXTERNAL FCN
REAL*8 L
DIMENSION X(3),FVEC(3),WA(1500)
COMMON/PASIT/A,B,A0,L,W(7)
C
PI=4.DO*ATAN(1.DO)
C
DO 111 i=1,3
PRINT*, 'ENTER X(',i,') VALUE '
111 READ*, X(i)
L=17.DO
C DO 112 i=1,7,2
C PRINT*, 'ENTER W(',i,') VALUE '
C112 READ*, W(i)
C W(i)=2.DO*PI*W(i)
TOL = 1.E-2
N = 3
LWA = 1500
a = 10.0*0.00114*0.35**2
b = 10.0/0.00114
A0 = 1
W(1) = 2.DO*PI*0.00240
W(3) = 2.DO*PI*0.01631
W(5) = 2.DO*PI*0.02112
W(7) = 2.DO*PI*0.02640
C
C ----- SOLVE NON-LINEAR SYSTEM -----
C
CALL HYBRD1(FCN,N,X,FVEC,TOL,INFO,WA,LWA)
C
PRINT*, 'INFO = ',INFO
PRINT*,
PRINT*,
PRINT*, ' COZUM VE HATA VEKTORU '
DO 4 I=1,N
PRINT*, X(I),FVEC(I)

```

```

4   CONTINUE
    PRINT*,
    STOP
    END

C
    SUBROUTINE FCN(N,X,FVEC)
    IMPLICIT REAL*8(A-H,O-Z)
    REAL*8 L
    COMMON/PASIT/A,B,A0,L,W(7)
    DIMENSION X(N),FVEC(N)

C
    a1 = X(1)
    a2 = X(2)
    a3 = X(3)

C
    w1 = W(1)
    w3 = W(3)
    w5 = W(5)
    w7 = W(7)

C
    FVEC(1)=a*(A0*b/(a1*1**2)+a1*b/(a2*1**2)+a2*b/(a3*1**2)+4*b/
    $1**2)-(w1**2+w3**2+w5**2+w7**2)

C
    FVEC(2)=a**2*(A0*b**2*(a1*a3+a2**2+2*a2*a3)/(a1*a2*a3*1**4)+
    $b**2*(a1*(a2+2*a3)+2*a2*(a2+3*a3))/(a2*a3*1**4))-(w1**2
    $*(w3**2+w5**2+w7**2)+w3**2*(w5**2+w7**2)+w5**2*w7**2)

C
    FVEC(3)=a**3*(A0*b**3*(a1*(a2+a3)+a2*a3)/(a1*a2*a3*1**6)+b**3*
    $(a1*(a2+a3)+a2*(a2+4*a3))/(a2*a3*1**6))-(w1**2*(w3**2*
    $*(w5**2+w7**2)+w5**2*w7**2)+w3**2*w5**2*w7**2)

C
    RETURN
    END

```

(10 tıp için)

```

IMPLICIT REAL*8(A-H,O-Z)
EXTERNAL FCN
REAL*8 L
DIMENSION X(9),FVEC(9),WA(1500)

```

```

COMMON/PASIT/A,B,AO,L,W(19)
C
C   PI=4.DO*ATAN(1.DO)
C
      DO 111 i=1,9
      PRINT*, 'ENTER X(',i,') VALUE '
111  READ*, X(i)
      L=17.DO
      DO 112 i=1,19,2
      PRINT*, 'ENTER W(',i,') VALUE '
112  READ*, W(i)
C 112  W(i)=2.DO*PI*W(i)
      TOL   = 1.E-4
      N     = 9
      LWA   = 1500
      a     = 10.0*0.00114*0.35**2
      b     = 10.0/0.00114
      AO    = 2
C   L=((a**10*b**10)/(W(1)**2*W(3)**2*W(5)**2*W(7)**2*W(9)**2*
C   $W(11)**2*W(13)**2*W(15)**2*W(17)**2*W(19)**2)**0.05
C   PRINT*, 'L = ',L
C
C ----- SOLVE NON-LINEAR SYSTEM -----
C
      CALL HYBRD1 (FCN,N,X,FVEC,TOL,INFO,WA,LWA)
C
      PRINT*, 'INFO = ',INFO
      PRINT*,
      PRINT*,
      PRINT*, ' COZUM VE HATA VEKTORU '
      DO 4 I=1,N
      PRINT*, X(I),FVEC(I)
4     CONTINUE
      PRINT*,
      STOP
      END
C
      SUBROUTINE FCN(N,X,FVEC)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 L
      COMMON/PASIT/A,B,AO,L,W(19)
      DIMENSION X(N),FVEC(N)
C
      A1 = X(1)

```

$$A2 = X(2)$$

$$A3 = X(3)$$

$$A4 = X(4)$$

$$A5 = X(5)$$

$$A6 = X(6)$$

$$A7 = X(7)$$

$$A8 = X(8)$$

$$A9 = X(9)$$

C

$$PAYDA=A1*A2*A3*A4*A5*A6*A7*A8*A9*L**20.$$

C

$$W1=W(1)$$

$$W3=W(3)$$

$$W5=W(5)$$

$$W7=W(7)$$

$$W9=W(9)$$

$$W11=W(11)$$

$$W13=W(13)$$

$$W15=W(15)$$

$$W17=W(17)$$

$$W19=W(19)$$

C

$$FVEC(1) = ((a*b*A3*A4*A5*A6*A7*A8*A9*(A0*A2+A1**2+3*A1*A2)*L**18+a*b*A1*A2*A3*(A9*(A3*A5*A6*A7*A8+A4*(A4*A6*A7*A8+A5*(A5*A7*A8+A6*(A6*A8+A7*(A7+7*A8)))))+A4*A5*A6*A7*A8**2)*L**18+a*b*A1*A2**2*A4*A5*A6*A7*A8*A9*L**18)/PAYDA - (W1**2+W3**2+W5**2+W7**2+W9**2+W11**2+W13**2+W15**2+W17**2+W19**2)$$

C

$$FVEC(2) = ((a**2*b**2*A3*A4*A5*A6*A7*A8*A9*(A0*(A1+A2)+A1**2+3*A1*A2)*L**16+a**2*b**2*A3*(A0*A2+A1**2+3*A1*A2)*A9*(A3*A5*A6*A7*A8+A4*(A4*A6*A7*A8+A5*(A5*A7*A8+A6*(A6*A8+A7*(A7+7*A8)))))+A4*A5*A6*A7*A8**2)*L**16+a**2*b**2*A1*A2*A3*(A9*(A3*(A4*A6*A7*A8+A5*(A5*A7*A8+A6*(A6*A8+A7*(A7+5*A8)))))+A4*(A4*(A5*A7*A8+A6*(A6*A8+A7*(A7+5*A8)))+A5*(A5*(A6*A8+A7*(A7+5*A8)))+A6*(A6*(A7+5*A8)+A7*(5*A7+21*A8$)))+A3*A5*A6*A7*A8**2+A4*A8*(A4*A6*A7*A8+A5*(A5*A7*A8+A6*(A6*A8+A7*(A7+5*A8)))))*L**16+a**2*b**2*A1*A2**2*(A9*(A3*A5*A6*A7*A8+A4*A8*(A4*A6*A7*A8+A5*(A5*A7*A8+A6*(A6*A8+A7*(A7+6*A8)))))+A4*A5*A6*A7*A8**2)*L**16+a**2*b**2*(A0*A2+A1**2+2*A1*A2)*A2*A4*A5*A6*A7*A8*A9*L**16)/PAYDA - (W1**2*(W3**2+W5**2+W7**2+W9**2+W11**2+W13**2)+W11**2*(W13**2+W3**2+W5**2+W7**2+W9**2)+W13**2*(W3**2+W5**2+W7**2+W9**2)+W3**2*(W5**2+W7**2+W9**2)+W5**2*(W7**2+W9**2)+W7**2*W9**2+(W1**2+W11**2+W13**2+W3**2+W5**2+W7**2+W9**2)*(W15**2+W17**2+W19**2)+$$

$$w_{15}^2 w_{17}^2 + w_{15}^2 w_{19}^2 + w_{17}^2 w_{19}^2$$

C

$$\begin{aligned} FVEC(3) = & ((a^3 b^3 A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8 A_9 L^{14} + \\ & a^3 b^3 \\ & A_3 (A_0 (A_1 + A_2) + A_1^2 + 3 A_1 A_2) (A_9 (A_3 A_5 A_6 A_7 A_8 + A_4 (A_4 A_6 A_7 A_8 \\ & + A_5 (A_5 A_7 A_8 + A_6 (A_6 A_8 + A_7 (A_7 + 7 A_8))))) + A_4 A_5 A_6 A_7 A_8^2) L^{14} + \\ & a^3 b^3 A_3 (A_0 A_2 + A_1^2 + 3 A_1 A_2) (A_9 (A_3 (A_4 A_6 A_7 A_8 + A_5 (A_5 A_7 \\ & A_8 + A_6 (A_6 A_8 + A_7 (A_7 + 5 A_8)))) + A_4 (A_4 (A_5 A_7 A_8 + A_6 (A_6 A_8 + A_7 (A_7 + 5 \\ & A_8))) + A_5 (A_5 (A_6 A_8 + A_7 (A_7 + 5 A_8)) + A_6 (A_6 (A_7 + 5 A_8) + A_7 (5 A_7 + 21 A_8) \\ &))) + A_3 A_5 A_6 A_7 A_8^2 + A_4 A_8 (A_4 A_6 A_7 A_8 + A_5 (A_5 A_7 A_8 + A_6 (A_6 A_8 + \\ & A_7 (A_7 + 5 A_8))))) L^{14} + a^3 b^3 A_1 A_2 A_3 (A_9 (A_3 (A_4 (A_5 A_7 A_8 + \\ & A_6 (A_6 A_8 + A_7 (A_7 + 4 A_8))) + A_5 (A_5 (A_6 A_8 + A_7 (A_7 + 3 A_8)) + A_6 (A_6 (A_7 + 3 \\ & A_8) + A_7 (3 A_7 + 10 A_8)))) + A_4 (A_4 (A_5 (A_6 A_8 + A_7 (A_7 + 4 A_8)) + A_6 (A_6 (A_7 + \\ & 3 A_8) + A_7 (3 A_7 + 10 A_8))) + A_5 (A_5 (A_6 (A_7 + 4 A_8) + A_7 (3 A_7 + 10 A_8)) + A_6 (\\ & 2 A_6 (2 A_7 + 5 A_8) + 5 A_7 (2 A_7 + 7 A_8))))) + A_8 A_3 (A_4 A_6 A_7 A_8 + A_5 (A_5 A_7 \\ & A_8 + A_6 (A_6 A_8 + A_7 (A_7 + 3 A_8)))) + A_4 A_8 (A_4 (A_5 A_7 A_8 + A_6 (A_6 A_8 + A_7 (A_7 \\ & + 3 A_8))) + A_5 (A_5 (A_6 A_8 + A_7 (A_7 + 3 A_8)) + A_6 (A_6 (A_7 + 3 A_8) + 2 A_7 (2 A_7 + 5 \\ & A_8))))) L^{14} + a^3 b^3 A_1 A_2^2 (A_9 (A_3 (A_4 A_6 A_7 A_8 + A_5 (A_5 A_7 A_8 \\ & + A_6 (A_6 A_8 + A_7 (A_7 + 5 A_8)))) + A_4 (A_4 (A_5 A_7 A_8 + A_6 (A_6 A_8 + A_7 (A_7 + 4 \\ & A_8))) + A_5 (A_5 (A_6 A_8 + A_7 (A_7 + 4 A_8)) + A_6 (A_6 (A_7 + 4 A_8) + A_7 (4 A_7 + 15 A_8) \\ &))) + A_3 A_5 A_6 A_7 A_8^2 + A_4 A_8 (A_4 A_6 A_7 A_8 + A_5 (A_5 A_7 A_8 + A_6 (A_6 A_8 + \\ & A_7 (A_7 + 4 A_8))))) L^{14} + a^3 b^3 A_2 (A_0 A_2 + A_1^2 + 2 A_1 A_2) (A_9 (A_3 \\ & A_5 A_6 A_7 A_8 + A_4 (A_4 A_6 A_7 A_8 + A_5 (A_5 A_7 A_8 + A_6 (A_6 A_8 + A_7 (A_7 + 6 A_8))) \\ &)) + A_4 A_5 A_6 A_7 A_8^2) L^{14} + a^3 b^3 (A_0 + A_1 + A_2) A_1 A_2 A_4 A_5 A_6 A_7 \\ & A_8 A_9 L^{14}) / PAYDA - (w_1^2 (w_{11}^2 (w_{13}^2 + w_3^2) + \\ & w_5^2 + w_7^2 + w_9 \\ &)^2 + w_{13}^2 (w_3^2 + w_5^2 + w_7^2 + w_9^2) + w_3^2 (w_5^2 + w_7^2 + w_9^2) + w_5 \\ &)^2 (w_7^2 + w_9^2) + w_7^2 w_9^2) + w_{11}^2 (w_{13}^2 (w_3^2 + w_5^2 + w_7^2 + \\ & w_9^2) + w_3^2 (w_5^2 + w_7^2 + w_9^2) + w_5^2 (w_7^2 + w_9^2) + w_7^2 w_9^2) + \\ & w_{13}^2 (w_3^2 (w_5^2 + w_7^2 + w_9^2) + w_5^2 (w_7^2 + w_9^2) + w_7^2 w_9^2) + \\ & w_3^2 (w_5^2 (w_7^2 + w_9^2) + w_7^2 w_9^2) + w_5^2 w_7^2 w_9^2 + (w_1^2 (\\ & w_{11}^2 + w_{13}^2 + w_3^2 + w_5^2 + w_7^2 + w_9^2) + w_{11}^2 (w_{13}^2 + w_3^2 + w_5^2 \\ & + w_7^2 + w_9^2) + w_{13}^2 (w_3^2 + w_5^2 + w_7^2 + w_9^2) + w_3^2 (w_5^2 + w_7^2 + \\ & w_9^2) + w_5^2 (w_7^2 + w_9^2) + w_7^2 w_9^2) * (w_{15}^2 + w_{17}^2 + w_{19}^2) + (\\ & w_1^2 + w_{11}^2 + w_{13}^2 + w_3^2 + w_5^2 + w_7^2 + w_9^2) * (w_{15}^2 w_{17}^2 + w_{15}^2 \\ & w_{19}^2 + w_{17}^2 w_{19}^2) + w_{15}^2 w_{17}^2 w_{19}^2) \end{aligned}$$

C

$$\begin{aligned} FVEC(4) = & ((a^4 b^4 A_1 A_2 A_3 (A_9 (A_3 A_5 A_6 A_7 A_8 + A_4 (A_4 A_6 A_7 \\ & A_8 + A_5 (A_5 A_7 A_8 + A_6 (A_6 A_8 + A_7 (A_7 + 7 A_8))))) + A_4 A_5 A_6 A_7 A_8^2) L^{12} + \\ & a^4 b^4 (A_0 (A_1 + A_2) + A_1^2 + 3 A_1 A_2) A_3 (A_9 (A_3 (A_4 A_6 A_7 A_8 + \\ & A_5 (A_5 A_7 A_8 + A_6 (A_6 A_8 + A_7 (A_7 + 5 A_8)))) + A_4 (A_4 (A_5 A_7 A_8 + A_6 (A_6 A_8 + \\ & A_7 (A_7 + 5 A_8))) + A_5 (A_5 (A_6 A_8 + A_7 (A_7 + 5 A_8)) + A_6 (A_6 (A_7 + 5 A_8) + A_7 (5 \\ & A_7 + 21 A_8)))) + A_3 A_5 A_6 A_7 A_8^2 + A_4 A_8 (A_4 A_6 A_7 A_8 + A_5 (A_5 A_7 A_8 + A_6 \\ & (A_6 A_8 + A_7 (A_7 + 5 A_8))))) L^{12} + a^4 b^4 A_1 A_2 A_3 (A_9 (A_3 (A_4 (A_5 \end{aligned}$$

$$\begin{aligned}
& \$ * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 * (A6 * (A7 + 2 * A8) + 2 * A7 * (A7 + 3 * A8)) + A5 * (A5 * (A6 * \\
& \$ * (A7 + 2 * A8) + A7 * (A7 + 3 * A8)) + A6 * (A6 * (2 * A7 + 3 * A8) + A7 * (3 * A7 + 10 * A8))) + A4 \\
& \$ * (A4 * (A5 * (A7 + 3 * A8)) * (A6 + 2 * A7) + A6 * (A6 * (2 * A7 + 3 * A8) + A7 * (3 * A7 + 10 * A8))) \\
& \$ + A5 * (A5 * (3 * A6 * (A7 + 2 * A8) + A7 * (3 * A7 + 10 * A8)) + A6 * (2 * A6 * (3 * A7 + 5 * A8) + 5 * \\
& \$ A7 * (2 * A7 + 7 * A8))) + A8 * A3 * (A4 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 2 * A8))) + A5 \\
& \$ * (A5 * (A6 * A8 + A7 * (A7 + A8)) + A6 * (A6 * (A7 + A8) + A7 * (2 * A7 + 3 * A8)))) + A4 * A8 * (A4 \\
& \$ * (A5 * (A6 * A8 + A7 * (A7 + 2 * A8)) + A6 * (A6 * (A7 + A8) + A7 * (2 * A7 + 3 * A8))) + A5 * (A5 * \\
& \$ (A6 * (A7 + 2 * A8) + A7 * (2 * A7 + 3 * A8)) + A6 * (3 * A6 * (A7 + A8) + 2 * A7 * (3 * A7 + 5 * A8))) \\
& \$) * L ** 12 + a ** 4 * b ** 4 * A1 * A2 ** 2 * (A9 * (A3 * (A4 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (\\
& \$ A7 + 4 * A8)))) + A5 * (A5 * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 * (A6 * (A7 + 3 * A8) + A7 * (3 * A7 + \\
& \$ 10 * A8)))) + A4 * (A4 * (A5 * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 * (A6 * (A7 + 2 * A8) + 2 * A7 * (\\
& \$ A7 + 3 * A8))) + A5 * (A5 * (A7 + 3 * A8)) * (A6 + 2 * A7) + A6 * (3 * A6 * (A7 + 2 * A8) + 2 * A7 * (3 * \\
& \$ A7 + 10 * A8)))) + A3 * A8 * (A4 * A6 * A7 * A8 + A5 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 3 * \\
& \$ A8)))) + A4 * A8 * (A4 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 2 * A8))) + A5 * (A5 * (A6 * A8 + \\
& \$ A7 * (A7 + 2 * A8)) + A6 * (A7 + 2 * A8)) * (A6 + 3 * A7))) * L ** 12 + a ** 4 * b ** 4 * A2 * (A0 * A2 + \\
& \$ A1 ** 2 + 2 * A1 * A2) * (A9 * (A3 * (A4 * A6 * A7 * A8 + A5 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 \\
& \$ + 5 * A8)))) + A4 * (A4 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 4 * A8))) + A5 * (A5 * (A6 * A8 + \\
& \$ A7 * (A7 + 4 * A8)) + A6 * (A6 * (A7 + 4 * A8) + A7 * (4 * A7 + 15 * A8)))) + A3 * A5 * A6 * A7 * A8 \\
& \$ ** 2 + A4 * A8 * (A4 * A6 * A7 * A8 + A5 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 4 * A8)))) * L ** \\
& \$ 12 + a ** 4 * b ** 4 * (A0 + A1 + A2) * A1 * A2 * (A9 * (A3 * A5 * A6 * A7 * A8 + A4 * (A4 * A6 * A7 * A8 \\
& \$ + A5 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 6 * A8)))) + A4 * A5 * A6 * A7 * A8 ** 2) * L ** 12 + \\
& \$ a ** 4 * b ** 4 * (A0 * A2 + A1 ** 2 + 3 * A1 * A2) * A3 * (A9 * (A3 * (A4 * (A5 * A7 * A8 + A6 * (A6 * \\
& \$ A8 + A7 * (A7 + 4 * A8))) + A5 * (A5 * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 * (A6 * (A7 + 3 * A8) + A7 \\
& \$ * (3 * A7 + 10 * A8)))) + A4 * (A4 * (A5 * (A6 * A8 + A7 * (A7 + 4 * A8)) + A6 * (A6 * (A7 + 3 * A8) \\
& \$ + A7 * (3 * A7 + 10 * A8))) + A5 * (A5 * (A6 * (A7 + 4 * A8) + A7 * (3 * A7 + 10 * A8)) + A6 * (2 * A6 \\
& \$ * (2 * A7 + 5 * A8) + 5 * A7 * (2 * A7 + 7 * A8)))) + A8 * A3 * (A4 * A6 * A7 * A8 + A5 * (A5 * A7 * A8 \\
& \$ + A6 * (A6 * A8 + A7 * (A7 + 3 * A8)))) + A4 * A8 * (A4 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 \\
& \$ + 3 * A8))) + A5 * (A5 * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 * (A6 * (A7 + 3 * A8) + 2 * A7 * (2 * A7 \\
& \$ + 5 * A8)))) * L ** 12) / PAYDA) - (W1 ** 2 * (W11 ** 2 * (W13 ** 2 * \\
& \$ (W3 ** 2 + W5 ** 2 + W7 ** 2 + W9 ** 2) + W3 ** 2 * (W5 ** 2 + \\
& \$ W7 ** 2 + W9 ** 2) + W5 ** 2 * (W7 ** 2 + W9 ** 2) + W7 ** 2 * W9 ** 2) + W13 ** 2 * (W3 ** 2 * (W5 ** 2 \\
& \$ + W7 ** 2 + W9 ** 2) + W5 ** 2 * (W7 ** 2 + W9 ** 2) + W7 ** 2 * W9 ** 2) + W11 ** 2 * (W13 ** 2 * (W3 ** 2 * (\\
& \$ W5 ** 2 + W7 ** 2 + W9 ** 2) + W5 ** 2 * (W7 ** 2 + W9 ** 2) + W7 ** 2 * W9 ** 2) + W3 ** 2 * (W5 ** 2 * \\
& \$ 2 * (W7 ** 2 + W9 ** 2) + W7 ** 2 * W9 ** 2) + W5 ** 2 * W7 ** 2 * W9 ** 2) + W13 ** 2 * (W3 ** 2 * (\\
& \$ W5 ** 2 * (W7 ** 2 + W9 ** 2) + W7 ** 2 * W9 ** 2) + W5 ** 2 * W7 ** 2 * W9 ** 2) + W3 ** 2 * W5 ** 2 * \\
& \$ W7 ** 2 * W9 ** 2 + (W1 ** 2 * (W11 ** 2 * (W13 ** 2 + W3 ** 2 + W5 ** 2 + W7 ** 2 + W9 ** 2) + W13 ** 2 \\
& \$ * (W3 ** 2 + W5 ** 2 + W7 ** 2 + W9 ** 2) + W3 ** 2 * (W5 ** 2 + W7 ** 2 + W9 ** 2) + W5 ** 2 * (W7 ** 2 + \\
& \$ W9 ** 2) + W7 ** 2 * W9 ** 2) + W11 ** 2 * (W13 ** 2 * (W3 ** 2 + W5 ** 2 + W7 ** 2 + W9 ** 2) + W3 ** 2 \\
& \$ * (W5 ** 2 + W7 ** 2 + W9 ** 2) + W5 ** 2 * (W7 ** 2 + W9 ** 2) + W7 ** 2 * W9 ** 2) + W13 ** 2 * (\\
& \$ W3 ** 2 * (W5 ** 2 + W7 ** 2 + W9 ** 2) + W5 ** 2 * (W7 ** 2 + W9 ** 2) + W7 ** 2 * W9 ** 2) + W3 ** 2 * \\
& \$ (W5 ** 2 * (W7 ** 2 + W9 ** 2) + W7 ** 2 * W9 ** 2) + W5 ** 2 * W7 ** 2 * W9 ** 2) * (W15 ** 2 + W17 \\
& \$ ** 2 + W19 ** 2) + (W1 ** 2 * (W11 ** 2 + W13 ** 2 + W3 ** 2 + W5 ** 2 + W7 ** 2 + W9 ** 2) + W11 ** 2 \\
& \$ * (W13 ** 2 + W3 ** 2 + W5 ** 2 + W7 ** 2 + W9 ** 2) + W13 ** 2 * (W3 ** 2 + W5 ** 2 + W7 ** 2 + W9 ** 2))
\end{aligned}$$

$$\begin{aligned} & \$+W3^{**2} * (W5^{**2} + W7^{**2} + W9^{**2}) + W5^{**2} * (W7^{**2} + W9^{**2}) + W7^{**2} * W9^{**2} * (W15 \\ & \$^{**2} * W17^{**2} + W15^{**2} * W19^{**2} + W17^{**2} * W19^{**2}) + (W1^{**2} + W11^{**2} + W13^{**2} + \\ & \$W3^{**2} + W5^{**2} + W7^{**2} + W9^{**2}) * (W15^{**2} * W17^{**2} * W19^{**2}) \end{aligned}$$

C

$$\begin{aligned} FVEC(5) = & (a^{**5} * b^{**5} * A1 * A2 * A3 * (A9 * (A3 * (A4 * A6 * A7 * A8 + A5 * (A5 * A7 * A8 \\ & \$ + A6 * (A6 * A8 + A7 * (A7 + 5 * A8)))) + A4 * (A4 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 5 * A8 \\ & \$))) + A5 * (A5 * (A6 * A8 + A7 * (A7 + 5 * A8)) + A6 * (A6 * (A7 + 5 * A8) + A7 * (5 * A7 + 21 * A8))) \\ & \$) + A5 * A6 * A3 * A7 * A8^{**2} + A4 * A8 * (A4 * A6 * A8 * A7 + A5 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * \\ & \$ (A7 + 5 * A8))))) * L^{**10} + a^{**5} * b^{**5} * A3 * (A0 * (A1 + A2) + A1^{**2} + 3 * A1 * A2) * (A9 * (\\ & \$ A3 * (A4 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 4 * A8))) + A5 * (A5 * (A6 * A8 + A7 * (A7 + 3 * \\ & \$ A8)) + A6 * (A6 * (A7 + 3 * A8) + A7 * (3 * A7 + 10 * A8)))) + A4 * (A4 * (A5 * (A6 * A8 + A7 * (A7 + \\ & \$ 4 * A8)) + A6 * (A6 * (A7 + 3 * A8) + A7 * (3 * A7 + 10 * A8))) + A5 * (A5 * (A6 * (A7 + 4 * A8) + A7 * \\ & \$ (3 * A7 + 10 * A8)) + A6 * (2 * A6 * (2 * A7 + 5 * A8) + 5 * A7 * (2 * A7 + 7 * A8))))) + A8 * A3 * \\ & \$ (A4 * A6 * A7 * A8 + A5 * (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 3 * A8)))) + A4 * A8 * (A4 * (A5 \\ & \$ * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 3 * A8))) + A5 * (A5 * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 * (A6 \\ & \$ * (A7 + 3 * A8) + 2 * A7 * (2 * A7 + 5 * A8))))) * L^{**10} + a^{**5} * b^{**5} * A3 * (A0 * A2 + A1^{**2} + 3 * \\ & \$ A1 * A2) * (A9 * (A3 * (A4 * (A5 * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 * (A6 * (A7 + 2 * A8) + 2 * A7 \\ & \$ * (A7 + 3 * A8))) + A5 * (A5 * (A6 * (A7 + 2 * A8) + A7 * (A7 + 3 * A8)) + A6 * (A6 * (2 * A7 + 3 * A8) \\ & \$ + A7 * (3 * A7 + 10 * A8)))) + A4 * (A4 * (A5 * (A7 + 3 * A8) * (A6 + 2 * A7) + A6 * (A6 * (2 * A7 + 3 \\ & \$ * A8) + A7 * (3 * A7 + 10 * A8))) + A5 * (A5 * (3 * A6 * (A7 + 2 * A8) + A7 * (3 * A7 + 10 * A8)) + A6 \\ & \$ * (2 * A6 * (3 * A7 + 5 * A8) + 5 * A7 * (2 * A7 + 7 * A8))))) + A8 * A3 * (A4 * (A5 * A7 * A8 + A6 * (A6 \\ & \$ * A8 + A7 * (A7 + 2 * A8))) + A5 * (A5 * (A6 * A8 + A7 * (A7 + A8)) + A6 * (A6 * (A7 + A8) + A7 * (2 \\ & \$ * A7 + 3 * A8)))) + A4 * A8 * (A4 * (A5 * (A6 * A8 + A7 * (A7 + 2 * A8)) + A6 * (A6 * (A7 + A8 \\ & \$) + A7 * (2 * A7 + 3 * A8))) + A5 * (A5 * (A6 * (A7 + 2 * A8) + A7 * (2 * A7 + 3 * A8)) + A6 * (3 * A6 * \\ & \$ (A7 + A8) + 2 * A7 * (3 * A7 + 5 * A8))))) * L^{**10} + a^{**5} * b^{**5} * A1 * A2 * A3 * (A9 * (A3 * (A4 \\ & \$ * (A5 * (A6 * (A7 + 2 * A8) + A7 * (A7 + 3 * A8)) + A6 * (A6 * (A7 + A8) + A7 * (A7 + 4 * A8))) + A5 * \\ & \$ (A5 * (A6 * (A7 + A8) + A7 * A8) + A6 * (A6 * (A7 + A8) + A7 * (A7 + 5 * A8))))) + A4 * (A4 * (A5 * \\ & \$ (A6 * (2 * A7 + 3 * A8) + A7 * (A7 + 4 * A8)) + A6 * (A6 * (A7 + A8) + A7 * (A7 + 5 * A8)))) + A5 * (A5 \\ & \$ * (A6 * (3 * A7 + 4 * A8) + A7 * (A7 + 5 * A8)) + A6 * (A6 * (4 * A7 + 5 * A8) + A7 * (5 * A7 + 21 * A8)) \\ & \$)) + A8 * A3 * (A4 * (A5 * (A6 * A8 + A7 * (A7 + A8)) + A6 * A7 * (A6 + A7 + A8)) + A5 * A6 * A7 * (\\ & \$ A5 + A6 + A7 + A8)) + A4 * A8 * (A4 * (A5 * (A7 + A8) * (A6 + A7) + A6 * A7 * (A6 + A7 + A8)) + A5 * \\ & \$ (A5 * (A6 * (2 * A7 + A8) + A7 * (A7 + A8)) + A6 * (A6 * (3 * A7 + A8) + A7 * (4 * A7 + 5 * A8))))) \\ & \$ * L^{**10} + a^{**5} * b^{**5} * A1 * A2^{**2} * (A9 * (A3 * (A4 * (A5 * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 \\ & \$ * (A6 * (A7 + 2 * A8) + 2 * A7 * (A7 + 3 * A8))) + A5 * (A5 * (A6 * (A7 + 2 * A8) + A7 * (A7 + 3 * A8)) \\ & \$ + A6 * (A6 * (2 * A7 + 3 * A8) + A7 * (3 * A7 + 10 * A8)))) + A4 * (A4 * (A5 * (A6 * (A7 + 2 * A8) + \\ & \$ A7 * (A7 + 3 * A8)) + A6 * (A6 * (A7 + A8) + A7 * (A7 + 4 * A8))) + A5 * (A5 * (A6 * (2 * A7 + 3 * A8) \\ & \$ + A7 * (A7 + 4 * A8)) + A6 * (A6 * (3 * A7 + 4 * A8) + A7 * (4 * A7 + 15 * A8))))) + A3 * A8 * (A4 * \\ & \$ (A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 2 * A8))) + A5 * (A5 * (A6 * A8 + A7 * (A7 + A8)) + A6 * \\ & \$ (A6 * (A7 + A8) + A7 * (2 * A7 + 3 * A8)))) + A4 * A8 * (A4 * (A5 * (A6 * A8 + A7 * (A7 + A8)) + A6 \\ & \$ * A7 * (A6 + A7 + A8)) + A5 * (A5 * (A7 + A8) * (A6 + A7) + A6 * (A6 * (2 * A7 + A8) + A7 * (3 * A7 + \\ & \$ 4 * A8))))) * L^{**10} + a^{**5} * b^{**5} * A2 * (A0 * A2 + A1^{**2} + 2 * A1 * A2) * (A9 * (A3 * (A4 * (\\ & \$ A5 * A7 * A8 + A6 * (A6 * A8 + A7 * (A7 + 4 * A8))) + A5 * (A5 * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 * (\\ & \$ A6 * (A7 + 3 * A8) + A7 * (3 * A7 + 10 * A8)))) + A4 * (A4 * (A5 * (A6 * A8 + A7 * (A7 + 3 * A8)) + A6 \\ & \$ * (A6 * (A7 + 2 * A8) + 2 * A7 * (A7 + 3 * A8))) + A5 * (A5 * (A7 + 3 * A8) * (A6 + 2 * A7) + A6 * (3 * \end{aligned}$$

$$\begin{aligned}
& \$A6*(A7+2*AB)+2*A7*(3*A7+10*AB)))+A3*AB*(A4*A6*A7*AB+A5*(A5*A7*AB \\
& \$+A6*(A6*AB+A7*(A7+3*AB))) +A4*AB*(A4*(A5*A7*AB+A6*(A6*AB+A7*(A7+2* \\
& \$AB))) +A5*(A5*(A6*AB+A7*(A7+2*AB))+A6*(A7+2*AB)*(A6+3*A7))) *L**10+ \\
& \$a**5*b**5*(A0+A1+A2)*A1*A2*(A9*(A3*(A4*A6*A7*AB+A5*(A5*A7*AB+A6* \\
& $(A6*AB+A7*(A7+5*AB)))) +A4*(A4*(A5*A7*AB+A6*(A6*AB+A7*(A7+4*AB))) \\
& $+A5*(A5*(A6*AB+A7*(A7+4*AB))+A6*(A6*(A7+4*AB)+A7*(4*A7+15*AB)))) \\
& $+A3*A5*A6*A7*AB**2+A4*AB*(A4*A6*A7*AB+A5*(A5*A7*AB+A6*(A6*AB+A7* \\
& $(A7+4*AB)))) *L**10)/PAYDA)-(W1**2*(W11**2*(W13**2 \\
& $*(W3**2*(W5**2+W7**2+W9**2)+W5**2*(W7**2+W9**2)+W7**2*W9**2)+ \\
& $W3**2*(W5**2*(W7**2+W9**2)+W7**2*W9**2) \\
& $**2)+W5**2*W7**2*W9**2)+W13**2*(W3**2*(W5**2*(W7**2+W9**2)+W7**2* \\
& $W9**2)+W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2*W9**2)+W11**2*(W13**2* \\
& $(W3**2*(W5**2*(W7**2+W9**2)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W3**2* \\
& $W5**2*W7**2*W9**2)+W13**2*W3**2*W5**2*W7**2*W9**2+(W1**2*(W11**2* \\
& $(W13**2*(W3**2+W5**2+W7**2+W9**2)+W3**2*(W5**2+W7**2+W9**2)+W5**2 \\
& $*(W7**2+W9**2)+W7**2*W9**2)+W13**2*(W3**2*(W5**2+W7**2+W9**2)+W5** \\
& $2*(W7**2+W9**2)+W7**2*W9**2)+W3**2*(W5**2*(W7**2+W9**2)+W7**2*W9** \\
& $2)+W5**2*W7**2*W9**2)+W11**2*(W13**2*(W3**2*(W5**2+W7**2+W9**2)+W5 \\
& $**2*(W7**2+W9**2)+W7**2*W9**2)+W3**2*(W5**2*(W7**2+W9**2)+W7**2*W9 \\
& $**2)+W5**2*W7**2*W9**2)+W13**2*(W3**2*(W5**2*(W7**2+W9**2)+W7** \\
& $2*W9**2)+W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2*W9**2)*(W15**2+W17** \\
& $2+W19**2)+(W1**2*(W11**2*(W13**2+W3**2+W5**2+W7**2+W9**2)+W13**2* \\
& $(W3**2+W5**2+W7**2+W9**2)+W3**2*(W5**2+W7**2+W9**2)+W5**2*(W7**2+ \\
& $W9**2)+W7**2*W9**2)+W11**2*(W13**2*(W3**2+W5**2+W7**2+W9**2)+W3**2 \\
& $*(W5**2+W7**2+W9**2)+W5**2*(W7**2+W9**2)+W7**2*W9**2)+W13**2*(W3** \\
& $2*(W5**2+W7**2+W9**2)+W5**2*(W7**2+W9**2)+W7**2*W9**2)+W3**2*(W5** \\
& $2*(W7**2+W9**2)+W7**2*W9**2)+W5**2*W7**2*W9**2)*(W15**2*W17**2+W15 \\
& $**2*W19**2+W17**2*W19**2)+(W1**2*(W11**2+W13**2+W3**2+W5**2+W7**2 \\
& $+W9**2)+W11**2*(W13**2+W3**2+W5**2+W7**2+W9**2)+W13**2*(W3**2+W5 \\
& $**2+W7**2+W9**2)+W3**2*(W5**2+W7**2+W9**2)+W5**2*(W7**2+W9**2) \\
& $+W7**2*W9**2)*(W15**2*W17**2*W19**2))
\end{aligned}$$

C

$$\begin{aligned}
FVED(6) = & ((a**6*b**6*A1*A2*A3*(A9*(A3*(A4*(A5*A7*AB+A6*(A6*AB+ \\
& $A7*(A7+4*AB)))+A5*(A5*(A6*AB+A7*(A7+3*AB))+A6*(A6*(A7+3*AB)+A7*(3* \\
& $A7+10*AB)))) +A4*(A4*(A5*(A6*AB+A7*(A7+4*AB))+A6*(A6*(A7+3*AB)+A7*(\\
& $3*A7+10*AB)))+A5*(A5*(A6*(A7+4*AB)+A7*(3*A7+10*AB))+A6*(2*A6*(2*A7 \\
& $+5*AB)+5*A7*(2*A7+7*AB)))) +AB*A3*(A4*A6*A7*AB+A5*(A5*A7*AB+A6*(A6 \\
& $*AB+A7*(A7+3*AB)))+A4*AB*(A4*(A5*A7*AB+A6*(A6*AB+A7*(A7+3*AB)))+ \\
& $A5*(A5*(A6*AB+A7*(A7+3*AB))+A6*(A6*(A7+3*AB)+2*A7*(2*A7+5*AB)))) \\
& $*L**8+a**6*b**6*A3*(A0*(A1+A2)+A1**2+3*A1*A2)*(A9*(A3*(A4*(A5*(A6 \\
& $*AB+A7*(A7+3*AB))+A6*(A6*(A7+2*AB)+2*A7*(A7+3*AB)))+A5*(A5*(A6*(A7 \\
& $+2*AB)+A7*(A7+3*AB))+A6*(A6*(2*A7+3*AB)+A7*(3*A7+10*AB)))) +A4*(A4* \\
& $(A5*(A7+3*AB)*(A6+2*A7)+A6*(A6*(2*A7+3*AB)+A7*(3*A7+10*AB)))+A5*(\\
& $A5*(3*A6*(A7+2*AB)+A7*(3*A7+10*AB))+A6*(2*A6*(3*A7+5*AB)+5*A7*(2*
\end{aligned}$$

$$\begin{aligned}
& \$A7+7*AB)))) +A8*A3*(A4*(A5*A7*AB+A6*(A6*AB+A7*(A7+2*AB))) +A5*(A5*(\\
& \$A6*AB+A7*(A7+AB)) +A6*(A6*(A7+AB)+A7*(2*A7+3*AB)))) +A4*AB*(A4*(A5*(\\
& \$A6*AB+A7*(A7+2*AB)) +A6*(A6*(A7+AB)+A7*(2*A7+3*AB))) +A5*(A5*(A6*(A7 \\
& \$+2*AB)+A7*(2*A7+3*AB)) +A6*(3*A6*(A7+AB)+2*A7*(3*A7+5*AB))))) *L**8+ \\
& \$a**6*b**6*A3*(A0*A2+A1**2+3*A1*A2)*(A9*(A3*(A4*(A5*(A6*(A7+2*AB)+ \\
& \$A7*(A7+3*AB)) +A6*(A6*(A7+AB)+A7*(A7+4*AB))) +A5*(A5*(A6*(A7+AB)+A7* \\
& \$AB)+A6*(A6*(A7+AB)+A7*(A7+5*AB)))) +A4*(A4*(A5*(A6*(2*A7+3*AB)+A7* \\
& \$ (A7+4*AB)) +A6*(A6*(A7+AB)+A7*(A7+5*AB))) +A5*(A5*(A6*(3*A7+4*AB) \\
& \$+A7*(A7+5*AB)) +A6*(A6*(4*A7+5*AB)+A7*(5*A7+21*AB))))) +A8*A3*(A4* \\
& \$ (A5*(A6*AB+A7*(A7+AB)) +A6*A7*(A6+A7+AB)) +A5*A6*A7*(A5+A6+A7+AB \\
& \$)) +A4*AB*(A4*(A5*(A7+AB)*(A6+A7)+A6*A7*(A6+A7+AB)) +A5*(A5*(A6*(2* \\
& \$A7+AB)+A7*(A7+AB)) +A6*(A6*(3*A7+AB)+A7*(4*A7+5*AB))))) *L**8+a**6* \\
& \$b**6*A1*A2*A3*(A9*(A3*(A4*(A5*(A6*(A7+AB)+A7*AB)+A6*A7*AB)+A5*A6* \\
& \$A7*AB)+A4*(A4*(A5*(A6*(A7+AB)+A7*AB)+A6*A7*AB)+A5*(A5*(A6*(A7+AB)+ \\
& \$A7*AB)+A6*(A6*(A7+AB)+A7*(A7+7*AB))))) +A3*A4*A5*A6*A7*AB+A4*A5*A6* \\
& \$A7*AB*(A4+A5+A6+A7+AB)) *L**8+a**6*b**6*A1*A2**2*(A9*(A3*(A4*(A5*(\\
& \$A6*(A7+2*AB)+A7*(A7+3*AB)) +A6*(A6*(A7+AB)+A7*(A7+4*AB))) +A5*(A5* \\
& \$ (A6*(A7+AB)+A7*AB)+A6*(A6*(A7+AB)+A7*(A7+5*AB))))) +A4*(A4*(A5*(A6* \\
& \$ (A7+AB)+A7*AB)+A6*A7*AB)+A5*(A5*(A6*(A7+AB)+A7*AB)+A6*(A6*(A7+AB) \\
& \$+A7*(A7+6*AB))))) +A3*AB*(A4*(A5*(A6*AB+A7*(A7+AB)) +A6*A7*(A6+A7+ \\
& \$AB)) +A5*A6*A7*(A5+A6+A7+AB)) +A4*A5*A6*A7*AB*(A4+A5+A6+A7+AB)) *L**8 \\
& \$+a**6*b**6*A2*(A0*A2+A1**2+2*A1*A2)*(A9*(A3*(A4*(A5*(A6*AB+A7*(A7 \\
& \$+3*AB)) +A6*(A6*(A7+2*AB)+2*A7*(A7+3*AB))) +A5*(A5*(A6*(A7+2*AB)+A7* \\
& \$ (A7+3*AB)) +A6*(A6*(2*A7+3*AB)+A7*(3*A7+10*AB)))) +A4*(A4*(A5*(A6*(\\
& \$A7+2*AB)+A7*(A7+3*AB)) +A6*(A6*(A7+AB)+A7*(A7+4*AB))) +A5*(A5*(A6*(2 \\
& \$*A7+3*AB)+A7*(A7+4*AB)) +A6*(A6*(3*A7+4*AB)+A7*(4*A7+15*AB))))) +A3* \\
& \$AB*(A4*(A5*A7*AB+A6*(A8*AB+A7*(A7+2*AB))) +A5*(A5*(A6*AB+A7*(A7+AB) \\
& \$) +A6*(A6*(A7+AB)+A7*(2*A7+3*AB)))) +A4*AB*(A4*(A5*(A6*AB+A7*(A7+AB) \\
& \$) +A6*A7*(A6+A7+AB)) +A5*(A5*(A7+AB)*(A6+A7)+A6*(A6*(2*A7+AB)+A7*(3* \\
& \$A7+4*AB))))) *L**8+a**6*b**6*(A0+A1+A2)*A1*A2*(A9*(A3*(A4*(A5*A7* \\
& \$AB+A6*(A6*AB+A7*(A7+4*AB))) +A5*(A5*(A6*AB+A7*(A7+3*AB)) +A6*(A6*(A7 \\
& \$+3*AB)+A7*(3*A7+10*AB)))) +A4*(A4*(A5*(A6*AB+A7*(A7+3*AB)) +A6*(A6*(\\
& \$A7+2*AB)+2*A7*(A7+3*AB))) +A5*(A5*(A7+3*AB)*(A6+2*A7)+A6*(3*A6*(A7+ \\
& \$2*AB)+2*A7*(3*A7+10*AB))))) +A3*AB*(A4*A6*A7*AB+A5*(A5*A7*AB+A6*(A6 \\
& \$*AB+A7*(A7+3*AB)))) +A4*AB*(A4*(A5*A7*AB+A6*(A6*AB+A7*(A7+2*AB))) + \\
& \$A5*(A5*(A6*AB+A7*(A7+2*AB)) \\
& \$+A6*(A7+2*AB)*(A6+3*A7)))) *L**8) /PAYDA) \\
& \$-(W1**2*(W11**2*(W13**2*(W3**2*(W5**2*(W7**2+W9**2)+W7**2*W9**2)+ \\
& \$W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2*W9**2)+W13**2*W3**2*W5**2*W7 \\
& \$**2*W9**2)+W11**2*W13**2*W3**2*W5**2*W7**2*W9**2+(W1**2*(W11**2*(\\
& \$W13**2*(W3**2*(W5**2+W7**2+W9**2)+W5**2*(W7**2+W9**2)+W7**2*W9**2) \\
& \$+W3**2*(W5**2*(W7**2+W9**2)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W13**2 \\
& \$*(W3**2*(W5**2*(W7**2+W9**2)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W3**2 \\
& \$*W5**2*W7**2*W9**2)+W11**2*(W13**2*(W3**2*(W5**2*(W7**2+W9**2)+W7
\end{aligned}$$

$$\begin{aligned}
 & \$**2*W9**2) + W5**2*W7**2*W9**2) + W3**2*W5**2*W7**2*W9**2) + W13**2*W3** \\
 & \$2*W5**2*W7**2*W9**2) * (W15**2+W17**2+W19**2) + (W1**2*(W11**2*(W13**2 \\
 & \$*(W3**2+W5**2+W7**2+W9**2) + W3**2*(W5**2+W7**2+W9**2) + W5**2*(W7**2+ \\
 & \$W9**2) + W7**2*W9**2) + W13**2*(W3**2*(W5**2+W7**2+W9**2) + W5**2*(W7**2 \\
 & \$+W9**2) + W7**2*W9**2) + W3**2*(W5**2*(W7**2+W9**2) + W7**2*W9**2) + W5**2 \\
 & \$*W7**2*W9**2) + W11**2*(W13**2*(W3**2*(W5**2+W7**2+W9**2) + W5**2*(W7 \\
 & \$**2+W9**2) + W7**2*W9**2) + W3**2*(W5**2*(W7**2+W9**2) + W7**2*W9**2) + \\
 & \$W5**2*W7**2*W9**2) + W13**2*(W3**2*(W5**2*(W7**2+W9**2) + W7**2*W9**2) \\
 & \$+W5**2*W7**2*W9**2) + W3**2*W5**2*W7**2*W9**2) * (W15**2*W17**2+W15**2 \\
 & \$*W19**2+W17**2*W19**2) + (W1**2*(W11**2*(W13**2+W3**2+W5**2+W7**2+W9 \\
 & \$**2) + W13**2*(W3**2+W5**2+W7**2+W9**2) + W3**2*(W5**2+W7**2+W9**2) + \\
 & \$W5**2*(W7**2+W9**2) + W7**2*W9**2) + W11**2*(W13**2*(W3**2+W5**2 \\
 & \$+W7**2+W9**2) + W3**2*(W5**2+W7**2+W9**2) + W5**2*(W7**2+W9**2) + W7**2 \\
 & \$*W9**2) + W13**2*(W3**2*(W5**2+W7**2+W9**2) + W5**2*(W7**2+W9**2) + W7 \\
 & \$**2*W9**2) + W3**2*(W5**2*(W7**2+W9**2) + W7**2*W9**2) + W5**2*W7**2*W9 \\
 & \$**2) * (W15**2*W17**2*W19**2))
 \end{aligned}$$

C

$$\begin{aligned}
 & FVEC(7) = ((a**7*b**7*A1*A2*A3*(A9*(A3*(A4*(A5*(A6*AB+A7* \\
 & \$ (A7+3*AB)) + A6*(A6*(A7+2*AB) + 2*A7*(A7+3*AB))) + \\
 & \$A5*(A5*(A6*(A7+2*AB) + A7*(A7+3 \\
 & \$*AB)) + A6*(A6*(2*A7+3*AB) + A7*(3*A7+10*AB)))) + A4*(A4*(A5*(A7+3*AB) * (\\
 & \$A6+2*A7) + A6*(A6*(2*A7+3*AB) + A7*(3*A7+10*AB))) + A5*(A5*(3*A6*(A7+2* \\
 & \$AB) + A7*(3*A7+10*AB)) + A6*(2*A6*(3*A7+5*AB) + 5*A7*(2*A7+7*AB))))) + AB* \\
 & \$A3*(A4*(A5*A7*AB+A6*(A6*AB+A7*(A7+2*AB))) + A5*(A5*(A6*AB+A7*(A7+AB \\
 & \$)) + A6*(A6*(A7+AB) + A7*(2*A7+3*AB)))) + A4*AB*(A4*(A5*(A6*AB+A7*(A7+2* \\
 & \$AB)) + A6*(A6*(A7+AB) + A7*(2*A7+3*AB))) + A5*(A5*(A6*(A7+2*AB) + A7*(2* \\
 & \$A7+3*AB)) + A6*(3*A6*(A7+AB) + 2*A7*(3*A7+5*AB))))) * L**6+a**7*b**7*(A0 \\
 & \$*(A1+A2) + A1**2+3*A1*A2) * A3*(A9*(A3*(A4*(A5*(A6*(A7+2*AB) + A7*(A7 \\
 & \$+3*AB)) + A6*(A6*(A7+AB) + A7*(A7+4*AB))) + A5*(A5*(A6*(A7+AB) + A7*AB) + \\
 & \$A6*(A6*(A7+AB) + A7*(A7+5*AB)))) + A4*(A4*(A5*(A6*(2*A7+3*AB) + A7*(A7+ \\
 & \$4*AB)) + A6*(A6*(A7+AB) + A7*(A7+5*AB))) + A5*(A5*(A6*(3*A7+4*AB) + A7*(A7 \\
 & \$+5*AB)) + A6*(A6*(4*A7+5*AB) + A7*(5*A7+21*AB))))) + AB*A3*(A4*(A5*(A6* \\
 & \$AB+A7*(A7+AB)) + A6*A7*(A6+A7+AB)) + A5*A6*A7*(A5+A6+A7+AB)) + A4*AB*(A4 \\
 & \$*(A5*(A7+AB) * (A6+A7) + A6*A7*(A6+A7+AB)) + A5*(A5*(A6*(2*A7+AB) + A7*(A7 \\
 & \$+AB)) + A6*(A6*(3*A7+AB) + A7*(4*A7+5*AB))))) * L**6+a**7*b**7*A3*(A0*A2 \\
 & \$+A1**2+3*A1*A2) * (A9*(A3*(A4*(A5*(A6*(A7+AB) + A7*AB) + A6*A7*AB) + \\
 & \$A5*A6*A7*AB) + A4*(A4*(A5*(A6*(A7+AB) + A7*AB) + A6*A7*AB) + A5*(A5*(A6* \\
 & \$ (A7+AB) + A7*AB) + A6*(A6*(A7+AB) + A7*(A7+7*AB))))) + A3*A4*A5*A6*A7*AB+ \\
 & \$A4*A5*A6*A7*AB*(A4+A5+A6+A7+AB)) * L**6+a**7*b**7*A1*A2*A3*A4*A5*A6* \\
 & \$A7*AB*A9*L**6+a**7*b**7*A1*A2**2*(A9*(A3*(A4*(A5*(A6*(A7+AB) + A7* \\
 & \$AB) + A6*A7*AB) + A5*A6*A7*AB) + A4*A5*A6*A7*AB) + A3*A4*A5*A6*A7*AB) * \\
 & \$L**6+a**7*b**7 \\
 & \$*A2*(A0*A2+A1**2+2*A1*A2) * (A9*(A3*(A4*(A5*(A6*(A7+2*AB) + A7*(A7+3* \\
 & \$AB)) + A6*(A6*(A7+AB) + A7*(A7+4*AB))) + A5*(A5*(A6*(A7+AB) + A7*AB) + A6*(
 \end{aligned}$$

$$\begin{aligned}
& \$A6*(A7+A8)+A7*(A7+5*AB)))+A4*(A4*(A5*(A6*(A7+A8)+A7*AB)+A6*A7*AB) \\
& \$+A5*(A5*(A6*(A7+A8)+A7*AB)+A6*(A6*(A7+A8)+A7*(A7+6*AB)))))+A3*AB* \\
& \$ (A4*(A5*(A6*AB+A7*(A7+A8))+A6*A7*(A6+A7+A8))+A5*A6*A7*(A5+A6+A7+ \\
& \$AB))+A4*A5*A6*A7*AB*(A4+A5+A6+A7+A8))*L**6+a**7*b**7*(A0+A1+A2)* \\
& \$A1*A2*(A9*(A3*(A4*(A5*(A6*AB+A7*(A7+3*AB))+A6*(A6*(A7+2*AB)+2*A7 \\
& \$*(A7+3*AB)))+A5*(A5*(A6*(A7+2*AB)+A7*(A7+3*AB))+A6*(A6*(2*A7+3* \\
& \$AB)+A7*(3*A7+10*AB)))))+A4*(A4*(A5*(A6*(A7+2*AB)+A7*(A7+3*AB))+A6* \\
& \$ (A6*(A7+A8)+A7*(A7+4*AB)))+A5*(A5*(A6*(2*A7+3*AB)+A7*(A7+4*AB))+ \\
& \$A6*(A6*(3*A7+4*AB)+A7*(4*A7+15*AB)))))+A3*AB*(A4*(A5*A7*AB+A6*(A6 \\
& \$*AB+A7*(A7+2*AB)))+A5*(A5*(A6*AB+A7*(A7+A8))+A6*(A6*(A7+A8)+A7*(2 \\
& \$*A7+3*AB)))))+A4*AB*(A4*(A5*(A6*AB+A7*(A7+A8))+A6*A7*(A6+A7+A8))+ \\
& \$A5*(A5*(A7+A8)*(A6+A7)+A6*(A6*(2*A7+A8)+A7*(3*A7+4*AB))))))* \\
& \$L**6)/PAYDA)-(W1**2*W11**2*W13 \\
& **2*W3**2*W5**2*W7**2*W9**2+(W1**2*(W11**2*(W13**2*(W3**2*(W5**2* \\
& \$ (W7**2+W9**2)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2*W9 \\
& **2)+W13**2*W3**2*W5**2*W7**2*W9**2)+W11**2*W13**2*W3**2*W5**2*W7 \\
& **2*W9**2)*(W15**2+W17**2+W19**2)+(W1**2*(W11**2*(W13**2*(W3**2* \\
& \$ (W5**2+W7**2+W9**2)+W5**2*(W7**2+W9**2)+W7**2*W9**2)+W3**2*(W5**2 \\
& **2*(W7**2+W9**2)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W13**2*(W3**2*(W5** \\
& \$2*(W7**2+W9**2)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2* \\
& \$W9**2)+W11**2*(W13**2*(W3**2*(W5**2*(W7**2+W9**2)+W7**2*W9**2 \\
& \$)+W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2*W9**2)+W13**2*W3**2*W5**2* \\
& \$W7**2*W9**2)*(W15**2*W17**2+W15**2*W19**2+W17**2*W19**2)+(W1**2* \\
& \$ (W11**2*(W13**2*(W3**2+W5**2+W7**2+W9**2)+W3**2*(W5**2+W7**2+W9**2 \\
& \$)+W5**2*(W7**2+W9**2)+W7**2*W9**2)+W13**2*(W3**2*(W5**2+W7**2+W9 \\
& **2)+W5**2*(W7**2+W9**2)+W7**2*W9**2)+W3**2*(W5**2*(W7**2+W9**2)+ \\
& \$W7**2*W9**2)+W5**2*W7**2*W9**2)+W11**2*(W13**2*(W3**2*(W5**2+W7**2 \\
& \$+W9**2)+W5**2*(W7**2+W9**2)+W7**2*W9**2)+W3**2*(W5**2*(W7**2+W9**2) \\
& \$)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W13**2*(W3**2*(W5**2*(W7**2+W9 \\
& **2)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2*W9**2)*(\\
& \$W15**2*W17**2*W19**2))
\end{aligned}$$

C

$$\begin{aligned}
& FVEC(B)=(a**8*b**8*A2*A1*A3*(A9*(A3*(A4*(A5*(A6* \\
& \$ (A7+2*AB)+A7*(\\
& \$A7+3*AB))+A6*(A6*(A7+A8)+A7*(A7+4*AB)))+A5*(A5*(A6*(A7+A8)+A7*AB)+ \\
& \$A6*(A6*(A7+A8)+A7*(A7+5*AB)))))+A4*(A4*(A5*(A6*(2*A7+3*AB)+A7*(A7+4 \\
& \$*AB))+A6*(A6*(A7+A8)+A7*(A7+5*AB)))+A5*(A5*(A6*(3*A7+4*AB)+A7*(A7 \\
& \$+5*AB))+A6*(A6*(4*A7+5*AB)+A7*(5*A7+21*AB)))))+A8*A3*(A4*(A5*(A6* \\
& \$AB+A7*(A7+A8))+A6*A7*(A6+A7+A8))+A5*A6*A7*(A5+A6+A7+A8))+A4*AB*(A4 \\
& **2*(A5*(A7+A8)*(A6+A7)+A6*A7*(A6+A7+A8))+A5*(A5*(A6*(2*A7+A8)+A7*(A7 \\
& \$+A8))+A6*(A6*(3*A7+A8)+A7*(4*A7+5*AB))))))*L**4+a**8*b**8*(A0*(A1+ \\
& \$A2)+A1**2+3*A1*A2)*A3*(A9*(A3*(A4*(A5*(A6*(A7+A8)+A7*AB)+A6*A7*AB \\
& \$)+A5*A6*A7*AB)+A4*(A4*(A5*(A6*(A7+A8)+A7*AB)+A6*A7*AB)+A5*(A5*(A6* \\
& \$ (A7+A8)+A7*AB)+A6*(A6*(A7+A8)+A7*(A7+7*AB)))))+A3*A4*A5*A6*A7*AB+
\end{aligned}$$

```

$A4*A5*A6*A7*AB*(A4+A5+A6+A7+AB))*L**4+a**8*b**8*A4*A5*A6*A7*AB*A3*
$A9*(A0*A2+A1**2+3*A1*A2))*L**4+a**8*b**8*A2*(A0*A2+A1**2+2*A1*A2)*
$(A9*(A3*(A4*(A5*(A6*(A7+AB)+A7*AB)+A6*A7*AB)+A5*A6*A7*AB)+A4*A5*
$A6*A7*AB)+A3*A4*A5*A6*A7*AB))*L**4+a**8*b**8*A2*(A0+A1+A2)*A1*(A9*
$(A3*(A4*(A5*(A6*(A7+2*AB)+A7*(A7+3*AB))+A6*(A6*(A7+AB)+A7*(A7+4*AB
$)))+A5*(A5*(A6*(A7+AB)+A7*AB)+A6*(A6*(A7+AB)+A7*(A7+5*AB)))))+A4*(
$A4*(A5*(A6*(A7+AB)+A7*AB)+A6*A7*AB)+A5*(A5*(A6*(A7+AB)+A7*AB)+A6*(
$A6*(A7+AB)+A7*(A7+6*AB)))))+A3*AB*(A4*(A5*(A6*AB+A7*(A7+AB))+A6*A7
$*(A6+A7+AB))+A5*A6*A7*(A5+A6+A7+AB))+A4*A5*A6*A7*AB*(A4+A5+A6+A7+
$AB))*L**4)/PAYDA)-(W1**2*W11**2*W13**2*
$W3**2*W5**2*W7**2*W9**2*(W15**2+W17**2
$+W19**2)+(W1**2*(W11**2*(W13**2*(W3**2*(W5**2*(W7**2+W9**2)+W7**2
$*W9**2)+W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2*W9**2)+W13**2*W3**2*
$W5**2*W7**2*W9**2)+W11**2*W13**2*W3**2*W5**2*W7**2*W9**2)*(W15**2
$*W17**2+W15**2*W19**2+W17**2*W19**2)+(W1**2*(W11**2*(W13**2*(W3**2
$*(W5**2+W7**2+W9**2)+W5**2*(W7**2+W9**2)+W7**2*W9**2)+W3**2*(W5**2
$*(W7**2+W9**2)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W13**2*(W3**2*(W5**
$2*(W7**2+W9**2)+W7**2*W9**2)+W5**2*W7**2*W9**2)+W3**2*W5**2*
$W7**2*W9**2)+W11**2*(W13**2*(W3**2*(W5**2*(W7**2+W9**2)+W7**2*W9
$**2)+W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2*W9**2)+W13**2*W3**2*W5**
$2*W7**2*W9**2)*(W15**2*W17**2*W19**2))

```

C

```

FVEC(9)=( (a**9*b**9*A1*A2*A3*(A9*(A3*(A4*(A5*(A6*(A7+AB)+A7*AB
$)+A6*A7*AB)+A5*A6*A7*AB)+A4*(A4*(A5*(A6*(A7+AB)+A7*AB)+
$A6*A7*AB)+A5
$*(A5*(A6*(A7+AB)+A7*AB)+A6*(A6*(A7+AB)+A7*(A7+7*AB)))))+A3*A4*A5*
$A6*A7*AB+A4*A5*A6*A7*AB*(A4+A5+A6+A7+AB))*L**2+a**9*b**9*A3*A4*A5*
$A6*A7*AB*A9*(A0*(A1+A2)+A1**2+3*A1*A2))*L**2+a**9*b**9*(A0+A1+A2)*
$A1*A2*(A9*(A3*(A4*(A5*(A6*(A7+AB)+A7*AB)+A6*A7*AB)+A5*A6*A7*AB)+
$A4*A5*A6*A7*AB)+A3*A4*A5*A6*A7*AB))*L**2)/PAYDA)-(W1**2*
$W11**2*W13
$**2*W3**2*W5**2*W7**2*W9**2*(W15**2*W17**2+W15**2*W19**2+W17**2*
$W19**2)+(W1**2*(W11**2*(W13**2*(W3**2*(W5**2*(W7**2+W9**2)+W7**2*
$W9**2)+W5**2*W7**2*W9**2)+W3**2*W5**2*W7**2*W9**2)+W13**2*W3**2*
$W5**2*W7**2*W9**2)+W11**2*W13**2*W3**2*W5**2*W7**2*W9**2)*(W15**2
$*W17**2*W19**2))

```

C

C

C

C

C

```

FVEC(10)=( (a**10*b**10*A2*A3*A4*A5*A6*A7*AB*A9*A10)
$/PAYDA)-W1**2*W3**2*W5**2*W7**2*W9**2*W11**2*W13**2*
$W15**2*W17**2*W19**2

```

RETURN

END

EK 4

DURUM UZAY MODELİNİN KAYIPLI DURUMU İÇİN PROGRAM

(4 tüp için)

```

IMPLICIT REAL*8(A-L,O-Z)
EXTERNAL FCN
REAL*8 M
DIMENSION X(8),FVEC(8),WA(1500)
COMMON/PASIT/A,B,C,Q,M,W(7)
C
PI=4.DO*ATAN(1.DO)
C
DO 111 i=1,8
PRINT*, 'ENTER X(',i,') VALUE '
111 READ*, X(i)
M=17.DO
DO 112 i=1,7,2
PRINT*, 'ENTER w(',i,') VALUE '
READ*, w(i)
112 w(i)=2.DO*PI*w(i)
C
TOL      = 1.E-8
N        = 8
LWA      = 1500
c        = 35000
q        = 0.00114
C
C ----- SOLVE NON-LINEAR SYSTEM -----
C
CALL HYBRD1(FCN,N,X,FVEC,TOL,INFO,WA,LWA)
C
PRINT*, 'INFO = ',INFO
PRINT*,
PRINT*,
PRINT*, ' COZUM VE HATA VEKTORU '
DO 4 I=1,N
PRINT*, X(I),FVEC(I)
4 CONTINUE
PRINT*,
STOP
END
C
SUBROUTINE FCN(N,X,FVEC)
IMPLICIT REAL*8(A-L,O-Z)
REAL*8 M
COMMON/PASIT/A,Q,C,B,M,W(7)

```

DIMENSION X(N),FVEC(N)

C

A1 = X(1)

A2 = X(2)

A3 = X(3)

A4 = X(4)

C

I1 = (q*M)/(10*A1)

I2 = (q*M)/(10*A2)

I3 = (q*M)/(10*A3)

I4 = (q*M)/(10*A4)

C

c1 = (A1*M)/(10*q*c**2)

c2 = (A2*M)/(10*q*c**2)

c3 = (A3*M)/(10*q*c**2)

c4 = (A4*M)/(10*q*c**2)

C

r1 = I1*((w(1)/A1)**(1/2))

r2 = I2*((w(3)/A2)**(1/2))

r3 = I3*((w(5)/A3)**(1/2))

r4 = I4*((w(7)/A4)**(1/2))

C

w1 = w(1)

w3 = w(3)

w5 = w(5)

w7 = w(7)

C

s1 = X(5)

s2 = X(6)

s3 = X(7)

s4 = X(8)

C

FVEC(1)=(3/2)*(I1*(I2*(I3*r4+I4*r3)+I3*I4*r2)+I2*I3*I4
 *r1)/(I1*I2*I3*I4)-(2*s1+2*s2+2*s3+2*s4)

C

FVEC(2)=(1/(c1*I1))+((1/4)*(c2*(c3*(c4*(I1**2*(I2**2*(2*
 \$I3**2*r4**2+9*I3*I4*r3*r4+2*I4**2*r3**2)+9*I2*I3*I4*r2*(I3*r4+I4
 \$*r3)+2*I3**2*I4**2*r2**2)+9*I1*I2*I3*I4*r1*(I2*(I3*r4+I4*r3)+I3*I4
 \$*r2)+2*I2**2*I3**2*I4**2*r1**2)+4*I1**2*I2**2*I3*I4*(I3+I4))+4*c4
 \$*I1**2*I2*I3*I4**2*(I2+I3))+4*c3*c4*I1*I2*I3**2*I4**2*(I1+I2))/
 \$(c2*c3*c4*I1**2*I2**2*I3**2*I4**2))-(s1**2+s1*(4*s2+4*s3+4*s4)+
 \$s2**2+s2*(4*s3+4*s4)+s3**2+4*s3*s4+s4**2+w1**2+w3**2+w5**2+w7**2)

C

FVEC(3)=(3/2)*(I2*(I3*r4+I4*r3)+I3*I4*r2)/(c1*I1*I2*I3*I4))+

$$\begin{aligned}
& \$((1/8) * (c2 * (c3 * (3 * c4 * (11 ** 2 * (2 * 12 ** 2 * r3 * r4 * (13 * r4 + 14 * r3) + 12 * r2 * \\
& \$ (2 * 13 ** 2 * r4 ** 2 + 9 * 13 * 14 * r3 * r4 + 2 * 14 ** 2 * r3 ** 2) + 2 * 13 * 14 * r2 ** 2 * (13 * r4 + \\
& \$ 14 * r3)) + 11 * r1 * (12 ** 2 * (2 * 13 ** 2 * r4 ** 2 + 9 * 13 * 14 * r3 * r4 + 2 * 14 ** 2 * r3 ** 2) + \\
& \$ 9 * 12 * 13 * 14 * r2 * (13 * r4 + 14 * r3) + 2 * 13 ** 2 * 14 ** 2 * r2 ** 2) + 2 * 12 * 13 * 14 * r1 ** 2 \\
& \$ * (12 * (13 * r4 + 14 * r3) + 13 * 14 * r2)) + 4 * 11 * 12 * 14 * (11 * (12 * (13 * (3 * r3 + 2 \\
& \$ * r4) + 14 * r3) + 3 * 13 * r2 * (13 + 14)) + 3 * 12 * 13 * r1 * (13 + 14))) + 4 * c4 * 11 * 13 \\
& \$ * 14 * (11 * (3 * 12 ** 2 * r4 + 12 * (3 * 13 * r4 + 14 * (3 * r2 + 2 * r3)) + 13 * 14 * r2) + 3 * \\
& \$ 12 * 14 * r1 * (12 + 13))) + 4 * c3 * c4 * 12 * 13 * 14 * (3 * 11 ** 2 * (13 * r4 + 14 * r3) + 11 \\
& \$ * (3 * 12 * (13 * r4 + 14 * r3) + 13 * 14 * (3 * r1 + 2 * r2)) + 12 * 13 * 14 * r1)) / \\
& \$ (c2 * c3 * c4 * 11 ** 2 * 12 ** 2 * 13 ** 2 * 14 ** 2) - 2 * (s1 ** 2 * (s2 + s3 + s4) + s1 * (s2 ** 2 \\
& \$ + 4 * s2 * (s3 + s4) + s3 ** 2 + 4 * s3 * s4 + s4 ** 2 + w3 ** 2 + w5 ** 2 + w7 ** 2) + s2 ** 2 * (s3 + \\
& \$ s4) + s2 * (s3 ** 2 + 4 * s3 * s4 + s4 ** 2 + w1 ** 2 + w5 ** 2 + w7 ** 2) + s3 ** 2 * s4 + s3 * (s4 ** 2 + \\
& \$ w1 ** 2 + w3 ** 2 + w7 ** 2) + s4 * (w1 ** 2 + w3 ** 2 + w5 ** 2))
\end{aligned}$$

C

$$\begin{aligned}
& FVEC(4) = ((1/4) * (c2 * (c3 * (c4 * (12 ** 2 * (2 * 13 ** 2 * r4 ** 2 + 9 * 13 * 14 * r3 * \\
& \$ r4 + 2 * 14 ** 2 * r3 ** 2) + 9 * 12 * 13 * 14 * r2 * (13 * r4 + 14 * r3) + 2 * 13 ** 2 * 14 ** 2 * r2 ** 2) \\
& \$ + 4 * 12 ** 2 * 13 * 14 * (13 + 14)) + 4 * c4 * 12 * 13 * 14 ** 2 * (12 + 13)) + 4 * c3 * c4 * 12 * \\
& \$ 13 ** 2 * 14 ** 2) / (c1 * c2 * c3 * c4 * 11 * 12 ** 2 * 13 ** 2 * 14 ** 2)) + ((1/16) * (c2 * (c3 * \\
& \$ (c4 * (2 * 11 ** 2 * (2 * 12 ** 2 * r3 ** 2 * r4 ** 2 + 9 * 12 * r2 * r3 * r4 * (13 * r4 + 1 \\
& \$ 4 * r3) + r2 ** 2 * (2 * 13 ** 2 * r4 ** 2 + 9 * 13 * 14 * r3 * r4 + 2 * 14 ** 2 * r3 ** 2)) + 9 * 11 * r1 * \\
& \$ (2 * 12 ** 2 * r3 * r4 * (13 * r4 + 14 * r3) + 12 * r2 * (2 * 13 ** 2 * r4 ** 2 + 9 * 13 * 14 * r3 * r4 \\
& \$ + 2 * 14 ** 2 * r3 ** 2) + 2 * 13 * 14 * r2 ** 2 * (13 * r4 + 14 * r3)) + 2 * r1 ** 2 * (12 ** 2 * (2 * \\
& \$ 13 ** 2 * r4 ** 2 + 9 * 13 * 14 * r3 * r4 + 2 * 14 ** 2 * r3 ** 2) + 9 * 12 * 13 * 14 * r2 * (13 * r4 + \\
& \$ 14 * r3) + 2 * 13 ** 2 * 14 ** 2 * r2 ** 2)) + 4 * 14 * (11 ** 2 * (2 * 12 ** 2 * r3 * (r3 + r4) + 3 * \\
& \$ 12 * r2 * (13 * (3 * r3 + 2 * r4) + 14 * r3) + 2 * 13 * r2 ** 2 * (13 + 14)) + 3 * 11 * 12 * r1 * \\
& \$ (12 * (13 * (3 * r3 + 2 * r4) + 14 * r3) + 3 * 13 * r2 * (13 + 14)) + 2 * 12 ** 2 * 13 * r1 ** 2 * \\
& \$ (13 + 14))) + 4 * 13 * (c4 * (11 ** 2 * (2 * 12 ** 2 * r4 ** 2 + 12 * r4 * (2 * 13 * r4 + 3 * 14 * \\
& \$ (3 * r2 + 2 * r3)) + 14 * r2 * (3 * 13 * r4 + 2 * 14 * (r2 + r3))) + 3 * 11 * 14 * r1 * (3 * \\
& \$ 12 ** 2 * r4 + 12 * (3 * 13 * r4 + 14 * (3 * r2 + 2 * r3)) + 13 * 14 * r2) + 2 * 12 * 14 ** 2 * r1 ** 2 \\
& \$ * (12 + 13)) + 4 * 11 ** 2 * 12 * 14 * (12 + 13 + 14))) + 4 * 12 * (c3 * (c4 * (11 ** 2 * (2 * \\
& \$ 13 ** 2 * r4 ** 2 + 9 * 13 * 14 * r3 * r4 + 2 * 14 ** 2 * r3 ** 2) + 11 * (12 * (2 * 13 ** 2 * r4 ** 2 + 9 * \\
& \$ 13 * 14 * r3 * r4 + 2 * 14 ** 2 * r3 ** 2) + 3 * 13 * 14 * (3 * r1 + 2 * r2)) * (13 * r4 + 14 * r3)) \\
& \$ + 13 * 14 * r1 * (3 * 12 * (13 * r4 + 14 * r3) + 2 * 13 * 14 * (r1 + r2))) + 4 * 11 * 13 * 14 * \\
& \$ (13 + 14) * (11 + 12)) + 4 * c4 * 11 * 13 * 14 ** 2 * (11 + 12 + 13))) / (c2 * c3 * c4 * \\
& \$ 11 ** 2 * 12 ** 2 * 13 ** 2 * 14 ** 2) - (s1 ** 2 * (s2 ** 2 + s2 * (4 * s3 + 4 * s4) + s3 ** 2 + 4 * s3 \\
& \$ * s4 + s4 ** 2 + w3 ** 2 + w5 ** 2 + w7 ** 2) + 4 * s1 * (s2 ** 2 * (s3 + s4) + s2 * (s3 ** 2 + 4 * s3 * \\
& \$ s4 + s4 ** 2 + w5 ** 2 + w7 ** 2) + s3 ** 2 * s4 + s3 * (s4 ** 2 + w3 ** 2 + w7 ** 2) + s4 * (w3 ** 2 + \\
& \$ w5 ** 2) + s2 ** 2 * (s3 ** 2 + 4 * s3 * s4 + s4 ** 2 + w1 ** 2 + w5 ** 2 + w7 ** 2) + 4 * s2 * \\
& \$ (s3 ** 2 * s4 + s3 * (s4 ** 2 + w1 ** 2 + w7 ** 2) + s4 * (w1 ** 2 + w5 ** 2)) + s3 ** 2 * (s4 ** 2 + \\
& \$ w1 ** 2 + w3 ** 2 + w7 ** 2) + 4 * s3 * s4 * (w1 ** 2 + w3 ** 2) + s4 ** 2 * (w1 ** 2 + w3 ** 2 + \\
& \$ w5 ** 2 + w1 ** 2 * (w3 ** 2 + w5 ** 2 + w7 ** 2) + w3 ** 2 * (w5 ** 2 + w7 ** 2) + \\
& \$ w5 ** 2 * w7 ** 2)
\end{aligned}$$

C

$$FVEC(5) = ((1/8) * (c2 * (c3 * (3 * c4 * (2 * 12 ** 2 * r3 * r4 * (13 * r4 + 14 * r3) + 12 * r2 * r4 + 2 * 14 ** 2 * r3 ** 2) + 9 * 13 * 14 * r3 * r4 + 2 * 14 ** 2 * r3 ** 2) + 2 * 13 * 14 * r2 ** 2 * (13 * r4 + 14 * r3)) + 11 * r1 * (12 ** 2 * (2 * 13 ** 2 * r4 ** 2 + 9 * 13 * 14 * r3 * r4 + 2 * 14 ** 2 * r3 ** 2) + 9 * 12 * 13 * 14 * r2 * (13 * r4 + 14 * r3) + 2 * 13 ** 2 * 14 ** 2 * r2 ** 2) + 2 * 12 * 13 * 14 * r1 ** 2 * (12 * (13 * r4 + 14 * r3) + 13 * 14 * r2)) + 4 * 11 * 12 * 14 * (11 * (12 * (13 * (3 * r3 + 2 * r4) + 14 * r3) + 3 * 13 * r2 * (13 + 14)) + 3 * 12 * 13 * r1 * (13 + 14))) + 4 * c4 * 11 * 13 * 14 * (11 * (3 * 12 ** 2 * r4 + 12 * (3 * 13 * r4 + 14 * (3 * r2 + 2 * r3)) + 13 * 14 * r2) + 3 * 12 * 14 * r1 * (12 + 13))) + 4 * c3 * c4 * 12 * 13 * 14 * (3 * 11 ** 2 * (13 * r4 + 14 * r3) + 11 * (3 * 12 * (13 * r4 + 14 * r3) + 13 * 14 * (3 * r1 + 2 * r2)) + 12 * 13 * 14 * r1)) / (c2 * c3 * c4 * 11 ** 2 * 12 ** 2 * 13 ** 2 * 14 ** 2) - 2 * (s1 ** 2 * (s2 + s3 + s4) + s1 * (s2 ** 2 + 4 * s2 * (s3 + s4) + s3 ** 2 + 4 * s3 * s4 + s4 ** 2 + w3 ** 2 + w5 ** 2 + w7 ** 2) + s2 ** 2 * (s3 + s4) + s2 * (s3 ** 2 + 4 * s3 * s4 + s4 ** 2 + w1 ** 2 + w5 ** 2 + w7 ** 2) + s3 ** 2 * s4 + s3 * (s4 ** 2 + w1 ** 2 + w3 ** 2 + w7 ** 2) + s4 * (w1 ** 2 + w3 ** 2 + w5 ** 2))$$

$$\begin{aligned}
& r_2 * (2 * 13 * r_4^2 + 9 * 13 * 14 * r_3 * r_4 + 2 * 14 * r_3^2) + 2 * 13 * 14 * r_2^2 * (13 * r_4 + 14 * r_3) + 4 * 12 * 14 * (12 * (13 * (3 * r_3 + 2 * r_4) + 14 * r_3) + 3 * 13 * r_2 * (13 + 14)) + 4 * c_4 * 13 * 14 * (3 * 12 * r_4 + 12 * (3 * 13 * r_4 + 14 * (3 * r_2 + 2 * r_3))) + 13 * 14 * r_2^2 + 12 * c_3 * c_4 * 12 * 13 * 14 * (13 * r_4 + 14 * r_3) / (c_1 * c_2 * c_3 * c_4 * 11 * 12 * 13 * 14 * r_2^2) + ((1/16) * (c_2 * (c_3 * (c_4 * (2 * 11 * r_2 * r_3 * r_4 * (12 * r_3 * r_4 + r_2 * (13 * r_4 + 14 * r_3))) + 11 * r_1 * (2 * 12 * r_3 * r_4^2 + 9 * 12 * r_2 * r_3 * r_4 * (13 * r_4 + 14 * r_3) + r_2^2 * (2 * 13 * r_4^2 + 9 * 13 * 14 * r_3 * r_4 + 2 * 14 * r_3^2))) + r_1 * (2 * 12 * r_3 * r_4 * (13 * r_4 + 14 * r_3) + 12 * r_2 * (2 * 13 * r_4^2 + 9 * 13 * 14 * r_3 * r_4 + 2 * 14 * r_3^2) + 2 * 13 * 14 * r_2^2 * (13 * r_4 + 14 * r_3))) + 2 * 14 * (2 * 11 * r_2 * r_3 * (3 * 12 * r_3 * (r_3 + r_4) + r_2 * (13 * (3 * r_3 + 2 * r_4) + 14 * r_3)) + 3 * 11 * r_1 * (2 * 12 * r_3 * (r_3 + r_4) + 3 * 12 * r_2 * (13 * (3 * r_3 + 2 * r_4) + 14 * r_3) + 2 * 13 * r_2^2 * (13 + 14)) + 2 * 12 * r_1^2 * (12 * (13 * (3 * r_3 + 2 * r_4) + 14 * r_3) + 3 * 13 * r_2 * (13 + 14)))) + 2 * 13 * (c_4 * (2 * 11 * r_2 * r_4 * (12 * r_4 * (3 * r_2 + 2 * r_3) + r_2 * (13 * r_4 + 3 * 14 * (r_2 + r_3))) + 3 * 11 * r_1 * (2 * 12 * r_4^2 + 12 * r_4 * (2 * 13 * r_4 + 3 * 14 * (3 * r_2 + 2 * r_3))) + 14 * r_2 * (3 * 13 * r_4 + 2 * 14 * (r_2 + r_3))) + 2 * 14 * r_1^2 * (3 * 12 * r_4 + 12 * (3 * 13 * r_4 + 14 * (3 * r_2 + 2 * r_3))) + 4 * 11 * 14 * (11 * (12 * (3 * r_2 + 2 * (r_3 + r_4)) + r_2 * (13 + 14)) + 3 * 12 * r_1 * (12 + 13 + 14)))) + 2 * 12 * (c_3 * (c_4 * (6 * 11 * r_3 * r_4 * (13 * r_4 + 14 * r_3) + 11 * (6 * 12 * r_3 * r_4 * (13 * r_4 + 14 * r_3) + (3 * r_1 + 2 * r_2) * (2 * 13 * r_4^2 + 9 * 13 * 14 * r_3 * r_4 + 2 * 14 * r_3^2))) + r_1 * (12 * (2 * 13 * r_4^2 + 9 * 13 * 14 * r_3 * r_4 + 2 * 14 * r_3^2) + 6 * 13 * 14 * (r_1 + r_2) * (13 * r_4 + 14 * r_3))) + 4 * 14 * (11 * (12 * (13 * (3 * r_3 + 2 * r_4) + 14 * r_3) + 11 * (12 * (13 * (3 * r_3 + 2 * r_4) + 14 * r_3) + 13 * (3 * r_1 + 2 * r_2) * (13 + 14)) + 12 * 13 * r_1 * (13 + 14))) + 4 * c_4 * 13 * 14 * (3 * 11 * r_2 * r_4 + 11 * (3 * 12 * r_4 + 3 * 13 * r_4 + 14 * (3 * r_1 + 2 * (r_2 + r_3))) + 14 * r_1 * (12 + 13)))) / (c_2 * c_3 * c_4 * 11 * 12 * 13 * 14 * r_2^2) - 2 * (s_1 * (s_2 * (s_3 + s_4) + s_2 * (s_3^2 + 4 * s_3 * s_4 + s_4^2 + w_5^2 + w_7^2) + s_3 * (s_4^2 + w_3^2 + w_7^2) + s_4 * (w_3^2 + w_5^2)) + s_1 * (s_2 * (s_3^2 + 4 * s_3 * s_4 + s_4^2 + w_5^2 + w_7^2) + 4 * s_2 * (s_3 * s_4 + s_3 * (s_4^2 + w_7^2) + s_4 * w_5^2) + s_3 * (s_4^2 + w_3^2 + w_7^2) + 4 * s_3 * s_4 * w_3^2 + s_4 * (w_3^2 + w_5^2) + w_3^2 * (w_5^2 + w_7^2) + w_5^2 * w_7^2) + s_2 * (s_3 * (s_4^2 + w_1^2 + w_7^2) + 4 * s_3 * s_4 * w_1^2 + s_4 * (w_1^2 + w_5^2) + w_1 * (w_5^2 + w_7^2) + w_5^2 * w_7^2) + s_3 * (s_4^2 * (w_1^2 + w_3^2) + w_1 * (w_3^2 + w_7^2) + w_3^2 * w_7^2) + s_4 * (w_1 * (w_3^2 + w_5^2) + w_5^2 + w_3^2 * w_5^2))
\end{aligned}$$

C

$$\begin{aligned}
& FVED(6) = ((1/8) * (c_2 * (c_3 * (c_4 * (2 * 12 * r_3^2 * r_4^2 + 9 * 12 * r_2 * r_3 * r_4 * (13 * r_4 + 14 * r_3) + r_2^2 * (2 * 13 * r_4^2 + 9 * 13 * 14 * r_3 * r_4 + 2 * 14 * r_3^2))) + 2 * 14 * (2 * 12 * r_3 * (r_3 + r_4) + 3 * 12 * r_2 * (13 * (3 * r_3 + 2 * r_4) + 14 * r_3) + 2 * 13 * r_2^2 * (13 + 14))) + 2 * 13 * (c_4 * (2 * 12 * r_4^2 + 12 * r_4 * (2 * 13 * r_4 + 3 * 14 * (3 * r_2 + 2 * r_3))) + 14 * r_2 * (3 * 13 * r_4 + 2 * 14 * (r_2 + r_3))) + 4 * 12 * 14 * (12 + 13 + 14))) + 2 * 12 * (c_3 * (c_4 * (2 * 13 * r_4^2 + 9 * 13 * 14 * r_3 * r_4 + 2 * 14 * r_3^2) + 4 * 13 * 14 * (13 + 14)) + 4 * c_4 * 13 * 14 * r_2^2) / (c_1 * c_2 * c_3 * c_4 * 11 * 12 * 13 * 14 * r_2^2) + ((1/16) * (c_2 * (c_3 * (c_4 * (2 * 11 * r_2 * r_3 * r_4^2 + 9 * 11 * r_1 * r_2 * r_3 * r_4 * (12 * r_3 * r_4 + r_2 * (13 * r_4 + 14 * r_3))) + r_1 * (2 * 12 * r_3^2 * r_4^2 + 9 * 12 * r_2 * r_3 * r_4 * (13 * r_4 + 14 * r_3) + r_2^2 * (2 * 13 * r_4^2 + 9 * 13 * 14 * r_3 * r_4 + 2 * 14 * r_3^2)))
\end{aligned}$$

$$\begin{aligned}
& \$14**2*r3**2)) + 2*14*(2*11**2*r2**2*r3*(r3+r4) + 3*11*r1*r2*(3*12*r3 \\
& \$*(r3+r4) + r2*(13*(3*r3+2*r4) + 14*r3)) + r1**2*(2*12**2*r3*(r3+r4) + 3* \\
& \$12*r2*(13*(3*r3+2*r4) + 14*r3) + 2*13*r2**2*(13+14))) + 2*13*(c4*(2* \\
& \$11**2*r2*r4**2*(r2+r3) + 3*11*r1*r4*(12*r4*(3*r2+2*r3) + r2*(13*r4+3* \\
& \$14*(r2+r3))) + r1**2*(2*12**2*r4**2 + 12*r4*(2*13*r4+3*14*(3*r2+2*r3) \\
& \$) + 14*r2*(3*13*r4+2*14*(r2+r3)))) + 2*14*(2*11**2*r2*(r2+r3+r4) + 3*11 \\
& \$*r1*(12*(3*r2+2*(r3+r4)) + r2*(13+14)) + 2*12*r1**2*(12+13+14))) + 2* \\
& \$12*(c3*(c4*(2*11**2*r3**2*r4**2 + 11*r3*r4*(2*12*r3*r4+3*(3*r1+2* \\
& \$r2)*(13*r4+14*r3)) + r1*(3*12*r3*r4*(13*r4+14*r3) + (r1+r2)*(2*13**2* \\
& \$r4**2 + 9*13*14*r3*r4+2*14**2*r3**2))) + 2*14*(2*11**2*r3*(r3+r4) + 11* \\
& \$*(2*12*r3*(r3+r4) + (3*r1+2*r2)*(13*(3*r3+2*r4) + 14*r3)) + r1*(12*(13* \\
& \$*(3*r3+2*r4) + 14*r3) + 2*13*(r1+r2)*(13+14)))) + 2*13*(c4*(2*11**2* \\
& \$r4**2 + 11*r4*(2*12*r4+2*13*r4+3*14*(3*r1+2*(r2+r3)))) + 14*r1*(3*12* \\
& \$r4+3*13*r4+2*14*(r1+r2+r3))) + 4*11*14*(11+12+13+14))) / (c2*c3*c4* \\
& \$11**2*12**2*13**2*14**2) - (s1**2*(s2**2*(s3**2+4*s3*s4+s4**2+w5**2 \\
& \$+w7**2) + 4*s2*(s3**2*s4+s3*(s4**2+w7**2) + s4*w5**2) + s3**2*(s4**2+ \\
& \$w3**2+w7**2) + 4*s3*s4*w3**2+s4**2*(w3**2+w5**2) + w3**2*(w5**2+w7**2) \\
& \$+w5**2*w7**2) + 4*s1*(s2**2*(s3**2*s4+s3*(s4**2+w7**2) + s4*w5**2) + s2 \\
& \$*(s4**2+w7**2)*(s3**2+w5**2) + w3**2*(s3**2*s4+s3*(s4**2+w7**2) + s4* \\
& \$w5**2) + s2**2*(s3**2*(s4**2+w1**2+w7**2) + 4*s3*s4*w1**2+s4**2*(\\
& \$w1**2+w5**2) + w1**2*(w5**2+w7**2) + w5**2*w7**2) + 4*s2*w1**2*(s3**2* \\
& \$s4+s3*(s4**2+w7**2) + s4*w5**2) + s3**2*(s4**2*(w1**2+w3**2) + w1**2*(\\
& \$w3**2+w7**2) + w3**2*w7**2) + 4*s3*s4*w1**2*w3**2+s4**2*(w1**2*(w3**2 \\
& \$+w5**2) + w3**2*w5**2) + w1**2*(w3**2*(w5**2+w7**2) + w5**2*w7**2) + \\
& \$w3**2*w5**2*w7**2)
\end{aligned}$$

C

$$\begin{aligned}
& FVEC(7) = ((1/8)*(c2*(c3*r2*(3*c4*r3*r4*(12*r3*r4+r2*(13*r4+ \\
& \$14*r3)) + 2*14*(3*12*r3*(r3+r4) + r2*(13*(3*r3+2*r4) + 14* \\
& \$r3))) + 2*13*(c4*r4*(12*r4*(3*r2+2*r3) + r2*(13*r4+3*14*(r2+r3))) \\
& \$+ 2*14*(12*(3*r2+2*(r3+r4)) + r2*(13+14))) + 2*12*(c3*(3*c4*r3*r4 \\
& \$*(13*r4+14*r3) + 2*14*(13*(3*r3+2*r4) + 14*r3)) + 6*c4*13*14*r4) / \\
& \$ (c1*c2*c3*c4*11*12**2*13**2*14**2) + ((1/16)*(c2*r1*(c3*r2*(3*c4* \\
& \$r3*r4*(11*r2*r3*r4+r1*(12*r3*r4+r2*(13*r4+14*r3))) + 2*14*(3*11* \\
& \$r2*r3*(r3+r4) + r1*(3*12*r3*(r3+r4) + r2*(13*(3*r3+2*r4) + 14*r3 \\
& \$)))) + 2*13*(c4*r4*(3*11*r2*r4*(r2+r3) + r1*(12*r4*(3*r2+2*r3) + r2 \\
& \$*(13*r4+3*14*(r2+r3)))) + 2*14*(3*11*r2*(r2+r3+r4) + r1*(12*(3*r2 \\
& \$+ 2*(r3+r4) + r2*(13+14)))) + 2*12*(c3*(c4*r3*r4*(11*r3*r4* \\
& \$*(3*r1+2*r2) + r1*(12*r3*r4+3*(r1+r2)*(13*r4+14*r3))) + 2*14* \\
& \$*(11*r3*(r3+r4)*(3*r1+2*r2) + r1*(12*r3*(r3+r4) + (r1+r2)*(13*(3* \\
& \$r3+2*r4) + 14*r3))) + 2*13*(c4*r4*(11*r4*(3*r1+2*(r2+r3)) + r1* \\
& \$*(12*r4+13*r4+3*14*(r1+r2+r3))) + 2*14*(11*(3*r1+2*(r2+r3+r4)) \\
& \$+ r1*(12+13+14)))) / (c2*c3*c4*11**2*12**2*13**2*14**2) - 2*(s1**2 \\
& \$*(s2**2*(s3**2*s4+s3*(s4**2+w7**2) + s4*w5**2) + s2*(s4**2+w7**2) \\
& \$*(s3**2+w5**2) + w3**2*(s3**2*s4+s3*(s4**2+w7**2) + s4*w5**2)) + s1
\end{aligned}$$

$$\begin{aligned}
 & \$*(s4**2+w7**2)*(s3**2+w5**2)*(s2**2+w3**2)+w1**2*(s2**2*(\\
 & \$s3**2*s4+s3*(s4**2+w7**2)+s4*w5**2)+s2*(s4**2+w7**2)*(s3**2+ \\
 & \$w5**2)+w3**2*(s3**2*s4+s3*(s4**2+w7**2)+s4*w5**2))
 \end{aligned}$$

C

$$\begin{aligned}
 \text{FVEC}(8) = & ((1/8)*(c2*r2*(c3*r2*r3*(c4*r3*r4**2+2*14*(r3+r4))+ \\
 & 2*13*(c4*r4**2*(r2+r3)+2*14*(r2+r3+r4)))+2*12*(c3*r3*(c4*r3* \\
 & r4**2+2*14*(r3+r4))+2*13*(c4*r4**2+2*14))) / (c1*c2*c3*c4*11*12**2 \\
 & *13**2*14**2) + ((1/16)*r1*(c2*r1*r2*(c3*r2*r3*(c4*r3*r4**2+2*14* \\
 & (r3+r4))+2*13*(c4*r4**2*(r2+r3)+2*14*(r2+r3+r4)))+2*12*(c3*r3*(r1+ \\
 & r2)*(c4*r3*r4**2+2*14*(r3+r4))+2*13*(c4*r4**2*(r1+r2+r3)+2*14* \\
 & (r1+r2+r3+r4)))) / (c2*c3*c4*11**2*12**2*13**2*14**2) - (s4**2+w7**2) \\
 & *(s3**2+w5**2)*(s2**2+w3**2)*(s1**2+w1**2)
 \end{aligned}$$

C

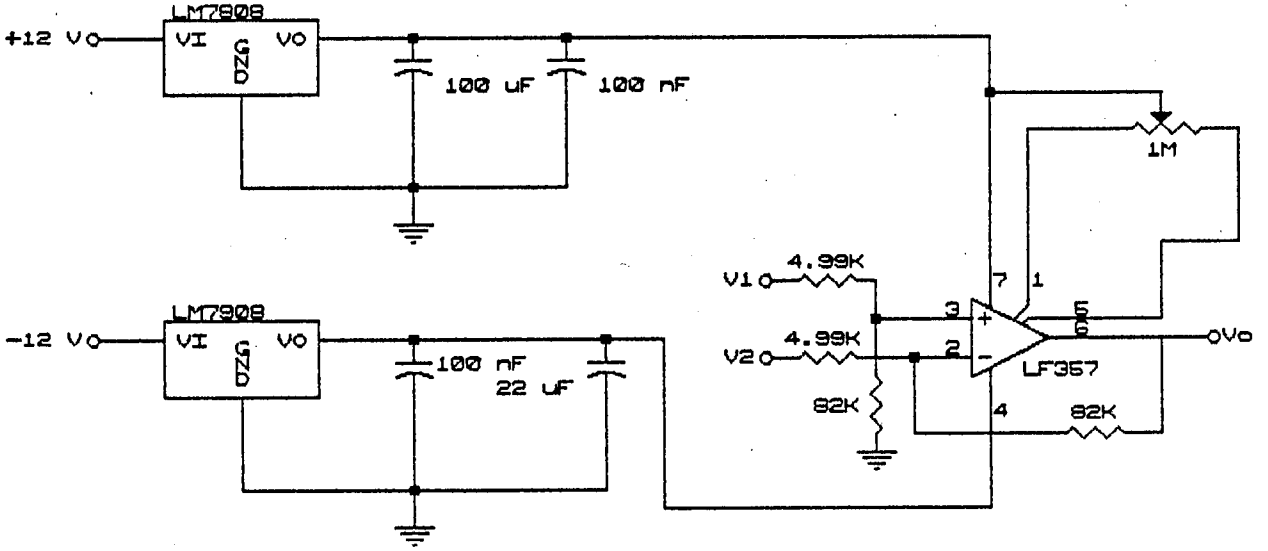
RETURN

END

BİLGİSAYARA SES VERİLERİNİN ALINMASINDA KULLANILAN DEVRELER

1-Ön Kuvvetlendirici Devresi

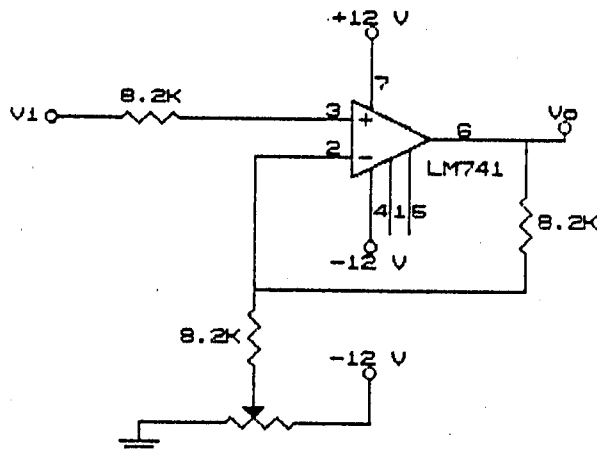
Ses sinyalinin genliğini büyütürken filtre devresi için hazır hale getiren devredir. Bu devrede işlemsel kuvvetlendirici (OPAMP) olarak LF357 entegresi, +8 volt için LM7808 ve -8 volt için LM7908 voltaj regülatörleri kullanılmıştır. Ön kuvvetlendirici devresi aşağıda verilmiştir.



Ön kuvvetlendirici devresi.

2-Seviye Kaydırıcı Devre

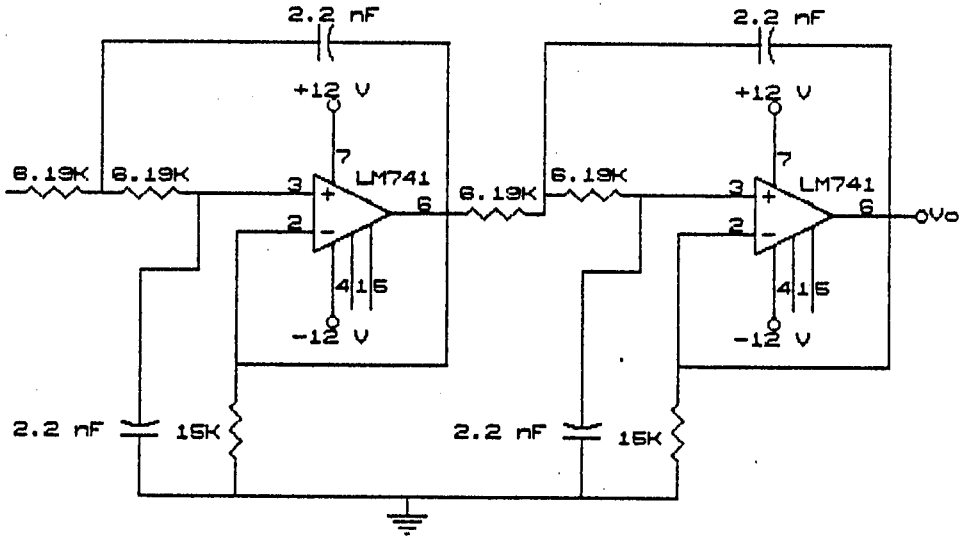
ADDA kartının girişi 0-9V arası olması gerektiğinden ses sinyaline 4 volt'luk bir d.a. seviye eklenir. Böylece örneklenen sinyalin minimum değeri bile pozitif olur. Elde edilen örneklenmiş değerler kullanılmadan önce 4 volt'a karşılık gelen değer çıkarılarak gerçek değerler bulunur. Sinyale d.a. seviyesi kazandıran devre aşağıda verilmiştir.



Seviye Kaydırıcı devre.

3-Alçak Geçiren Filtre

Analog ses sinyalinin doğru bir şekilde sayısal çevrilebilmesi için yüksek frekanslı gürültü bileşenlerinin yok edilmesi gerekir. Frekansı 5 kHz ve altında olan sinyalleri geçirmesi istenen filtre devresi aşağıda verildiği gibi tasarlanmıştır. Devrede OPAMP olarak LM741 entegresi kullanılmıştır. Devre, toplam kazancı 1 ve kesme frekansı 5 kHz olacak şekilde ayarlanan 4'ncü dereceden bir filtre devresidir.



Alçak geçiren filtre devresi.

4-ADDA Kartının Özellikleri

IBM PC uyumlu 12 bitlik ADDA kartı, hassasiyeti yüksek olan veri çevrim sistemidir. Kart analogtan dijitale ve dijitalden analoga çevrim yapmaktadır ve şu özelliklere sahiptir.

Dijitalden analoga (D/A):

- 12 bit, 1 kanal
- Çıkış voltajı 0-9 Volt arası
- Akım ayarlama (current setting) zamanı 500 ns
- Lineerlikten uzak oluşu (nonlinearity) %0.2'dir.

Analogtan dijitale (A/D):

- 12 bit, 16 kanal
- Giriş voltajı 0-9 Volt arası

- Dizi yaklařtırma (succesion approximation) yöntemine göre çalışır.
- Her bir kanalın çevirme zamanı 30 us'dir.

ADDA kartının katalogunda örnekleme aralığının 30 us ve hassas olduđu belirtilmesine rağmen, 50 us civarındaki örnekleme aralıkları için denemeler yapılmıř ve örneklenen deđerlerin hassas olmadığı gözlenmiřtir. Özellikle 4-5 V arası aynı deđerleri vermesi dikkat çekmiřtir. Hem örnekleme aralığının küçük tutulması ve hem de deđerlerin hassas olması gerektiğinden en uygun örnekleme aralığı olarak 100 us (10 kHz) seçilmiřtir ve bütün sinyaller 100 us'de örneklenmiřtir.

TUM KUTUP MODELINDEN ENINE KESİT ALANLARI BULAN PROGRAM

```

load d7;
load d8;
l=length(DT1);
pik=DT1(3)-DT1(1)+1;
DT=DT1;
p=DT1(2)-DT1(1)+1
clear DT1;
for i=1:l-3,
    DT1(i)=DT(i+3);
end;
sayac=1;
l=length(DT1)
N=input('INPUT NUMBER OF CHANNELO DATA...:');
for i=1:N,
    V1(i,1)=2.119706319e-3*DT1(i)+.078720725;
    SES1(i,1)=V1(i,1)-3.79;
end;
for i=2:N,
    SESP(i,1)=SES1(i,1)-SES1(i-1,1);
end;
SESP(1,1)=SES1(1,1);
NN=input('NUMBER OF CHANNEL1 DATA...:');
for i=1:NN,
    V2(i,1)=2.119706319e-3*DT2(i)+.078720725;
    SES2(i,1)=V2(i,1)-3.78;
end;
clg
subplot(211),plot(SES1);pause
subplot(212),plot(SES2);pause
meta ses12
%=====
peak=p+pik;
peak1=p+peak;
peak2=p+peak1;
saya=1;
yy=NN-p;
for i=1:NN,
    % *** girtlak icin girdi ***
if i<=p,
    y=p;
    if i<pik,
        U1(i,1)=(((DT2(pik)-DT2(sayac))*i)-(DT2(pik)*sayac)+

```

```

        (DT2(sayac)*pik))/(pik-sayac);
    else
    U1(i,1)=(((DT2(pik)-DT2(y))*i)-(DT2(pik)*y)+(DT2(y)*pik))/(pik-y);
    end;
else
sayac=p+1;
y=NN;
    if i<peak,
        U1(i,1)=(((DT2(pik)-DT2(sayac))*(i-p))-(DT2(pik)*saya)+
            (DT2(sayac)*pik))/(peak-sayac);
    else
        U1(i,1)=(((DT2(pik)-DT2(y))*(i-p))-(DT2(pik)*yy)+
            (DT2(y)*pik))/(peak-y);
    end;
end;
%   if i==pik,
%       U1(i,1)=DT2(pik);U(i,1)=2.119706319e-3*U1(i)+.078720725-4;
%   else
%       U1(i,1)=0;U(i,1)=0;
%   end;
    U(i,1)=2.119706319e-3*U1(i,1)+.078720725-3.78;
end;
clg
subplot(211),plot(U);pause
subplot(212),plot(U1);pause
meta ainput
N=NN-1;
    for i=N:-1:1,
        for j=1:1,
            SES(i,j)=SESP(i+1-j);
        end;
    end;
m=1;
disp('          !.....WORKING.....! ')
n=10;
fprintf(' (m,n)=(%g,%g)\n',m,n)
    for i=N:-1:1,
        for j=m:-1:1,
            if (i-j)<0,
                H(i,j)=0;
            else
                H(i,j)=SES2(i+2-j);
            end;
        end;
    end;
end;

```



```

end;
for i=N:-1:1,
  for j=n:-1:1,
    if (i-j)<0,
      L(i,j)=0;
    else
      L(i,j)=SES(i+1-j);
    end;
  end;
end;
A=[H,L];
for i=N:-1:1,
  for j=1:1,
    SESX(i,j)=SESP(i+2-j);
  end;
end;
P=((inv(A'*A))*A')*(SESX);
PRMTR=P';
end;
m3=1;
n3=10;
PYD=-PRMTR(:,m3+1:m3+n3);
PAYDA=[1 PYD];
B=[zeros(1:n3+1)];
B=PAYDA
M=n3;
for j=1:M,
  MR=M+1-j;
  MRP1=MR+1;
  D=1-B(MRP1)*B(MRP1);
  RC(MR)=B(MRP1);
  for k=1:MR,
    MM=MR+2-k;
    C(k)=B(MM);
  end;
  for k=1:MR,
    B(k)=(B(k)-B(MRP1)*C(k))/D;
  end;
end;
end;
AREA1(n3+1)=1;
for i=M:-1:1,
  AR=(1-RC(i))/(1+RC(i));
  AREA1(i)=AREA1(i+1)/AR;
end;

```

```
AREA1
disp(' ')
pause
fname=input('enter filename:','s');
disp('master dosya icin (1) giriniz')
disp('kendi dosyaniz icin (2) giriniz')
select=input('seciminiz(1/2)..:');
    if select==1,
        harf=input('hangi harfi kayit edeceksiniz:','s');
        harf=[harf,'\n'];
        fprintf(fname,harf);
        fprintf(fname,'%g\n',n3+1);
        for i=1:n3+1,
            fprintf(fname,'%g\n',AREA1(i));
        end;
    end;
    if select==2,
        fprintf(fname,'%g\n',n3+1);
        for i=1:n3+1,
            fprintf(fname,'%g\n',AREA1(i));
        end;
    end;
! H/U
!BILGIN1
! H/M
```

EŞİTLİK (4.37)'YE DAYANAN KUTUP SIFIR MODELİNDEN
ENİNE KESİT ALANLARI BULAN PROGRAM

```

load d15;
load d16;
l=length(DT1);
% sayac=DT1(1);
% y=DT1(2);
% peak=DT1(3);
pik=DT1(3)-DT1(1)+1;
DT=DT1;
p=DT1(2)-DT1(1)+1
clear DT1;
for i=1:l-3.
    DT1(i)=DT(i+3);
end;
sayac=1;
l=length(DT1)
N=input('INPUT NUMBER OF CHANNEL DATA...:');
for i=1:N,
%   if i==a,
% SMP1(i,1)=(DT1(i-x)-DT1(i-x-1))/2+DT1(i-x-1);
%       x=x+1;a=a+5;
%   else
%       SMP1(i,1)=DT1(i-x);
%   end;
%=====
    V1(i,1)=2.119706319e-3*DT1(i)+.078720725;
    SES1(i,1)=V1(i,1)-3.79;
end;
for i=2:N.
    SESP(i,1)=SES1(i,1)-SES1(i-1,1);
end;
SESP(1,1)=SES1(1,1);
for m=1:N,
    w(m,1)=0.54-(0.46*cos(2*pi*m/N));
end;
for i=1:N,
    SESW(i,1)=SESP(i,1)*w(N+1-i,1);
end;
%=====
NN=input('NUMBER OF CHANNEL DATA...:');
%   j=1;
% for i=1:NN,
%   SMP2(j,1)=DT2(i);

```

```

%   j=j+5;
% end;
%   k=1;
% for x=1:NN-1,
%   for i=1:4,
%       SMP2(i+k,1)=(SMP2(k+5)-SMP2(k))*i/5+SMP2(k);
%   end;
%       k=k+5;
% end;
%=====
for i=1:NN,
    V2(i,1)=2.119706319e-3*DT2(i)+.078720725;
    SES2(i,1)=V2(i,1)-3.78;
    %DT21(i,1)=DT2(i,1)-1855;
end;
%clc
%subplot(211),plot(SES1);pause
%subplot(212),plot(SES2);pause
%meta girdukad
pr92
N=input('Ornek sayisi.....:');clc
    for i=N:-1:1,
        for j=1:1,
            SES(i,j)=SESP(i+1-j);
        end;
    end;
SR=-10;MDL=100;
TN=1000;AIC=100;
for m=12:20,
    for n=m:m,
        disp('          !.....WORKING.....! ')
        fprintf(' (m,n)=(%g,%g)\n',m,n)
        for i=N:-1:1,
            for j=m+1:-1:1,
                if (i-j)<-1,
                    H(i,j)=0;
                else
                    H(i,j)=SES2(i+2-j);
                end;
            end;
        end;
    end;
for i=N:-1:1,
    for j=n:-1:1,
        if (i-j)<0,

```

```

        L(i,j)=0;
    else
        L(i,j)=SES(i+1-j);
    end;
end;
end;
end;
A=[H,L];
clear H,clear L;
for i=N:-1:1,
    for j=1:1,
        SESX(i,j)=SESP(i+2-j);
    end;
end;
P=((inv(A'*A))*A')*(SESX);
% PRMTR=P';
for i=1:N,
    for j=n:-1:1,
        if (i-n-1+j)<0,
            H1(i,n-j+1)=0;
        else
            H1(i,n-j+1)=SES(i-n+j);
        end;
    end;
end;
for i=1:N,
    for j=m+1:-1:1,
        if (i-m-1+j)<1,
            L1(i,m-j+2)=0;
        else
            L1(i,m-j+2)=SES2(i-m+j-1);
        end;
    end;
end;
end;
S1=[L1 H1];
clear H1,clear L1;
W=SESX-S1*P;
for i=1:n+m,
    for j=1:n+m,
        if i==j,
            PO(i,j)=100;
        else
            PO(i,j)=0;
        end;
    end;
end;
end;

```

```

end;
K=PO*(S1(1,1:n+m)')*(inv(S1(1,1:n+m)*PO*(S1(1,1:n+m)')+1));
for i=1:N,
    if i==1;W(i)=SESX(i);r(i)=(W(i)^2)/i;Q1=K*SESX(i+1);
P1=PO-(PO*(S1(i,1:n+m)')*(inv(S1(i,1:n+m)*PO*(S1(i,1:n+m)')+1))*S1(i,1:n+m)*PO);
    K1=P1*(S1(i+1,1:n+m)')*(inv(S1(i+1,1:n+m)*P1*(S1(i+1,1:n+m)')+1));
    else W(i)=SESX(i)-S1(i-1,1:n+m)*Q1;
    r(i)=(((i-1)/i)*r(i-1))+((W(i)^2)/i);
    if i==N,break;
    end;
    Q2=Q1+(K1*(SESX(i+1)-S1(i,1:n+m)*Q1));
    Q1=Q2;
    P2=P1-(P1*(S1(i,1:n+m)')*(inv(S1(i,1:n+m)*P1*(S1(i,1:n+m)')+1))*S1(i,1:
    P1=P2;
    K2=P2*(S1(i+1,1:n+m)')*(inv(S1(i+1,1:n+m)*P2*(S1(i+1,1:n+m)')+1));
    K1=K2;
end;
end;
for i=1:N,
    if i==1;b(i)=(SESX(i)^2)/i;
    else b(i)=(((i-1)/i)*b(i-1))+((SESX(i)^2)/i);
    end;
end;
end;
V=SESX-A*P;
V2=V'*V;
t=((V1-V2)/V2)*(N-(n+m));
t1=abs(t);
fprintf('t1 degeri...:%g\n',t1)
V1=V2;
fprintf('SSE degeri...:%g\n',V1)
if t1<3,break;
end;
if V1<TN,
    m3=m;n3=n;
    TN=V1;PRMTR=P';PRMTR1=Q1';
end;
SRR=10*log10(b(N)/r(N));
if SRR>SR,
    m1=m;n1=n;
    SR=SRR;
end;
fprintf('SRR degeri...:%g\n',SRR)
AC=(N*log(r(N)))+(n+m)*2;
if AC<AIC,

```

```

    m2=m;n2=n;
    AIC=AC;
end;
fprintf('AIC degeri...:%g\n',AC)
ML=(N*log(r(N)))+(n+m)*log(N);
if ML<MDL,
    m4=m;n4=n;
    MDL=ML;
end;
fprintf('MDL degeri...:%g\n',ML)
clear P0,clear P1,clear K,clear K1,clear Q1,clear P,clear S1,clear W,clear r;
clear Q2,clear P2,clear K2,clear b,clear V,clear V2,clear t;
end;
if t1<3,break;
end;
end;
clear t1;
PAY=PRMTR(:,1:m3+1);
PYD=-PRMTR(:,m3+2:m3+n3+1);
PAYDA=[1 PYD];
clear PRMTR,clear PYD;
% PAY1=PRMTR1(:,1:m3)
% PYD1=-PRMTR1(:,m3+1:m3+n3)
zero=roots(PAY)
r=0;
for i=1:m3,
    if imag(zero(i))~=0,
        zer(i-r)=zero(i);
        y=i-r;
    else
        r=r+1;
    end;
end;
x=1;
u=0;
for i=1:y-1,
    if imag(zer(i))== -imag(zer(i+1)),
        o(x)=i;
        u=1;
        x=x+1;
    else
        o(x)=i+u;
        x=x+1;
    end;
end;

```

```

end;
clear r,clear i,clear y,clear u,clear zero;
u=0;
for i=1:x-2,
    if o(i)==o(i+1),
        u=u+1;
    else
        ze(i-u)=abs(imag(zer(o(i)))));
    end;
end;
ze(x-1-u)=abs(imag(zer(o(x-1)))));
clear o,clear i,clear zer;
s=x-1-u;
t=0;
for i=1:s,
    zero(i)=min(ze);
    if i==s,break;
end;
for j=1:s-t,
    l=s-t;
    if zero(i)==ze(j),
        k=j;
        if k==1,
            zef=ze(k+1:s-t);
        else
            if k==1,
                zef=ze(1:k-1);
            else
                zel=ze(1:k-1);
                ze2=ze(k+1:s-t);
                zef=[zel,ze2];
            end;
        end;
    end;
end;
t=i;
ze=zef;
end;
m3=s
clear u,clear x,clear ze,clear s,clear t,clear i,clear j,clear l,clear k;
clear zef,clear zel,clear ze2;
zero
pole=roots(PAYDA)
r=0;

```



```

for i=1:n3,
    if imag(pole(i))~=0,
        pol(i-r)=pole(i);
        y=i-r;
    else
        r=r+1;
    end;
end;
x=1;
u=0;
for i=1:y-1,
    if imag(pol(i))==-imag(pol(i+1)),
        o(x)=i;
        u=1;
        x=x+1;
    else
        o(x)=i+u;
        x=x+1;
    end;
end;
clear r,clear i,clear y,clear u,clear pole;
u=0;
for i=1:x-2,
    if o(i)==o(i+1),
        u=u+1;
    else
        po(i-u)=abs(imag(pol(o(i)))));
    end;
end;
po(x-1-u)=abs(imag(pol(o(x-1)))));
clear o,clear i,clear pol;
s=x-1-u;
t=0;
for i=1:s,
    pole(i)=min(po);
    if i==s, break;
end;
for j=1:s-t,
    l=s-t;
    if pole(i)==po(j);
        k=j;
        if k==1,
            pof=po(k+1:s-t);
        else

```

```

        if k==1,
            pof=po(1:k-1);
        else
            po1=po(1:k-1);
            po2=po(k+1:s-t);
            pof=[po1,po2];
        end;
    end;
end;
end;
t=i;
po=pof;
end;

clear u,clear x,clear po,clear t,clear i,clear j,clear l,clear k;
clear pol,clear po1,clear po2;
pole
clear PAY,clear PAYDA;
fprintf('SRR in maximum oldugu (m,n) degeri (%g,%g) dir.\n',m1,n1)
disp(' ')
fprintf('AIC nin minimum oldugu (m,n) degeri (%g,%g) dir.\n',m2,n2)
disp(' ')
fprintf('MDL in minimum oldugu (m,n) degeri (%g,%g) dir.\n',m4,n4)
disp(' ')
fprintf('SSE nin minimum oldugu (m,n) degeri (%g,%g) dir.\n',m3,n3)
disp(' ')

n3=s
clear s;
disp('          !....WORKING....!          ')
L=17;
h=0.5;
se=min(n3,m3);
for i=1:2:2*se+14,
    a(i)=0.5;
end;
for i=2:2:2*se+14,
    a(i)=-0.5;
end;
a1=a;
a2=a;
z=2;
of=2*se+14;
c=35000;
p=0.00114;
w1=2*pi*25;

```

```

w2=2*pi*4000;
for su=1:100,
clear A;
j=sqrt(-1);
for i=1:L/h,
b=0;
for y=1:2*se,
b=b+(a(y)*cos((y*pi*(i-0.5)*h)/L));
end;
A(i)=exp(b);
end;
if z==2*se,
break;
end;
clear wo,clear wfark;
for w=w1:w1:w2,
NS=[1,0;0,1];
for i=1:L/h,
R(i)=cos(w*h/c);
B(i)=((j*p*c)/A(i))*(sin(w*h/c));
C(i)=((j*A(i))/(p*c))*(sin(w*h/c));
D(i)=cos(w*h/c);
M=[R(i),B(i);C(i),D(i)];
NS=NS*M;
end;
f=w/(2*pi*25);
PT(f)=NS(1,1)/NS(2,1);
ZT(f)=1/NS(2,1);
end;
%clg
%subplot(211),plot(PT);pause
%subplot(212),plot(ZT);pause
%meta girdukap
clear R,clear B,clear C,clear D,clear M,clear NS,clear b;
n=1;
for i=1:f-1,
car=PT(i+1)*PT(i);
if car>=0,
w4p(n)=(i+1)*2*pi*25;
w3p(n)=2*i*pi*25;
g(n)=i;
n=n+1;
end;
end;

```

```

end;
clear car;
m=1;
for i=1:f-1,
    car=ZT(i+1)*ZT(i);
    if car>=0,
        w4z(m)=(i+1)*2*pi*25;
        w3z(m)=2*i*pi*25;
        l(m)=i;
        m=m+1;
    end;
end;
clear car;
for i=1:n-1,
    wcp(i)=w3p(i)-(PT(g(i))*((w4p(i)-w3p(i))/(PT(g(i)+1)-PT(g(i))))));
end;
for r=1:m-1,
    wcz(r)=w3z(r)-(ZT(l(r))*((w4z(r)-w3z(r))/(ZT(l(r)+1)-ZT(l(r))))));
end;
if m<n,
    wcz(n-1)=wcp(n-1)+((wcp(n-1)-wcp(n-2))/2);
    if m<n-1,
        wcz(n-2)=wcp(n-2)+((wcp(n-1)-wcp(n-2))/2);
        if m<n-2,
            wcz(n-3)=wcp(n-3)+((wcp(n-2)-wcp(n-3))/2);
        end;
    end;
end;
clear PT,clear ZT,clear g,clear l,clear w3p,clear w4p,clear w3z,clear w4z,
clear f,clear r;
T=100E-6;
for i=1:n-1,
    wcp(i)=j*wcp(i);wcz(i)=j*wcz(i);
end;
for i=1:n-1,
    v(i)=sqrt(exp(2*real(wcp(i))*T)/(1+(tan(imag(wcp(i))*T))^2));
    d(i)=v(i)*tan(imag(wcp(i))*T);
    wcp1(i)=v(i)+j*d(i);
    wcp2(i)=abs(imag(wcp1(i)));
end;
for i=1:n-1,
    e(i)=sqrt(exp(2*real(wcz(i))*T)/(1+(tan(imag(wcz(i))*T))^2));
    f(i)=e(i)*tan(imag(wcz(i))*T);
    wcz1(i)=e(i)+j*f(i);

```

```

        wcz2(i)=abs(imag(wcz1(i)));
    end;
clear v,clear d,clear e,clear f,clear wcp,clear wcz,clear T;
s=n-1;
t=0;
for i=1:s,
    wcp(i)=min(wcp2);
    if i==s, break;
end;
    for j=1:s-t,
        l=s-t;
        if wcp(i)==wcp2(j),
            k=j;
            if k==1,
                wcpf=wcp2(k+1:s-t);
            else
                if k==1,
                    wcpf=wcp2(1:k-1);
                else
                    wc1p=wcp2(1:k-1);
                    wc2p=wcp2(k+1:s-t);
                    wcpf=[wc1p,wc2p];
                end;
            end;
        end;
    end;
end;
t=i;
wcp2=wcpf;
end;
clear wc1p,clear wcp2,clear s,clear t,clear i,clear j,clear l,clear k,clear
wcpf;
clear wc1p,clear wc2p;
s=n-1;t=0;
for i=1:s,
    wcz(i)=min(wcz2);
    if i==s, break;
end;
    for j=1:s-t,
        l=s-t;
        if wcz(i)==wcz2(j),
            k=j;
            if k==1,
                wczf=wcz2(k+1:s-t);
            else

```

```

    if k==1,
        wczf=wcz2(1:k-1);
    else
        wc1z=wcz2(1:k-1);
        wc2z=wcz2(k+1:s-t);
        wczf=[wc1z,wc2z];
    end;
end;
end;
end;
t=i;
wcz2=wczf;
end;
clear wcz1,clear wcz2, clear s,clear t,clear i,clear j,clear k,clear l;
clear wczf,clear wc1z,clear wc2z;
if (2*n-2)==2*se,
    if n3<=m3,
        for i=1:2:2*n3,
            j=(i+1)/2;
            wt(i)=pole(j);
        end;
        for i=2:2:2*n3,
            j=i/2;
            wt(i)=zero(j);
        end;
    else
        for i=1:2:2*m3,
            j=(i+1)/2;
            wt(i)=pole(j);
        end;
        for i=2:2:2*m3,
            j=i/2;
            wt(i)=zero(j);
        end;
    end;
end;
if (2*n-2)~=2*se,
    clear a2;
    a2=a;
    clear pole,clear zero,clear wt;
    n=2*(n-1);
    q=n;
    if (2*n)<2*se,
        clear of;

```

```

    of=2*se;
    clear a1;
    a1=a;
    end;
for i=N:-1:1,
    for j=q+1:-1:1,
        if (i-j)<-1,
            .. H1(i,j)=0;
        else
            H1(i,j)=SES2(i+2-j);
        end;
    end;
end;
for i=N:-1:1,
    for j=n:-1:1,
        if (i-j)<0,
            L1(i,j)=0;
        else
            L1(i,j)=SES(i+1-j);
        end;
    end;
end;
A1=[H1,L1];
P=((inv(A1'*A1))*A1')*(SESX);
PRMTR=P';
clear H1,clear L1,clear A1,clear P;
PAY=PRMTR(:,1:q+1);
PYD=-PRMTR(:,q+2:q+n+1);
PAYDA=[1 PYD];
clear PRMTR,clear PYD;
zero=roots(PAY)
if all(imag(zero)==0),
    clear zero;
    for i=1:n,
        zero(i)=0;
    end;
    s=n/2;
else
    r=0;
    for i=1:q,
        if imag(zero(i))~=0;
            zer(i-r)=zero(i);
            y=i-r;
        else

```

```

    r=r+1;
end;
end;
x=1;
u=0;
for i=1:y-1,
    if imag(zer(i))==-imag(zer(i+1)),
        o(x)=i;
        u=1;
        x=x+1;
    else
        o(x)=i+u;
        x=x+1;
    end;
end;
clear r,clear i,clear y,clear u,clear zero;
u=0;
for i=1:x-2,
    if o(i)==o(i+1),
        u=u+1;
    else
        ze(i-u)=abs(imag(zer(o(i)))));
    end;
end;
ze(x-1-u)=abs(imag(zer(o(x-1)))));
clear o,clear i,clear zer;
s=x-1-u;
t=0;
for i=1:s,
    zero(i)=min(ze);
    if i==s, break;
end;
for j=1:s-t,
    l=s-t;
    if zero(i)==ze(j),
        k=j;
        if k==1,
            zef=ze(k+1:s-t);
        else
            if k==1,
                zef=ze(1:k-1);
            else
                zel=ze(1:k-1);
                ze2=ze(k+1:s-t);
            end;
        end;
    end;
end;

```



```

        zef=[ze1,ze2];
    end;
    end;
    end;
    end;
    t=i;
    ze=zef;
    end;
    end;
m4=s;
zero
clear u,clear x,clear ze,clear s,clear t,clear i,clear j,clear l,clear k;
clear zef,clear ze1,clear ze2;
pole=roots(PAYDA)
if all(imag(pole)==0),
    clear pole;
    for i=1:n,
        pole(i)=0;
    end;
    s=n/2;
else
    r=0;
    for i=1:q,
        if imag(pole(i))~=0,
            pol(i-r)=pole(i);
            y=i-r;
        else
            r=r+1;
        end;
    end;
    end;
x=1;
u=0;
for i=1:y-1,
    if imag(pol(i))== -imag(pol(i+1)),
        o(x)=i;
        u=1;
        x=x+1;
    else
        o(x)=i+u;
    end;
end;

```

```

    x=x+1;
end;
end;
clear r,clear i,clear y,clear u,clear pole;
u=0;
for i=1:x-2,
    if o(i)==o(i+1),
        u=u+1;
    else
        po(i-u)=abs(imag(pol(o(i)))));
    end;
end;
    po(x-1-u)=abs(imag(pol(o(x-1)))));
clear o,clear i,clear pol;
s=x-1-u;
t=0;
for i=1:s,
    pole(i)=min(po);
    if i==s, break;
end;
    for j=1:s-t,
        l=s-t;
        if pole(i)==po(j),
            k=j;
            if k==1,
                pof=po(k+1:s-t);
            else
                if k==1,
                    pof=po(1:k-1);
                else
                    po1=po(1:k-1);
                    po2=po(k+1:s-t);
                    pof=[po1,po2];
                end;
            end;
        end;
    end;
end;
t=i;
po=pof;
end;
end;
n4=s;
clear s,clear u,clear x,clear po,clear t,clear i,clear j,clear l,clear k;
clear pol,clear po1,clear po2;

```

```

pole
clear PAY,clear PAYDA,clear a;
if (n4+m4)>(2*se),
    for i=1:2*se,
        a(i)=a2(i);
    end;
    for i=2*n3+1:of,
        a(i)=a1(i);
    end;
    if n4<=m4
        for i=of+1:2:2*n4,
            a(i)=0.5;
        end;
        for i=of+2:2:2*n4,
            a(i)=-0.5;
        end;
    else
        for i=of+1:2:2*m4,
            a(i)=0.5;
        end;
        for i=of+2:2:2*m4,
            a(i)=-0.5;
        end;
    end;
end;
else
    if n4<=m4,
        a=a2(1:2*n4);
    else
        a=a2(1:2*m4);
    end;
end;
m3=m4
n3=n4
clear m4,clear n4;
if n3<=m3,
    for i=1:2:2*n3,
        j=(i+1)/2;
        wt(i)=pole(j);
    end;
    for i=2:2:2*n3,
        j=i/2;
        wt(i)=zero(j);
    end;
else

```

```

    for i=1:2:2*m3,
        j=(i+1)/2;
        wt(i)=pole(j);
    end;
    for i=2:2:2*m3,
        j=i/2;
        wt(i)=zero(j);
    end;
end;
end;
%clear pole,clear zero;
if 2*se==4,
    clear A4;
    A4=A;
end;
if 2*se==6,
    clear A6;
    A6=A;
end;
if 2*se==8,
    clear A8;
    A8=A;
end;
if 2*se==10,
    clear A10;
    A10=A;
end;
if n3~=4,
    gec=su
end;
if n3<=m3,
    for i=1:2:2*n3,
        j=(i+1)/2;
        wo(i)=wcp(j);
    end;
    for i=2:2:2*n3,
        j=i/2;
        wo(i)=wcz(j);
    end;
    se=n3;
else
    for i=1:2:2*m3,
        j=(i+1)/2;
        wo(i)=wcp(j);

```

```

end;
for i=2:2:2*m3,
    j=i/2;
    wo(i)=wcz(j);
end;
se=m3;
end;
clear wcp,clear wcz;
wfark=wt-wo;
for i=1:2:2*se,
    if abs(wfark(i))<=0.01*wt(i),
        ah=2+i;
    else
        if wfark(i)<0,
            a(i)=a(i)-0.05;
        else
            a(i)=a(i)+0.05;
        end;
    end;
end;
for i=2:2:2*se,
    if abs(wfark(i))<=0.01*wt(i),
        ay=2+i;
    else
        if wfark(i)<0,
            a(i)=a(i)+0.05;
        else
            a(i)=a(i)-0.05;
        end;
    end;
end;
for i=1:2*se,
    if abs(wfark(i))<=0.01*wt(i),
        clear z;
        z=2+i;
    else
        r=2+i;
    end;
end;
clear ah,clear ay;
end;
for i=1:2*se,
    fprintf('wo(i) degeri.....:%g\n',wo(i))
end;

```

```
for r=1:2*se,  
    fprintf('wfark(r) degeri.....:%g\n',wfark(r))  
end;  
for i=1:L/h,  
    fprintf('A(i) degeri.....:%g\n',A(i))  
end;
```

EK 8

EŞİTLİK (4.38)'E DAYANAN KUTUP SIFIR MODELİNDEN
ENİNE KESİT ALANLARI BULAN PROGRAM

```

load d13;
load d14;
    l=length(DT1);
% sayac=DT1(1);
% y=DT1(2);
% peak=DT1(3);
pik=DT1(3)-DT1(1)+1;
DT=DT1;
p=DT1(2)-DT1(1)+1
clear DT1;
    for i=1:l-3,
        DT1(i)=DT(i+3);
    end;
% DT1=1.5*DT1;
% d=1000;
% for i=5:30,
%     if DT12(i)>d,
%         d=DT12(i);
%         n=i;
%     end;
% end;
% e=1000;
% for i=5:30,
%     if DT2(i)>e,
%         e=DT2(i);
%         m=i;
%     end;
% end;
%     r=n-m
% s=input('input number of s.....:');
%     l1=length(DT12);
%     for i=l1-s:l1,
%         DT1(i+s-l1+1)=DT12(i);
%     end;
%     for i=1:l1-s-1,
%         DT1(i+s+1)=DT12(i);
%     end;
sayac=1;
l=length(DT1)
N=input('INPUT NUMBER OF CHANNEL DATA...:');
% x=0;a=5;
% for i=N:-1:1,

```

```

%   for j=1:1,
%       DT11(i,j)=DT1(i+1-j);
%   end;
% end;
    for i=1:N,
%       if i==a,
% SMP1(i,1)=(DT1(i-x)-DT1(i-x-1))/2+DT1(i-x-1);
%           x=x+1;a=a+5;
%       else
%           SMP1(i,1)=DT1(i-x);
%       end;
%=====
    V1(i,1)=2.119706319e-3*DT1(i)+.078720725;
    SES1(i,1)=V1(i,1)-3.79;
% SES1(i,1)=V1(i,1)-4;
end;
    for i=2:N,
        SESP(i,1)=SES1(i,1)-SES1(i-1,1);
    end;
    SESP(1,1)=SES1(1,1);
    for m=1:N,
        w(m,1)=0.54-(0.46*cos(2*pi*m/N));
    end;
    for i=1:N,
%DIKKAT!..SESP SES1 ile degisti.
        SESW(i,1)=SESP(i,1)*w(N+1-i,1);
    end;
%for i=1:N,
% SMP(i,1)=DT11(i,1)-1855;
%end;
%   for i=1:1:N,
%       for j=1:1,
%           SMP1(i,j)=(SMP(i+1-j))/100;
%       end;
%   end;
% save orop SESW;
%=====
NN=input('NUMBER OF CHANNEL1 DATA...:');
%   j=1;
% for i=1:NN,
%     SMP2(j,1)=DT2(i);
%     j=j+5;
% end;
%   k=1;

```



```

% for x=1:NN-1,
%   for i=1:4,
%     SMP2(i+k,1)=(SMP2(k+5)-SMP2(k))*i/5+SMP2(k);
%   end;
%   k=k+5;
% end;
%=====
for i=1:NN,
  V2(i,1)=2.119706319e-3*DT2(i)+.078720725;
  SES2(i,1)=V2(i,1)-3.78;
% SE2(i,1)=SES2(i,1)*0.005;
end;
%DT21(i,1)=DT2(i,1)-1855;
  for m=1:NN,
    w(m,1)=0.54-(0.46*cos(2*pi*m/NN));
  end;
  for i=1:NN,
    for j=1:1,
      DC2(i,j)=SES2(i,j)-SES2(1,1);
    end;
  end;
for i=1:NN,
  for j=1:1,
    if SES2(i,j)<0
      SEV(i,j)=0;
    else
      SEV(i,j)=SES2(i,j);
    end;
  end;
end;
%   for i=1:NN,
%     SEV(i,1)=SEV(i,1)/6;
%   end;
%   for i=1:NN,
%     SES2(i,1)=SES2(NN-i+1,1);
%   end;
% for i=1:NN,
%   SES3(i,1)=SES2(NN-i+1,1);
% end;
  for i=1:NN,
    DC2W(i,1)=DC2(i,1)*w(N+1-i,1);
  end;
%   for i=1:NN,
%     if i<90,

```

```

% SEYW(i,1)=0;
% else
% SEYW(i,1)=SEVW(i,1);
% end;
% end;
% for i=1:NN,
% if i>110,
% SEW(i,1)=0;
% else
% SEW(i,1)=SEYW(i,1);
% end;
% end;

clc
%subplot(211),plot(SSE1);pause
%subplot(212),plot(SSE2);pause
%subplot(313),plot(SSEW);pause
%meta ses12
pr8
N=input('Ornek sayisi.....:');clc
for i=N:-1:1,
for j=1:1,
SES(i,j)=SESP(i+1-j);
end;
end;
% load ali;U=T;
% U=U;
SR=-10;MDL=100;
TN=1000;AIC=100;
% for m=1:1,
m=1;
% for k=1:N,
% if m==0,
% b=1;
% else
% b=m;
% end;
for n=6:20,
disp(' .....WORKING.....! ')
fprintf(' (m,n)=(%g,%g)\n',m,n)
for i=N:-1:1,
for j=m:-1:1,
if (i-j)<0,
H(i,j)=0;
else

```

```

        H(i,j)=SES2(i+2-j);
    end;
end;
end;
for i=N:-1:1,
    for j=n:-1:1,
        if (i-j)<0,
            L(i,j)=0;
        else
            L(i,j)=SES(i+1-j);
        end;
    end;
end;
A=[H,L];
clear L;
for i=N:-1:1,
    for j=1:1,
        SESX(i,j)=SESP(i+2-j);
    end;
end;
%clg
%subplot(211),plot(DC2);
%subplot(212),plot(H5);pause
%meta input
P=((inv(A'*A))*A')*(SESX);
% PRMTR=P';
%end;
for i=1:N,
    for j=n:-1:1,
        if (i-n-1+j)<0,
            H1(i,n-j+1)=0;
        else
            H1(i,n-j+1)=SES(i-n+j);
        end;
    end;
end;
for i=1:N,
    for j=m:-1:1,
        if (i-m-1+j)<0,
            L1(i,m-j+1)=0;
        else
            L1(i,m-j+1)=SES2(i-m+j);
        end;
    end;
end;

```

```

end;
S1=[L1 H1];
clear H1,clear L1;
W=SESX-S1*P;
for i=1:n+m,
    for j=1:n+m,
        if i==j,
            PO(i,j)=100;
        else
            PO(i,j)=0;
        end;
    end;
end;
K=PO*(S1(1,1:n+m)')*(inv(S1(1,1:n+m)*PO*(S1(1,1:n+m)')+1));
for i=1:N,
    if i==1;W(i)=SESX(i);r(i)=(W(i)^2)/i;Q1=K*SESX(i+1);
        P1=PO-(PO*(S1(i,1:n+m)')*(inv(S1(i,1:n+m)*PO*(S1(i,1:n+m)')+1))
            *S1(i,1:n+m)*PO);
        K1=P1*(S1(i+1,1:n+m)')*(inv(S1(i+1,1:n+m)*P1*(S1(i+1,1:n+m)')+1));
    else W(i)=SESX(i)-S1(i-1,1:n+m)*Q1;
        r(i)=(((i-1)/i)*r(i-1))+((W(i)^2)/i);
        if i==N,break;
        end;
        Q2=Q1+(K1*(SESX(i+1)-S1(i,1:n+m)*Q1));
        Q1=Q2;
        P2=P1-(P1*(S1(i,1:n+m)')*(inv(S1(i,1:n+m)*P1*(S1(i,1:n+m)')+1))
            *S1(i,1:n+m)*P1);
        P1=P2;
        K2=P2*(S1(i+1,1:n+m)')*(inv(S1(i+1,1:n+m)*P2*(S1(i+1,1:n+m)')+1));
        K1=K2;
    end;
end;
for i=1:N,
    if i==1;b(i)=(SESX(i)^2)/i;
    else b(i)=(((i-1)/i)*b(i-1))+((SESX(i)^2)/i);
    end;
end;
V=SESX-A*P;
V2=V'*V;
t=((V1-V2)/V2)*(N-(n+m));
t1=abs(t);
fprintf('t1 degeri...:%g\n',t1)
V1=V2;
fprintf('SSE degeri...:%g\n',V1)

```

```

    if t1<3,break;
    end;
    if V1<TN,
        m3=m;n3=n;
        TN=V1;PRMTR=P';PRMTR1=Q1';
    end;
    SRR=10*log10(b(N)/r(N));
    if SRR>SR,
        m1=m;n1=n;
        SR=SRR;
    end;
    fprintf('SRR degeri...:%g\n',SRR)
    AC=(N*log(r(N)))+(n+m)*2;
    if AC<AIC,
        m2=m;n2=n;
        AIC=AC;
    end;
    fprintf('AIC degeri...:%g\n',AC)
    ML=(N*log(r(N)))+(n+m)*log(N);
    if ML<MDL,
        m4=m;n4=n;
        MDL=ML;
    end;
    fprintf('MDL degeri...:%g\n',ML)
clear P0,clear P1,clear K,clear K1,clear Q1,clear P,clear S1,clear W,clear r;
clear Q2,clear P2,clear K2,clear b,clear V,clear V2,clear t,clear t1;
end;
    PYD=-PRMTR(:,m3+1:m3+n3);
    PAYDA=[1 .PYD];
clear PRMTR,clear PYD;
% PYD1=-PRMTR1(:,m3+1:m3+n3);
% PAYDA1=[1 PYD1];
    B=[zeros(1:n3+1)];
    B=PAYDA;
% B=[1,1.0714,0.2995,-0.3305,-0.7195,-0.6261,-0.2222];
% B=PAYDA1;
    M=n3;
    for j=1:M,
        MR=M+1-j;
        MRP1=MR+1;
        D=1-B(MRP1)*B(MRP1);
        RC(MR)=B(MRP1);
        for k=1:MR,
            MM=MR+2-k;

```

```

        C(k)=B(MM);
    end;
        for k=1:MR,
            B(k)=(B(k)-B(MRP1)*C(k))/D;
        end;
    end;
clear B,clear MR,clear MRP1,clear D,clear MM,clear C;
REC=-RC;
LPC(1)=1;
LPC(2)=REC(1);
    for MINC=2:M,
        for J=1:MINC,
            JB=MINC-J+1;
            B(J)=LPC(JB);
        end;
        for IP=2:MINC,
            LPC(IP)=LPC(IP)+REC(MINC)*B(IP-1);
        end;
        LPC(MINC+1)=REC(MINC);
    end;
clear RC,clear REC,clear JB,clear B,clear M,clear MINC,clear IP;
zero=roots(LPC)
pole=roots(PAYDA)
clear LPC,clear PAYDA;
r=0;
for i=1:n3,
    if imag(zero(i))~=0,
        zer(i-r)=zero(i);
        y=i-r;
    else
        r=r+1;
    end;
end;
x=1;
u=0;
for i=1:y-1,
    if imag(zer(i))== -imag(zer(i+1)),
        o(x)=i;
        u=1;
        x=x+1;
    else
        o(x)=i+u;
        x=x+1;
    end;
end;

```

```

end;
clear r,clear i,clear y,clear u,clear zero;
u=0;
for i=1:x-2,
    if o(i)==o(i+1),
        u=u+1;
    else
        ze(i-u)=abs(imag(zer(o(i))));
    end;
end;
ze(x-1-u)=abs(imag(zer(o(x-1))));
clear o,clear i,clear zer;
s=x-1-u;
t=0;
for i=1:s,
    zero(i)=min(ze);
    if i==s, break;
end;
for j=1:s-t,
    l=s-t;
    if zero(i)==ze(j),
        k=j;
        if k==1,
            zef=ze(k+1:s-t);
        else
            if k==l,
                zef=ze(1:k-1);
            else
                zel=ze(1:k-1);
                ze2=ze(k+1:s-t);
                zef=[zel,ze2];
            end;
        end;
    end;
end;
end;
t=i;
ze=zef;
end;
m3=s;
clear u,clear x,clear ze,clear s,clear t,clear i,clear j,clear l,clear k;
clear zef,clear zel,clear ze2;
zero
r=0;
for i=1:n3,

```

```

if imag(pole(i))~=0,
    pol(i-r)=pole(i);
    y=i-r;
else
    r=r+1;
end;
end;
x=1;
u=0;
for i=1:y-1,
    if imag(pol(i))== -imag(pol(i+1)),
        o(x)=i;
        u=1;
        x=x+1;
    else
        o(x)=i+u;
        x=x+1;
    end;
end;
clear r,clear i,clear y,clear u,clear pole;
u=0;
for i=1:x-2,
    if o(i)==o(i+1),
        u=u+1;
    else
        po(i-u)=abs(imag(pol(o(i)))));
    end;
end;
po(x-1-u)=abs(imag(pol(o(x-1)))));
clear o,clear i,clear pol;
s=x-1-u;
t=0;
for i=1:s,
    pole(i)=min(po);
    if i==s, break;
end;
for j=1:s-t,
    l=s-t;
    if pole(i)==po(j),
        k=j;
        if k==1,
            pof=po(k+1:s-t);
        else
            if k==l,

```



```

j=sqrt(-1);
for i=1:L/h,
    b=0;
    for y=1:2*se,
        b=b+(a(y)*cos((y*pi*(i-0.5)*h)/L));
    end;
    A(i)=exp(b);
end;
if z==2*se,
    break;
end;
clear wo,clear wfark;
for w=w1:w1:w2,
    NS=[1,0;0,1];
    for i=1:L/h,
        R(i)=cos(w*h/c);
        B(i)=((j*p*c)/A(i))*(sin(w*h/c));
        C(i)=((j*A(i))/(p*c))*(sin(w*h/c));
        D(i)=cos(w*h/c);
        M=[R(i),B(i);C(i),D(i)];
        NS=NS*M;
    end;
    VEC=NS*[1;0];
    f=w/(2*pi*25);
    PT(f)=VEC(1);
    ZT(f)=VEC(2);
end;
%clg
%subplot(211),plot(PT);pause
%subplot(212),plot(ZT);pause
%meta girdukap
clear R,clear B,clear C,clear D,clear M,clear NS,clear VEC,clear b;
n=1;
for i=1:f-1,
    car=PT(i+1)*PT(i);
    if car<=0,
        w4p(n)=(i+1)*2*pi*25;
        w3p(n)=2*i*pi*25;
        g(n)=i;
        n=n+1;
    end;
end;
clear car;
m=1;

```

```

for i=1:f-1,
    car=ZT(i+1)*ZT(i);
    if car>=0,
        w4z(m)=(i+1)*2*pi*25;
        w3z(m)=2*i*pi*25;
        l(m)=i;
        m=m+1;
    end;
end;
clear car;
for i=1:n-1,
    wcp(i)=w3p(i)-(PT(g(i))*((w4p(i)-w3p(i))/(PT(g(i)+1)-PT(g(i)))));
end;
for r=1:m-1,
    wcz(r)=w3z(r)-(ZT(l(r))*((w4z(r)-w3z(r))/(ZT(l(r)+1)-ZT(l(r)))));
end;
if m<n,
    wcz(n-1)=wcp(n-1)+((wcp(n-1)-wcp(n-2))/2);
    if m<n-1,
        wcz(n-2)=wcp(n-2)+((wcp(n-1)-wcp(n-2))/2);
        if m<n-2,
            wcz(n-3)=wcp(n-3)+((wcp(n-2)-wcp(n-3))/2);
        end;
    end;
end;
clear PT,clear ZT,clear g,clear l,clear w3p,clear w4p,clear w3z,clear w4z,
clear f,clear r;
for i=1:n-1,
    wcp(i)=j*wcp(i);
    wcz(i)=j*wcz(i);
end;
T=100E-6;
for i=1:n-1,
    v(i)=sqrt(exp(2*real(wcp(i))*T)/(1+(tan(imag(wcp(i))*T))^2));
    d(i)=v(i)*tan(imag(wcp(i))*T);
    wcp1(i)=v(i)+j*d(i);
    wcp2(i)=abs(imag(wcp1(i)));
end;
for i=1:n-1,
    e(i)=sqrt(exp(2*real(wcz(i))*T)/(1+(tan(imag(wcz(i))*T))^2));
    f(i)=e(i)*tan(imag(wcz(i))*T);
    wcz1(i)=e(i)+j*f(i);
    wcz2(i)=abs(imag(wcz1(i)));
end;

```

```

clear v,clear d,clear e,clear f,clear wcp,clear wcz,clear T;
s=n-1;
t=0;
for i=1:s,
    wcp(i)=min(wcp2);
    if i==s, break;
end;
for j=1:s-t,
    l=s-t;
    if wcp(i)==wcp2(j),
        k=j;
        if k==1,
            wcpf=wcp2(k+1:s-t);
        else
            if k==1,
                wcpf=wcp2(1:k-1);
            else
                wc1p=wcp2(1:k-1);
                wc2p=wcp2(k+1:s-t);
                wcpf=[wc1p,wc2p];
            end;
        end;
    end;
end;
end;
t=i;
wcp2=wcpf;
end;
clear wc1,clear wcp2,clear s,clear t,clear i,clear j,clear l,clear k.
clear wcpf;
clear wc1p,clear wc2p;
s=n-1;
t=0;
for i=1:s,
    wcz(i)=min(wcz2);
    if i==s, break;
end;
for j=1:s-t,
    l=s-t;
    if wcz(i)==wcz2(j),
        k=j;
        if k==1,
            wczf=wcz2(k+1:s-t);
        else
            if k==1,

```



```

clear a1;
a1=a;
end;
for i=N:-1:1,
    for j=n:-1:1,
        if (i-j)<0,
            L1(i,j)=0;
        else
            L1(i,j)=SES(i+1-j);
        end;
    end;
end;
A1=[H,L1];
P=((inv(A1'*A1))*A1')*(SESX);
PRMTR=P';
clear L1,clear A1,clear P,;
PYD=-PRMTR(:,q+1:q+n);
PAYDA=[1 PYD];
B=[zeros(1:n+1)];
B=PAYDA;
clear PRMTR, clear PYD;
M=n;
for j=1:M,
    MR=M+1-j;
    MRP1=MR+1;
    D=1-B(MRP1)*B(MRP1);
    RC(MR)=B(MRP1);
    for k=1:MR,
        MM=MR+2-k;
        C(k)=B(MM);
    end;
    for k=1:MR,
        B(k)=(B(k)-B(MRP1)*C(k))/D;
    end;
end;
clear MR,clear MRP1,clear D,clear MM,clear C,clear B;
REC=-RC;
LPC(1)=1;
LPC(2)=REC(1);
for MINC=2:M,
    for J=1:MINC,
        JB=MINC-J+1;
        B(J)=LPC(JB);
    end;
end;

```

```

    for IP=2:MINC,
        LPC(IP)=LPC(IP)+REC(MINC)*B(IP-1);
    end;
    LPC(MINC+1)=REC(MINC);
end;
clear RC,clear REC,clear JB,clear B,clear M,clear MINC,clear IP;
zero=roots(LPC)
pole=roots(PAYDA)
clear PAYDA,clear LPC;
if all(imag(zero)==0),
    for i=1:n,
        zero(i)=0;
    end;
    s=n/2;
else
    r=0;
    for i=1:n,
        if imag(zero(i))~=0,
            zer(i-r)=zero(i);
            y=i-r;
        else
            r=r+1;
        end;
    end;
end;
x=1;
u=0;
for i=1:y-1,
    if imag(zer(i))==-imag(zer(i+1)),
        o(x)=i;
        u=1;
        x=x+1;
    else
        o(x)=i+u;
        x=x+1;
    end;
end;
clear r,clear u,clear y,clear zero,clear i;
u=0;
for i=1:x-2,
    if o(i)==o(i+1),
        u=u+1;
    else
        ze(i-u)=abs(imag(zer(o(i)))));
    end;
end;

```

```

end;
ze(x-1-u)=abs(imag(zer(o(x-1)))));
clear o,clear i,clear zer;
s=x-1-u;
t=0;
for i=1:s,
    zero(i)=min(ze);
    if i==s, break;
end;
for j=1:s-t,
    l=s-t;
    if zero(i)==ze(j),
        k=j;
        if k==1,
            zef=ze(k+1:s-t);
        else
            if k==1,
                zef=ze(1:k-1);
            else
                ze1=ze(1:k-1);
                ze2=ze(k+1:s-t);
                zef=[ze1,ze2];
            end;
        end;
    end;
end;
end;
end;
t=i;
ze=zef;
end;
end;
m4=s;
zero
clear u,clear x,clear ze,clear s,clear t,clear j,clear l,clear k;
clear zef,clear ze1,clear ze2;
if all(imag(pole)==0),
    for i=1:n,
        pole(i)=0;
    end;
    s=n/2;
else
    r=0;
    for i=1:n,
        if imag(pole(i))~=0,
            pol(i-r)=pole(i);

```



```

    y=i-r;
    else
        r=r+1;
    end;
end;
end;
x=1;
u=0;
for i=1:y-1,
    if imag(pol(i))==-imag(pol(i+1)),
        o(x)=i;
        u=1;
        x=x+1;
    else
        o(x)=i+u;
        x=x+1;
    end;
end;
clear r,clear y,clear u,clear pole;
u=0;
for i=1:x-2;
    if o(i)==o(i+1),
        u=u+1;
    else
        po(i-u)=abs(imag(pol(o(i)))));
    end;
end;
po(x-1-u)=abs(imag(pol(o(x-1)))));
clear o,clear i,clear pol;
s=x-1-u;
t=0;
for i=1:s,
    pole(i)=min(po);
    if i==s, break;
end;
for j=1:s-t,
    l=s-t;
    if pole(i)==po(j);
        k=j;
        if k==1,
            pof=po(k+1:s-t);
        else
            if k==l,
                pof=po(1:k-1);
            else

```

```

        po1=po(1:k-1);
        po2=po(k+1:s-t);
        pof=[po1,po2];
    end;
end;
end;
end;
t=i;
po=pof;
end;
end;
n4=s;
clear u,clear x,clear s,clear po,clear t,clear j,clear l,clear k,clear i;
clear po1,clear po2,clear a;
pole
    if (n4+m4)>(2*se),
        for i=1:2*se,
            a(i)=a2(i);
        end;
        for i=2*n3+1:of,
            a(i)=a1(i);
        end;
        if n4<=m4,
            for i=of+1:2:2*n4,
                a(i)=0.5;
            end;
            for i=of+2:2:2*n4,
                a(i)=-0.5;
            end;
        else
            for i=of+1:2:2*m4,
                a(i)=0.5;
            end;
            for i=of+2:2:2*m4,
                a(i)=-0.5;
            end;
        end;
    end;
else
    if n4<=m4,
        a=a2(1:2*n4);
    else
        a=a2(1:2*m4);
    end;
end;
end;

```

```

clear i;
n3=n4
m3=m4
clear m4,clear n4;
  if n3<=m3,
    for i=1:2:2*n3,
      j=(i+1)/2;
      wt(i)=pole(j);
    end;
    for i=2:2:2*n3,
      j=i/2;
      wt(i)=zero(j);
    end;
  else
    for i=1:2:2*m3,
      j=(i+1)/2;
      wt(i)=pole(j);
    end;
    for i=2:2:2*m3,
      j=i/2;
      wt(i)=zero(j);
    end;
  end;
end;
if 2*se==4,
  clear A4;
  A4=A;
end;
if 2*se==6,
  clear A6;
  A6=A;
end;
if 2*se==8,
  clear A8;
  A8=A;
end;
if 2*se==10,
  clear A10;
  A10=A;
end;
if (2*se)~=2*n,
  gec=su
end;
% wt=[2*pi*420,2*pi*1077,2*pi*1900,2*pi*2220,2*pi*2950,2*pi*3260];

```

```

if n3<=m3,
  for i=1:2:2*n3,
    j=(i+1)/2;
    wo(i)=wcp(j);
  end;
  for i=2:2:2*n3,
    j=i/2;
    wo(i)=wcz(j);
  end;
  se=n3;
else
  for i=1:2:2*m3,
    j=(i+1)/2;
    wo(i)=wcp(j);
  end;
  for i=2:2:2*m3,
    j=i/2;
    wo(i)=wcz(j);
  end;
  se=m3;
end;
clear wcp,clear wcz;
wfark=wt-wo;
for i=1:2:2*se,
  if abs(wfark(i))<=0.01*wt(i),
    ah=2+i;
  else
    if wfark(i)<0,
      a(i)=a(i)+0.05;
    else
      a(i)=a(i)-0.05;
    end;
  end;
end;
for i=2:2:2*se,
  if abs(wfark(i))<=0.01*wt(i),
    ay=2+i;
  else
    if wfark(i)<0,
      a(i)=a(i)-0.05;
    else
      a(i)=a(i)+0.05;
    end;
  end;
end;

```

```
    end;
z=0;
for i=1:2*se,
    if abs(wfark(i))<=0.01*wt(i),
        z=z+1;
    else
        r=2+i;
    end;
end;
clear r;
clear ah,clear ay;
end;
for i=1:2*se,
    fprintf('wo(i) degeri.....:%g\n',wo(i))
end;
for r=1:2*se,
    fprintf('wfark(r) degeri.....:%g\n',wfark(r))
end;
for i=1:L/h,
    fprintf('A(i) degeri.....:%g\n',A(i))
end;
```

ENINE KESİT ALANLARI İNSAN KAFASINA YERLEŞTİREN PROGRAM

```

#include <bios.h>
#include <conio.h>
#include <stdio.h>
#include <graphics.h>
#include <math.h>
#include <time.h>
#include <dos.h>
clock_t start,end;
char   kutuk_ismi[12];
char   filename[12];
char   kutukadi[12];
char   temp[10];
char   anychar;
double an,bn;
float  w[100],w1[100],f[100],f1[100];
float  alan,a0,uzunluk,tempuzunluk,n_value,x_value,aci,oran,tempvalue,p,x,n,t;
int    GraphDriver,GraphMode;
int    dataloc,curveloc,filenameeloc,say,location;
FILE   *fp, *kg;
int    x_point,y_point,y_pointp,count;
float  delta_x,delta_y;
int    x2,y2,fxx,fx,sx;
hesap_yap()
{
    a0=a0*2/alan;
    gotoxy(40,8);printf("A0=%f ",a0);
    start=clock();
    gotoxy(40,1);printf("HESAPLAMA SURESI:");
    for(x=1;x<=x_value;x++)
    {
        p=0;
        gotoxy(40,3);printf("[ %g.X",x);
        gotoxy(47,3);printf("] Deger Icin Hesap");
        for(n=1;n<=n_value;n++)
        {
            an=0;
            bn=0;
            gotoxy(44,4);printf("  ");
            gotoxy(40,4);printf("[ n=%g",n);
            gotoxy(47,4);printf("] Icin An=Bn Degeri");
            for(t=1;t<=alan;t++)
            {
                an=(f[t]*(sin(2*n*3.14159*t/alan)-sin(2*n*3.14159*(t-1)/alan)))+an;
                bn=(f[t]*(cos(2*n*3.14159*t/alan)-cos(2*n*3.14159*(t-1)/alan)))+bn;
            }
            an=an/(n*3.14159);
            bn=-bn/(n*3.14159);
            gotoxy(41,5);printf("  ");
            gotoxy(41,6);printf("  ");
            gotoxy(40,5);printf("A%g",n);
            gotoxy(40,6);printf("B%g",n);
            gotoxy(43,5);printf("=[");
            gotoxy(43,6);printf("=[");
            gotoxy(45,5);printf("  ");
            gotoxy(45,6);printf("  ");
        }
    }
}

```

```

        gotoxy(45,5);printf("%g",an);
        gotoxy(45,6);printf("%g",bn);
        gotoxy(57,5);printf("J");
        gotoxy(57,6);printf("J");
        p=(an*cos(2*n*3.14159*x/uzunluk)+bn*sin(2*n*3.14159*x/uzunluk))+p;
    }
    w[x]=(a0/2)+p;
    fprintf(fp,"%g\n",w[x]);
    gotoxy(40,7);printf("W[%g]=[%g]",x,w[x]);
}
end=clock();
gotoxy(40,1);printf("HESAPLAMA SURESI:%f ",(end - start)/CLK_TCK);
gotoxy(50,22);printf("1.ENTER- Karsilastirma icin..");
gotoxy(50,23);printf("2.ENTER- Kafa cizimi icin....");
sound(257);delay(999);nosound();
getch();
}
data_yaz_f(dataloc)
{
    initgraph(&GraphDriver,&GraphMode," ");
    setlinestyle(0,0,1);
    for(say=1;say<alan+1;say++)
    {
        outtextxy(455,dataloc+8*say,".ALAN=");
        sprintf(temp,"%g",f[say]);
        outtextxy(506,dataloc+8*say,temp);
        sprintf(temp,"%d",say);
        outtextxy(442,dataloc+8*say,temp);
    }
}
data_yaz_w(dataloc)
{
    for(say=1;say<=uzunluk;say++)
    {
        outtextxy(455,dataloc+8*say,".ALAN");
        sprintf(temp,"%g",w[say]);
        outtextxy(506,dataloc+8*say,temp);
        sprintf(temp,"%d",say);
        outtextxy(442,dataloc+8*say,temp);
    }
}
location_of_filename(curveloc)
{
    if (curveloc==50) filenameloc=80;
    if (curveloc==170) filenameloc=200;
    if (curveloc<270)
    {
        outtextxy(100,filenameloc,kutukadi);
    }
}
egri_ciz(curveloc)
{
    oran=uzunluk/alan;
    moveto(20,curveloc-15*w[1]);
    location=0;
}

```

```

for (say=1;say<=uzunluk;say++)
{
    lineto(20+location,curveloc-15*w[say]);
    location=location+39/oran;
}
moveto(20,curveloc+15*w[1]);
location=0;
for (say=1;say<=uzunluk;say++)
{
    lineto(20+location,curveloc+15*w[say]);
    location=location+39/oran;
}
location_of_filename(curveloc);
}
kesikli_egri_ciz(curveloc)
{
    oran=uzunluk/alan;
    moveto(20,curveloc-15*f[1]);
    location=0;
    for (say=1;say<=alan;say++)
    {
        lineto(20+location,(curveloc-15*f[say]));
        location=location+35/oran;
        lineto(20+location,(curveloc-15*f[say]));
    }
    moveto(20,(curveloc+15*f[1]));
    location=0;
    for (say=1;say<=alan;say++)
    {
        lineto(20+location,(curveloc+15*f[say]));
        location=location+35/oran;
        lineto(20+location,(curveloc+15*f[say]));
    }
    location_of_filename(curveloc);
}
karsilastir1()
{
    kesikli_egri_ciz(270);
    egri_ciz(270);
    outtextxy(20,320,"Devam icin herhangi bir tusa basiniz");
    getch();
}
karsilastir2()
{
    egri_ciz(270);
    say=1;
    do
    {
        w[say]=w1[say];
        say++;
    }
    while (say<=uzunluk);
    egri_ciz(270);
    outtextxy(20,320,"Devam icin herhangi bir tusa basiniz");
    getch();
}

```



```

}
kafa_ciz()
{
    initgraph(&GraphDriver,&GraphMode," ");
    setlinestyle(0,0,1);
    circle(232,106,3);           /* d1 */
    circle(232,106,5);           /* d2 */
    circle(232,106,7);           /* d3 */
    ellipse(360,32,5,180,253,30); /* e1 */
    ellipse(260,32,180,300,153,25); /* e2 */
    ellipse(610,180,350,85,30,150); /* e3 */
    arc(144,179,180,270,10);     /* r1 */
    arc(187,295,135,255,28);     /* r2 */
    arc(225,108,45,135,51);      /* r3 */
    arc(306,146,10,78,124);      /* r4 */
    arc(252,108,147,213,29);     /* r5 */
    arc(99,108,0,25,63);         /* r6 */
    arc(174,216,180,360,9);      /* r7 */
    arc(176,241,30,200,9);       /* r8 */
    line(156,88,157,50);         /* l1 */
    line(428,130,438,150);       /* l2 */
    line(438,150,470,130);       /* l3 */
    line(470,130,485,130);       /* l4 */
    line(485,130,520,150);       /* l5 */
    line(520,150,520,160);       /* l6 */
    line(520,160,495,220);       /* l7 */
    line(495,220,480,221);       /* l8 */
    line(480,221,477,216);       /* l9 */
    line(640,208,625,240);       /* l10 */
    line(625,240,575,290);       /* l11 */
    line(575,290,525,320);       /* l12 */
    line(495,220,485,237);       /* l13 */
    line(485,237,525,320);       /* l14 */
    line(575,290,590,320);       /* l15 */
    line(305,315,310,330);       /* l16 */
    line(252,108,216,90);        /* l17 */
    line(252,108,216,126);       /* l18 */
    line(162,108,135,179);       /* l19 */
    line(144,185,171,190);       /* l20 */
    line(174,267,166,280);       /* l21 */
    line(180,315,305,315);       /* l22 */
    line(171,190,165,215);       /* l23 */
    line(167,243,174,267);       /* l24 */
    line(184,216,186,206);       /* l25 */
    line(186,206,189,223);       /* l26 */
    line(184,237,186,249);       /* l27 */
    line(186,249,189,235);       /* l28 */
    line(190,220,300,185);       /* u1 ***** */
    line(300,185,340,185);       /* u2 ** Ust Damak ** */
    line(340,185,420,215);       /* u3 ** Cizimi ** */
    line(420,215,500,345);       /* u4 ***** */
    moveto(189,235);
    fx=298;
    fxx=298;
    for(count=1;count<=3;count++)

```

```

{
  if (fx-(190+36*(4-count))==0)
  {
    sx=190+(fxx-(190+36*(3-count)));
  }
  x2=sx-30*w[count]*sin(atan(-0.318));
  y2=(-0.318*sx+280.45)+30*w[count]*cos(atan(-0.318));
  lineto(x2,y2);
  fx=fx-36;
}
x2=340;
y2=185+30*w[count];
lineto(x2,y2);
fx=420;
fxx=420;
say=count+1;
do
{
  if (fx-(340+40*(7-say))==0)
  {
    sx=340+(fxx-(340+40*(6-say)));
  }
  x2=sx-30*w[say]*sin(atan(0.375));
  y2=(0.375*sx+57.5)+30*w[say]*cos(atan(0.375));
  lineto(x2,y2);
  fx=fx-40;
  say++;
}
while(say<=count+2);
fx=500;
fxx=500;
count=say;
do
{
  if (fx-(420+20*(11-count))==0)
  {
    sx=420+(fxx-(420+20*(10-count)));
  }
  x2=sx-30*w[count]*sin(atan(1.625));
  y2=(1.625*sx-467.5)+30*w[count]*cos(atan(1.625));
  lineto(x2,y2);
  fx=fx-20;
  count++;
}
while(count<=10);
outtextxy(20,320,"Devam icin herhangi bir tusa basiniz");
getch();
}
menu()
{
  clearviewport();
  settextstyle(6,0,10);
  rectangle(0,0,718,345);
  rectangle(15,15,703,330);
  floodfill(10,10,1);
}

```

```

    outtextxy(100,50,"Kesikli Egri Karsilastirma Hesabi.....:<H>");
    outtextxy(100,100,"Iki Egri Karsilastirmasi.....:<E>");
    outtextxy(100,150,"Dosyadan Egri Karsilastir.....:<D>");
    outtextxy(100,200,"C I K I S .....:<X>");
    settextstyle(GOTHIC_FONT,0,2);
    outtextxy(100,250,"Bu Program Ses Bolgesinin Simulasyonu Icin
    Hazirlanmistir.");
}
first()
{
    initgraph( &GraphDriver, &GraphMode, "" );
    settextstyle(8,0,8);
    outtextxy(145,100,"Bu Program Ses Bolgesinin Simulasyonu Icin
    Hazirlanmistir.");
    settextstyle(16,0,8);
    outtextxy(245,300,"Baslamak icin herhangi bir tusa basiniz!");
    rectangle(0,0,717,345);
    rectangle(15,15,702,330);
    floodfill(10,10,1);
    setviewport(0,0,getmaxx()-1,getmaxy()-1, 1);
    sound(389);delay(1000);nosound();
    getch();
}
main()
{
    GraphDriver = DETECT;
    first();
    menu();
    while((anychar=(toupper(getch())))!='X')
    {
        setlinestyle(0,0,1);
        if(anychar=='H')
        {
            closegraph();
            gotoxy(2,1);printf("Yeni Dosya Ismini Girin=");
            scanf("%s",&kutuk_ismi);
            strcat(&kutuk_ismi, ".MAS");
            if ((fp=fopen(kutuk_ismi, "wt"))==NULL)
            {
                gotoxy(2,2);printf("Boyle bir Dosya Yok \n Devam icin herhangi
                bir tusa basin");
                getch();
                exit(1);
                menu();
            }
            gotoxy(2,2);printf("Dosya Ismini Girin=");
            scanf("%s",&filename);
            strcat(&filename, ".DAT");
            if ((kg=fopen(filename, "rt"))==NULL)
            {
                printf("Dosya Yok!");
                gotoxy(2,3);printf("Boyle bir Dosya Yok \n Devam icin
                herhangi bir tusa basin");
                getch();
                exit(1);
            }
        }
    }
}

```

```

    }
    fscanf(kg, "%g\n", &alan);
    gotoxy(2,3);printf("Alan Sayisi =%g", alan);
    gotoxy(2,4);printf("Uzunluk Ne Kadar Olacak Giriniz=");
    gotoxy(34,4);scanf("%g",&uzunluk);
    a0=0;
    gotoxy(2,5);printf("Alt Sinir=0 Ust Sinir=%g Adim=%g",
    uzunluk,uzunluk/alan);
    gotoxy(2,6);printf("N- Giriniz=");
    gotoxy(13,6);scanf("%g",&n_value);
    gotoxy(2,7);printf("X- Giriniz=");
    gotoxy(13,7);scanf("%g",&x_value);
    fprintf(fp, "%g\n", x_value);
    gotoxy(2,8);printf("Alanlar:");
    for(say=1;say<alan+1;say++)
    {
        gotoxy(2,8+say);printf("%d.ALAN...=", say);
        fscanf(kg, "%g\n", &f[say]);
        f[say]=sqrt(f[say]/3.14159);
        gotoxy(14,8+say);printf("%g", f[say]);
        a0=f[say]+a0;
    }
    fclose(kg);
    hesap_yap();
    fprintf(fp, "%g\n", uzunluk);
    fprintf(fp, "%g\n", alan);
    fprintf(fp, "%g\n", n_value);
    say=1;
    do
    {
        fprintf(fp, "%g\n", f[say]);
        say++;
    }
    while (say<=alan);
    fclose(fp);
    data_yaz_f(2);
    dataloc=2+alan*8+10;
    data_yaz_w(dataloc);
    strcpy(kutukadi, filename);
    kesikli_egri_ciz(50);
    strcpy(kutukadi, kutuk_ismi);
    egri_ciz(170);
    karsilastir1();
    kafa_ciz();
    menu();
}
if(anychar=='E')
{
    cleardevice();
    closegraph();
    gotoxy(2,1);printf("1.Dosya ismini giriniz=" );
    scanf("%s",&kutuk_ismi);
    strcat(&kutuk_ismi, ".MAS");
    if ((fp=fopen(kutuk_ismi, "rt"))==NULL)
    {

```

```

        printf("HATA: Dosya Bulunamadi");
        return;
    }
    fscanf(fp, "%g\n", &uzunluk);
    say=1;
    do
    {
        fscanf(fp, "%g", &w[say]);
        gotoxy(2, 3+say);
        printf("%d. alan=%g", say, w[say]);
        say++;
    }
    while (say<(uzunluk+1));
    fclose(fp);
    gotoxy(40, 1); printf("2. Dosya ismini giriniz=" );
    scanf("%s", &filename);
    strcat(&filename, ".MAS");
    if ((kg=fopen(filename, "rt"))==NULL)
    {
        printf("HATA: Dosya Bulunamadi");
        return;
    }
    fscanf(kg, "%g\n", &tempuzunluk);
    say=1;
    do
    {
        fscanf(kg, "%g\n", &w1[say]);
        gotoxy(40, 3+say);
        printf("%d. alan=%g", say, w1[say]);
        say++;
    }
    while (say<tempuzunluk+1);
    fclose(kg);
    gotoxy(20, 23); printf("Devam icin herhangi bir tusa basiniz");
    getch();
    initgraph(&GraphDriver, &GraphMode, " ");
    data_yaz_w(2);
    strcpy(kutukadi, kutuk_ismi);
    egri_ciz(50);
    say=1;
    do
    {
        tempvalue=w[say];
        w[say]=w1[say];
        w1[say]=tempvalue;
        say++;
    }
    while (say<=tempuzunluk);
    tempuzunluk=uzunluk;
    dataloc=12+tempuzunluk*8;
    data_yaz_w(dataloc);
    strcpy(kutukadi, filename);
    egri_ciz(170);
    karsilastir2();
    menu();

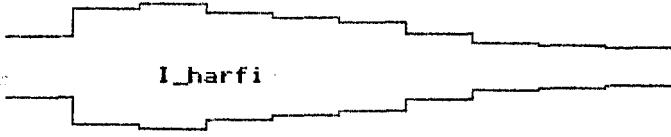
```

```

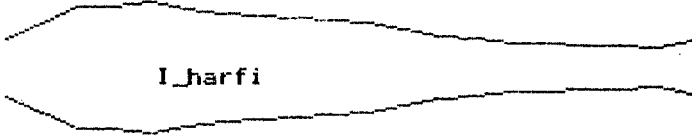
    }
if(anyschar=='D')
{
    cleardevice();
    closegraph();
    gotoxy(2,1);printf("Dosya ismini giriniz=" );
    scanf("%s",&kutuk_ismi);
    strcat(&kutuk_ismi, ".MAS");
    if ((fp=fopen(kutuk_ismi, "rt"))==NULL)
    {
        printf("HATA: Dosya Bulunamadi");
        return;
    }
    fscanf(fp, "%g\n", &x_value);
    say=1;
    do
    {
        fscanf(fp, "%g\n", &w[say]);
        say++;
    }
    while(say<=x_value);
    fscanf(fp, "%g\n", &uzunluk);
    fscanf(fp, "%g\n", &alan);
    fscanf(fp, "%g\n", &n_value);
    say=1;
    do
    {
        fscanf(fp, "%g\n", &f[say]);
        say++;
    }
    while(say<=alan);
    strcpy(kutukadi, kutuk_ismi);
    data_yaz_f(2);
    dataloc=12+8*alan;
    data_yaz_w(dataloc);
    kesikli_egri_ciz(50);
    egri_ciz(170);
    karsilastir1();
    menu();
}
}
}

```

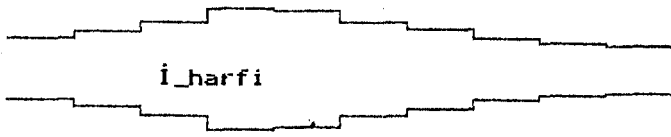
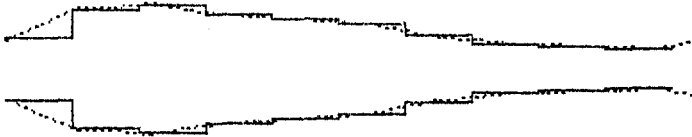
20 KİŞİDEN ALINAN 8 UNLU İÇİN ELDE EDİLEN
ENİNE KESİT ALANLARININ ORTALAMASI



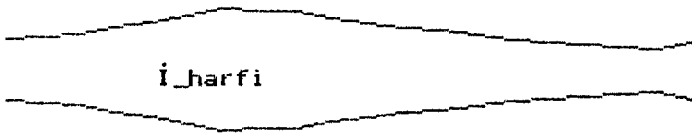
1 .ALAN= 0.874039
2 .ALAN= 1.72284
3 .ALAN= 1.85669
4 .ALAN= 1.57266
5 .ALAN= 1.44393
6 .ALAN= 1.27037
7 .ALAN= 0.93399
8 .ALAN= 0.722515
9 .ALAN= 0.630783
10 .ALAN= 0.594411



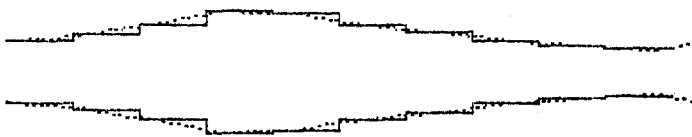
1 . . .td. Sapma=1.43
2 . . .td. Sapma=5.1
3 . . .td. Sapma=6.18
4 . . .td. Sapma=3.7
5 . . .td. Sapma=3.57
6 . . .td. Sapma=3.78
7 . . .td. Sapma=1.199
8 . . .td. Sapma=0.568
9 . . .td. Sapma=0.258
10 . . .td. Sapma=0.25

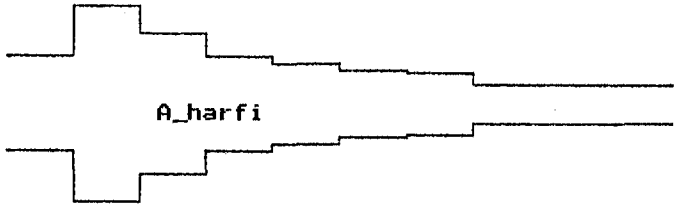


1 .ALAN= 0.914962
2 .ALAN= 1.068992
3 .ALAN= 1.36459
4 .ALAN= 1.74817
5 .ALAN= 1.69398
6 .ALAN= 1.34698
7 .ALAN= 1.13499
8 .ALAN= 0.921203
9 .ALAN= 0.764263
10 .ALAN= 0.677263



1 . . .td. Sapma=2.5455
2 . . .td. Sapma=2.54
3 . . .td. Sapma=4.225
4 . . .td. Sapma=5.068
5 . . .td. Sapma=4.021
6 . . .td. Sapma=3.581
7 . . .td. Sapma=1.672
8 . . .td. Sapma=0.941
9 . . .td. Sapma=0.412
10 . . .td. Sapma=0.34

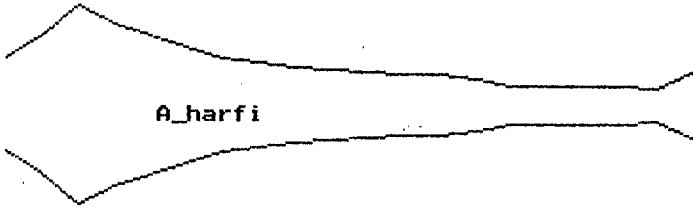




```

1 .ALAN=1.35564
2 .ALAN=2.83894
3 .ALAN=2.04202
4 .ALAN=1.38082
5 .ALAN=1.13961
6 .ALAN=0.977205
7 .ALAN=0.870389
8 .ALAN=0.89974
9 .ALAN=0.861362
10 .ALAN=0.535237

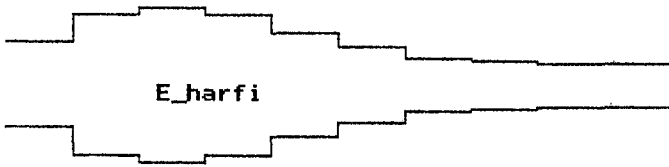
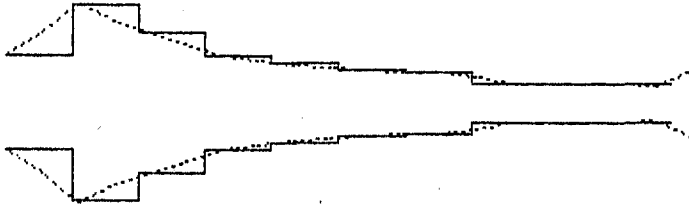
```



```

1 .std $apma=2.79
2 .std $apma=14
3 .std $apma=7.9
4 .std $apma=3.28
5 .std $apma=1.85
6 .std $apma=1
7 .std $apma=0.65
8 .std $apma=0.42
9 .std $apma=0.2
10 .std $apma=0.14

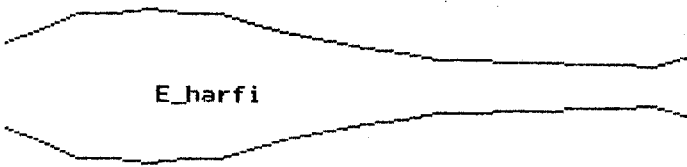
```



```

1 .ALAN=1.21399
2 .ALAN=0.8591
3 .ALAN=2.05323
4 .ALAN=1.48308
5 .ALAN=1.07641
6 .ALAN=0.775632
7 .ALAN=0.681713
8 .ALAN=0.66517
9 .ALAN=0.620609
10 .ALAN=0.620609

```



```

1 .std $apma=1.76
2 .std $apma=5.4
3 .std $apma=6.26
4 .std $apma=6.76
5 .std $apma=3.27
6 .std $apma=1.98
7 .std $apma=0.66
8 .std $apma=0.43
9 .std $apma=0.39
10 .std $apma=0.28

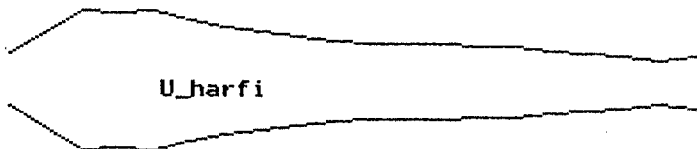
```





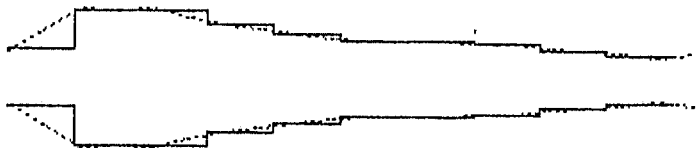
```

1 .ALAN= 0.815639
  .ALAN= 1.96067
  .ALAN= 1.9819
  .ALAN= 1.58476
  .ALAN= 1.28754
  .ALAN= 1.094
  .ALAN= 1.09952
  .ALAN= 1.00862
  .ALAN= 0.831103
10. ALAN= 0.669463
  
```



```

1 . std. S apna= 1.23
  . std. S apna= 6.45
  . std. S apna= 5.6
  . std. S apna= 4.1
  . std. S apna= 3.3
  . std. S apna= 1.6
  . std. S apna= 1.51
  . std. S apna= 1.44
  . std. S apna= 0.76
10. std. S apna= 0.26
  
```



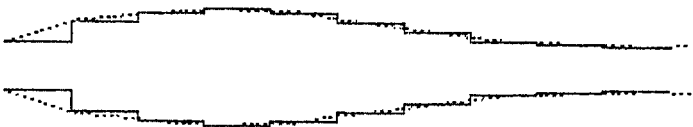
```

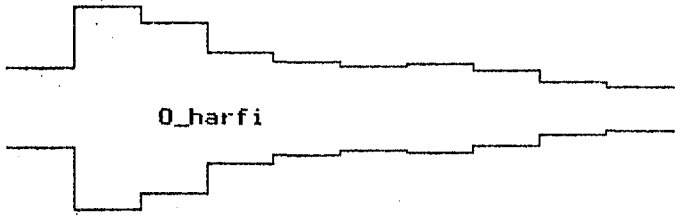
1 .ALAN= 0.69558
  .ALAN= 1.32434
  .ALAN= 1.55536
  .ALAN= 1.69257
  .ALAN= 1.59577
  .ALAN= 1.30863
  .ALAN= 1.00609
  .ALAN= 0.789865
  .ALAN= 0.690988
10. ALAN= 0.653097
  
```



```

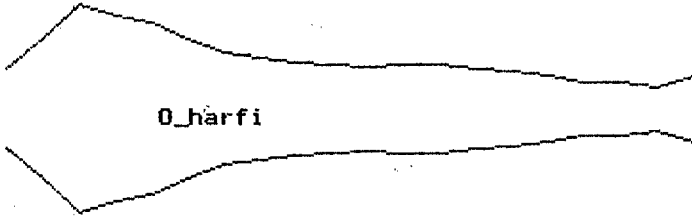
1 . std. S apna= 0.62
  . std. S apna= 0.94
  . std. S apna= 0.55
  . std. S apna= 0.24
  . std. S apna= 0.4
  . std. S apna= 0.33
  . std. S apna= 1.39
  . std. S apna= 0.79
  . std. S apna= 0.5
10. std. S apna= 0.37
  
```





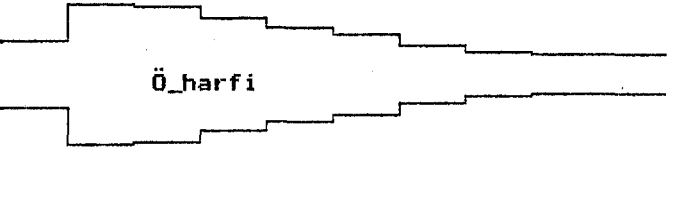
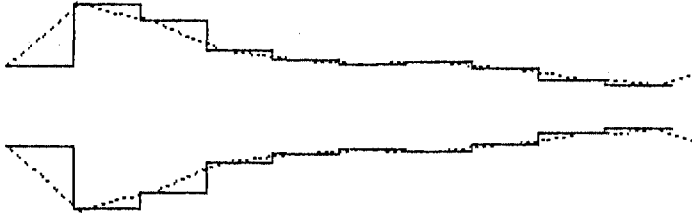
```

1 .ALAN= 1.1915
2 .ALAN= 2.95864
3 .ALAN= 3.47601
4 .ALAN= 1.65453
5 .ALAN= 1.36109
6 .ALAN= 1.25016
7 .ALAN= 1.31348
8 .ALAN= 1.0823
9 .ALAN= 0.797885
10.ALAN= 0.660367
    
```



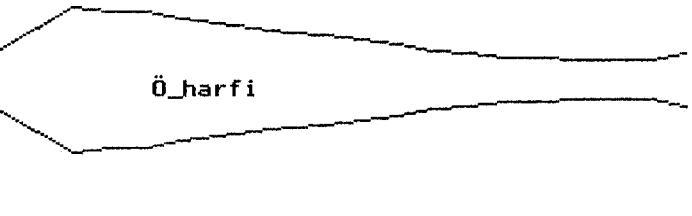
```

1 . std.$apna=1.87
2 . std.$apna=13
3 . std.$apna=6.24
4 . std.$apna=3.88
5 . std.$apna=3.12
6 . std.$apna=1.83
7 . std.$apna=3
8 . std.$apna=1.4
9 . std.$apna=0.58
10. std.$apna=0.27
    
```



```

1 .ALAN= 0.950789
2 .ALAN= 2.06527
3 .ALAN= 1.95441
4 .ALAN= 1.60769
5 .ALAN= 1.37621
6 .ALAN= 1.18211
7 .ALAN= 0.855636
8 .ALAN= 0.625717
9 .ALAN= 0.56419
10.ALAN= 0.561362
    
```



```

1 . std.$apna=1.25
2 . std.$apna=5.11
3 . std.$apna=5.7
4 . std.$apna=4
5 . std.$apna=3.72
6 . std.$apna=3.65
7 . std.$apna=1.2
8 . std.$apna=0.57
9 . std.$apna=0.39
10. std.$apna=0.22
    
```

