

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**DOĞRUSAL ÖNGÖRÜ İLE KONUŞMA İŞARETİ
KODLAYICISI TASARIMI**

Elektronik ve Haberleşme Mühendisi Yılmaz KIRÇIÇEK

**FBE Elektronik ve Haberleşme Anabilim Dalı Elektronik Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Prof. Dr. Vedat TAVŞANOĞLU (YTÜ)

İSTANBUL, 2007

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	1
2. DOĞRUSAL ÖNGÖRÜ İLE KODLAMA - LPC	3
2.1 Konuşma İşaretlerinin Üretilmesi	3
2.2 LPC Modeli	5
2.3 Konuşma İşaretinin Sınıflandırılması	7
2.4 Perde (Pitch) Tespiti	9
2.5 Doğrusal Öngörü Analizi	12
2.5.1 Özilişki Metodu	17
2.5.2 Kovaryans Metodu	19
2.5.3 Doğrusal Öngörü Süzgeç Parametrelerinin Hesaplanması	21
2.6 Öngörü Hata İşareti	24
2.7 LPC Parametrelerinin Deneysel Olarak Değerlendirilmesi	24
2.8 Normalize Öngörü Hatasını Çerçeve Pozisyonu İle Değişimi	28
2.9 Doğrusal Öngörü Analizin Frekans Domeni Yorumu	29
2.10 Doğrusal Öngörü Parametreleri İle Konuşma İşareti Sentezi	30
2.11 SIFT Algoritması	32
2.12 LPC Parametreleri Kullanılarak Formant Analizi	34
2.13 LPC Parametrelerinin Kuantizasyonu	35
3. UYGULAMA	36
3.1 Sistem Değişkenlerinin Girilmesi ve Giriş İşaretinin Seçimi	37
3.2 LPC Analizi	39
3.3 LPC Parametrelerinin Dosyaya Yazılması ve Dosyadan Okunması	45
3.4 LPC Sentezi	48
4. SONUÇ	50

KAYNAKLAR.....	54
EKLER	55
Ek 1 TURTEL kütüphanesi örnek cümleleri.....	56
Ek 2 FS1015 standart LPC-10 algoritması kuantizasyon yapısı	57
Ek 3 MATLAB uygulaması, LPC ana program	58
Ek 4 MATLAB uygulaması, LPC analizi	61
Ek 5 MATLAB uygulaması, LPC parametrelerinin özel formatlı dosyaya yazılması.....	63
Ek 6 MATLAB uygulaması, LPC parametrelerinin özel formatlı dosyadan okunması	64
Ek 7 MATLAB uygulaması, LPC sentezi.....	65
Ek 8 Ses kodlama tekniklerinin karşılaştırılması	66
ÖZGEÇMİŞ.....	67

SİMGE LİSTESİ

a_k	Doğrusal öngörü (polinomu) katsayıları
$AMDF_n(.)$	Ortalama genlik fark fonksiyonu
$e(.)$	Öngörü hata işareti
E_n	İşaretin enerjisi
E_n	Ortalama karesel hata
k_i	Yansıma katsayıları, PARCOR
N	Çerçeve boyutu
p	Öngörü derecesi
$R(.)$	Kısa zaman özilişki fonksiyonu
$\phi_n(.,.)$	Kısa zamanlı kovaryans matrisi
$V^{(.)}$	Normalize öngörü hatası
ZC_n	İşaretin sıfır geçiş sayısı

KISALTMA LİSTESİ

ACR	Absolute Category Rating
ADPCM	Adaptive Differential Pulse Code Modulation
AMDF	Average Magnitude Difference Function
AR	Auto Regressive - Özbağımlı
CELP	Code Excited Linear Predictive
CS-ACELP	Conjugate-Structure Algebraic Code Excited Linear Prediction
FFT	Fast Fourier Transform
GUI	Graphical User Interface
IIR	Infinite Impulse response
Ksps	Kilo sample per second
LD-CELP	Low Delay - Code Excited Linear Prediction
LP	Linear Prediction
LPC	Linear Predictive Coding - Doğrusal Öngörülü Kodlama
MIPS	Mega Instruction Per Second
MOS	Mean Opinion Score
MTRD	Multimedia Technologies Research & Development (UEKAE Lab.)
PARCOR	Partial Correlation
PCM	Pulse Code Modulation
QCELP	QualComm Code Excited Linear Prediction
SIFT	Simple Inverse Filtering Tracking
UEKAE	Ulusal Elektronik Kriptoloji Araştırma Enstitüsü
TURTEL	UEKAE'nin Türkçe konuşma işareti kütüphanelerinden biri (telefon kaydı)
US DoD	United States Department of Defense
Vocoder	İngilizce voice-coder kelimelerinin kısaltılmışı
VSELP	Vector Selectable Excited Linear Predictive

ŞEKİL LİSTESİ

Şekil 2.1 İnsan ses üretim sistemi.....	4
Şekil 2.2 Konuşma İşareti Üretimi İçin İnsan Anatomisi ve Ses Kodlayıcı Analogisi.....	5
Şekil 2.3 LPC analiz ve sentez çerçevelerinin oluşturulması.....	7
Şekil 2.4 Test kelimesindeki ‘e’ harfi (a), test kelimesindeki ‘s’ harfi (b)	7
Şekil 2.5 İşaretteki ofsetin sıfır geçiş sayısına etkisi	8
Şekil 2.6 LPC sentezi blok diyagramı	9
Şekil 2.7 Özilişki fonksiyonu ve AMDF ile ses perdesi tespiti.....	11
Şekil 2.8 Seslendirilmeyen işaret için özilişki fonksiyonu ve AMDF çıktıları.....	12
Şekil 2.9 Doğrusal öngörü sentez filtresi	14
Şekil 2.10 Doğrusal öngörü sentez filtresi	15
Şekil 2.11 Doğrusal öngörü sentez filtreleri.....	24
Şekil 2.12 ‘heed’ kelimesin seslendirilen kısmı için $V^{(p)}$ ’nin p ’ye göre değişimi – I	25
Şekil 2.13 ‘heed’ kelimesin seslendirilen kısmı için $V^{(p)}$ ’nin p ’ye göre değişimi – II	26
Şekil 2.14 ‘heed’ kelimesin seslendirilen kısmı için $V^{(p)}$ ’nin N ’ye göre değişimi **.....	26
Şekil 2.15 Doğal konuşmada seslendirilen kısmı için $V^{(p)}$ ’nin p ’ye göre değişimi – I	27
Şekil 2.16 Doğal konuşmada seslendirilen kısmı için $V^{(p)}$ ’nin p ’ye göre değişimi – II **	27
Şekil 2.17 Doğal konuşmada seslendirilen kısmı için $V^{(p)}$ ’nin N ’ye göre değişimi	28
Şekil 2.18 Üç ayrı sistem için normalize öngörü hatası, V	29
Şekil 2.19 Doğrusal öngörü sentezleyicisinin blok diyagramı.....	31
Şekil 2.20 Eşdeğer kayıpsız tüp modeli - tek ve iki çarpıcı birleşimli	31
Şekil 2.21 SIFT Algoritmasına ilişkin blok diyagram.....	32
Şekil 2.22 Doğrusal öngörü sentez filtresi	32
Şekil 2.23 SIFT algoritmasındaki aşamalardaki örnek sinyaller	33
Şekil 2.24 Temel bir LPC ses kodlayıcı blok diyagramı.....	35
Şekil 3.1 MATLAB uygulaması genel alış diyagramı	37
Şekil 3.2 MATLAB GUI - sistem parametrelerinin alınması	37
Şekil 3.3 MATLAB GUI – giriş işaretinin seçilmesi	38
Şekil 3.4 MATLAB GUI – animasyon sorgusu	39
Şekil 3.5 Ön-vurgulama ve vurgu kaldırma süzgeçlerinin frekans yanıtları.....	40
Şekil 3.6 Ön-vurgulama süzgeci uygulanmayan işlem çerçevesi için frekans spektrumları....	41
Şekil 3.7 Ön-vurgulama süzgeci uygulanan işlem çerçevesi için frekans spektrumları	41
Şekil 3.8 Uygulamada analiz için kullanılan çerçeve yapısı	42

Şekil 3.9 Hamming penceresi.....	42
Şekil 3.10 Kısa zaman özilişki fonksiyonu	43
Şekil 3.11 Analiz animasyon örneği - I	44
Şekil 3.12 Analiz animasyon örneği - II.....	45
Şekil 3.13 Kazanç parametresi ve logaritması.....	47
Şekil 3.14 Seslendiren çerçeveler için uyarım işareti teşkili	49
Şekil 4.1 ACR test verileri için histogram.....	51
Şekil 4.2 Konuşmacılar için ACR test sonuçları	52
Şekil 4.3 Test cümleleri için ACR test sonuçları.....	52
Şekil 4.4 Denekler için ACR test sonuçları.....	53

ÇİZELGE LİSTESİ

Çizelge 1.1 Doğrusal öngörü kullanılan kodlama standartları	3
Çizelge 3.1 Özel tanımlı dosya yapısı	46
Çizelge 3.2 Çerçeve başına bit miktarı- I	48
Çizelge 3.3 Çerçeve başına bit miktarı- II	48
Çizelge 3.4 Mutlak Kategori Test Dereceleri	50
Çizelge Ek2.1 FS1015 kuantizasyon yapısı	57
Çizelge Ek8.1 Ses kodlama tekniklerinin karşılaştırılması	66

ÖNSÖZ

Yüksek lisans eğitimim süresince bizlere emekleri geçen Sayın Hocalarımız'a, tez çalışması süresince özgür araştırma ortamını sağlayan danışman hocam Sayın Prof. Dr. Vedat Tavşanoğlu'na, kullanılan test verilerini sağlayan UEKAE çalışanları Yücel Bicil ve Alper Kanak'a, testlerde yardımlarını esirgemeyen Türk Musikisi Haberleşme Grubu Üyeleri'ne, yüksek lisans eğitim isteğimi anlayışla karşılayarak destek ve izinler veren yöneticilerime, haklarını ödemek için azimle çalışacağım vergileri ile tüm olanaklarından faydalandığım eğitim kurumlarımızın ayakta durmasını sağlayan vatandaşlarımıza ve Yüce Devletimiz'e ve son olarak sevgisini, ilgisini, maddi ve manevi desteğini eksik etmeyen aileme sonsuz teşekkürler ederim.

ÖZET

İnsan konuşma sistemi üzerine bina edilen ve matematiksel araç olarak doğrusal öngörüü kullanan model tabanlı bir konuşma işareti kodlayıcı MATLAB programı üzerinde gerçekleştirilmiştir.

Gerçek zamanda çalışmayan sistem, kayıtlı konuşma işaretlerini doğrusal öngörü ile kodlayarak (analiz) tanımlanan biçimleme ile .lpc uzantılı dosyaya yazar, sonra dosyayı okuyup kodu çözerek (sentez) giriş-çıkış işaretlerini kullanıcıya dinletir ve görsel animasyonlar yapar. Ses kodlayıcı 3.84 kb/s ve 2.56 kb/s olmak üzere iki veri çıkış hızında çalışabilmektedir.

Testlerde Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü'nden temin edilen Turtel Kütüphanesi'nin bir alt kümesi (5 erkek 5 kadın konuşmacının 15'er cümlesi) ve 3.84 kb/s çıkış hızına sahip ses kodlayıcı kullanılmıştır. Yaş ortalamaları 31 olan 8 erkek 7 kadın katılımcı ile yapılan Mutlak Kategori Test'i sonucunda ses kodlayıcın kalitesi 3.4 olarak tespit edilmiştir.

Anahtar Kelimeler : Model tabanlı ses kodlayıcı, doğrusal öngörü, LPC, MATLAB

ABSTRACT

A speech coder was designed on MATLAB environment based on human speech production system that uses linear prediction as mathematical tool.

System does not work in real time. It processes the recorded speech signals by coding with linear prediction (analysis), writes to a file in a predefined format with .lpc extension. Then it reads and decodes the file (synthesis), plays the input-output signal and shows some visual animations. System can work 3.84 kb/s or 2.56 kb/s data output rates.

A subset of the Turtel Library (5 male and 5 female speaker, 15 sentences of each) provided by National Electronic and Cryptology Research Institute and 3.84 kb/s data rated vocoder was used in tests. Sound quality of the vocoder is defined as 3.4 with Absolute Category Rating test by participation of 8 men and 7 women audiences whose average age is 31.

Keywords: Model based speech coder, linear prediction, LPC, MATLAB

1. GİRİŞ

Doğrusal öngörülü kodlama (LPC), konuşma işaretlerinin kayıplı olarak kodlanmasında kullanılan ve isminden de anlaşılacağı üzere, işarete ait bir örneğin, önceki örneklerin doğrusal bir fonksiyonu olarak öngörülebilmesi esasına dayanan bir yöntemdir. İlk kez ABD (US DoD – FS1015 LPC-10 Vocoder) tarafından konuşma işaretlerinin kodlanması için 1984 yılında önerilmiştir.

Temel olarak, konuşma işaretlerinin üretildiği insanın vokal sistemi, çapı değişebilen bir tüp olarak modellenip, bu tüpün temel matematiksel parametreleri üzerine LPC modeli bina edilmiştir. Doğrusal öngörü ile belirli bir t (n_t . örnek) anında, işaret örneği $s(n_t)$, bu işarete ait önceki p örneğin doğrusal fonksiyonu olarak elde edilebilir. Bu durumda, eğer $s(n_t)$ ve işarete ait önceki p örnek mevcut ise, ayrıca daha sonra detaylandırılacak olan şartlar da mevcut ise, işaretin ilgili çerçevesi ile ilgili bir bağıntı ve bu bağıntının parametreleri elde edilmiş olur. Doğrusal öngörü kodlaması ile analiz ve sentez kısımları arasında bu birkaç parametreler iletilir. Genelde konuşma işaretleri 8000 örnek/saniye ile örneklenir ve örnekler 8 bit ile temsil edilir. Bu durumda veri hızı 64000 bit/saniye olmaktadır. LPC ile bu hız 2400 bit/saniye'ye düşürülebilir. Bu düşük hızda kodlanan konuşma işareti hissedilir derecede sentetiktir ve ses kalitede belirgin azalma vardır. Yani LPC kayıplı bir sıkıştırma tekniğidir. Fakat yine de konuşma işareti anlaşılabilir düzeydedir.

Farklı teknikler kullanan pek çok konuşma işareti sıkıştırma yöntemi vardır. Yine de çoğu metot konuşma işaretinin yavaş anatomik hareketler ile oluşması ve sınırlı frekans bandına (300 ila 3400 Hz) sahip olması ilkesi üzerine kurulmuştur. Konuşma işaretinin sıkıştırılması, işaretin daha az veri ile kodlanması olarak tanımlanabilir. Dolayısıyla çoğu kodlama yöntemi kayıplıdır.

Genelde konuşmanın %50'sinden fazlası sessizlik ile geçer. (Kelimeler ve cümleler arası boşluklar, nazik insanlarda – herksin bir ağızdan konuşmadığı ortamlarda – konuşmacı değişimi arasındaki boşluklar vb.) Bant genişliğini düşürmenin bir yolu da bu sessiz anların kodlamada dikkate alınmasıdır. Ayrıca konuşma işaretlerinin ardışık örneklerinin arasında yüksek korelasyon vardır. Pek çok yöntem konuşma işaretinin üretilmesini doğrusal filtreler ile modeller ve bu filtrelerin yavaş değişen parametreleri de konuşma işaretini temsil eden kodlar olarak kullanır.

LPC konuşma işaretlerinin üretilmesini model olarak kullanan kodlama-sıkıştırma yöntemlerinden biridir. LPC konuşma işaretlerinin üretilmesini bir özbağımlı (AR) süreç

olarak modeller ve kodlamada konuşma işareti yerine bu modelin parametrelerini kullanır. Bu yöntem ilk defa United States Department of Defense tarafından federal standart (FS-1015) olarak önerildi ve 1984 yılında yayımlandı. Bu standart 10. dereceden doğrusal öngörü kullanıldığından LPC-10 olarak da anılır.

Ses kodlayıcılar (vocoder – voice coder) model tabanlı ve dalga şekli tabanlı olmak üzere iki ana gruba ayrılırlar. Dalga şekli kodlayıcıları kuantalama hataları haricinde ses işaretinin zaman domeninde dalga şeklini tam olarak korur. Model tabanlı kodlayıcılar ise, kuantalama hatası olsun veya olmasın, dalga şeklini koruyamaz, yalnızca işarete ait model parametrelerini kullanırlar.

Ses kodlayıcılarının *bit oranı*, *gecikme*, *komplekslik* ve *kalite* olmak üzere 4 ana parametresi vardır. Kullanılan algoritma ne olursa olsun tüm ses kodlayıcılar bu dört parametre arasında bazılarını diğerlerine tercih ederek, amaca özel tasarlanır. Bu parametrelerden bit oranı sıkıştırma başarımının ölçüsüdür. 8 bit ile 8 kHz 'de örneklenen işaret 64 kb/s gibi bir veri hızına sahiptir. Bundan küçük değerler sıkıştırılmış olarak kabul edilir. LPC ile 2.4 kb/s gibi çok iyi bir sıkıştırma gerçekleştirilir. Her ne kadar depolama amaçlı kullanımda kodlama gecikmesi çok önemli bir kısıt olmasa da, gerçek zamanlı veri iletiminde, konuşma işaretinin anlaşılır olması için, gecikme 300ms'den az olmalıdır. Kodlama algoritması kompleksliği ise hem maliyet hem de gerekli işlemci gücü açısından belirleyici olmaktadır. En son parametre olan kalite ise neredeyse tamamen öznedir. Kalite konusunda yapılan testler Mutlak Kategori Testleri (ACR, Absolute Category Rating) olarak adlandırılır. Dinleyiciler sonuçları *mükemmel*, *iyi*, *orta*, *zayıf* ve *kötü* olarak sınıflandırır. LPC algoritmasında düşük bit oranları elde edebilmek için kaliteden bir miktar feragat edilir ve sonuçta ses hissedilir derecede sentetiktir.

Genel LPC algoritması analiz (kodlama) ve sentez (kod çözme) birimlerinden oluşur. Analiz kısmında konuşma işareti çerçeveler halinde işlenerek model parametreleri elde edilir. Elde edilen bu parametreler kuantalanarak sentez kısmına gönderilir. Sentez kısmında model parametreleri ile süzgeç gerçekleştirilir ve bu süzgeç uygun şekilde uyarılarak konuşma işaretinin tekrar elde edilmesine çalışılır. Analiz ve sentez kısımları arasında gerekli olan diğer veriler ise kazanç ve uygulanacak uyarım biçimidir.

Bu tez çalışmasında elde edilmek istenen sonuç Mutlak Kategori Test'ine göre orta ve üstünde ses kalitesine sahip, 2.4 kb/s mertebesinde sıkıştırma yapabilen bir ses kodlayıcının MATLAB programı üzerinde gerçekleştirilmesidir.

Ses ve konuşma işaretlerinin sıkıştırılması 1930’larda darbe kod modülasyonunun (PCM, Pulse Code Modulation) geliştirilmesi ile başlar. Fakat asıl gereksinim 1960’larda telefon şirketlerinin sınırlı bant genişliğinin verimli kullanılması isteği ile ortaya çıkmıştır. 1970’li yıllara kadar konuşma işareti kodlama çok popüler olmamakla beraber 40 yıllık bir geçmişe sahipti. İlk ses kodlayıcı 1939 yılında Homer Dudley tarafından Bell Laboratuvarlarında yapılmıştır. İlk konuşma sentezi ise 1930’larda Kempelen Farkas Lovag tarafından yapılmıştır. İlk LPC algoritması 1970’li yıllarda gerçekleştirilmiştir. 1984 yılında ise standartlaştırılmıştır. Sonraki yıllarda LPC ’nin CELP (Code Excited Linear Predictive), VSELP (Vector Selectable Excited Linear Predictive) gibi pek çok türevi geliştirilmiştir. Çizelge 1.1 ‘de doğrusal öngörü kullanılan konuşma işareti kodlama standartları verilmiştir.

Çizelge 1.1 Doğrusal öngörü kullanılan kodlama standartları

FS1015-LPC10	10 Coefficient LP
FS1016-CELP	Code Excitation LP
MELP	Mixed Excitation LP
IS-54 VSELP	Vector Sum Excited LP
IS-96 QCELP	QualComm Code Excited LP
LD-CELP G.728	Low Delay Code Excited LP
G.729 CS-ACELP	Conjugate-structure Algebraic-Code-Excited LP

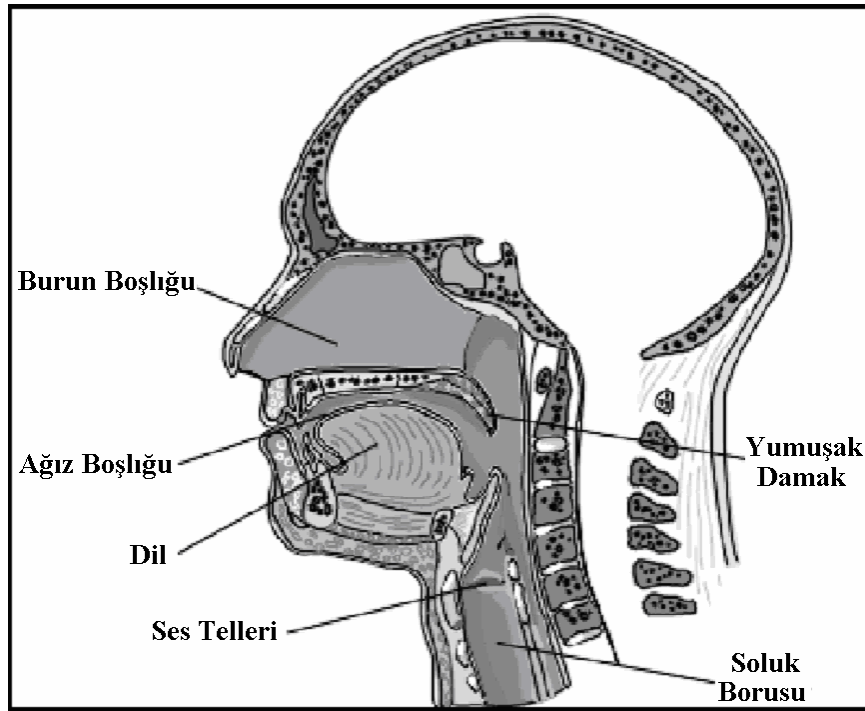
2. DOĞRUSAL ÖNGÖRÜ İLE KODLAMA - LPC

2.1 Konuşma İşaretlerinin Üretilmesi

Konuşulan dil ne olursa olsun tüm insanlar konuşmak için aynı anatomiye kullanırlar. Çıkarabileceğimiz sesler anatomimiz nedeniyle sınırlıdır. Yalnızca bazı lisanlarda gırtlak ve burun sesleri daha fazla kullanılır. Konuşma işaretinin üretilmesini akciğerlerin havayı vokal sisteme ve oradan ağza pompalaması şeklinde çok kabaca tarif edebiliriz. Bu tanıma göre akciğer ses kaynağı, vokal sistem ise çeşitli sesleri üreterek konuşma işaretini üreten bir filtre olarak düşünülebilir.

Vokal sistemin akciğerden aldığı havayı sese nasıl dönüştürdüğünü anlamak için bazı temel tanımlamaları anlamak gerekir. *Fon* dildeki en küçük ayırıcı öğedir. Türkçe fonetik bir dil olduğundan her harf bir fonem olarak düşünülebilir. Fonların oluşturduğu sınırlı-bağımsız kümelere de *fonem* denir. Seslendirilen ve seslendirilmeyen olmak üzere iki çeşit fonem vardır ve LCP sentez ve analizinde bu farklılık uyarım işaretinin belirlenmesinde kullanılır. (Bu kavramlar hakkında daha sonra detaylı bilgi verilecektir.)

Seslendirilen fonemler genellikle sesli harflerdir, yüksek ortalama enerji seviyeleri ve farklı rezonans (formant) frekansları vardır. Seslendirilen fonemler akciğerlerden gelen havanın ses tellerini periyodik titreştirmesi ile oluşurlar. Titreşimin frekansı üretilen sesin perdesini (pitch) belirler. Bu titreşimler sonucu oluşan hava darbeleri vokal sistemin içerisinden geçer ve bazı frekanslarda rezonanslar oluşur. Genellikle kadın ve çocuklarda oluşan sesler daha yüksek perdedendir, dolayısıyla daha yüksek frekanslıdır. LPC analiz ve sentezinde sesin perdesi (pitch) de bir parametredir.



Şekil 2.1 İnsan ses üretim sistemi

Fonemlerin sınıflandırılması LPC analiz ve sentezi açısından önemlidir. Esasen ses tellerinin titreşip titreşmemesi farklı tür seslerin üretilmesi açısından önemli bir parametredir. Ses üretimini etkileyen diğer bileşen vokal sistemin şeklidir. Geometriye göre farklı sesler ve rezonans frekansları oluşur. Vokal sistem gırtlak, dil, burun ve ağızdan müteşekkildir. İnsan konuştuğu vokal sistem düşük hızlarda, sürekli olarak şekil değiştirerek, sesleri ve kelimeleri oluşturur. Konuşmanın oluşmasına etki eden diğer parametre de ise akciğerlerden gelen havanın miktarıdır. Artan hava miktarı ile sesin yüksekliği de artar.

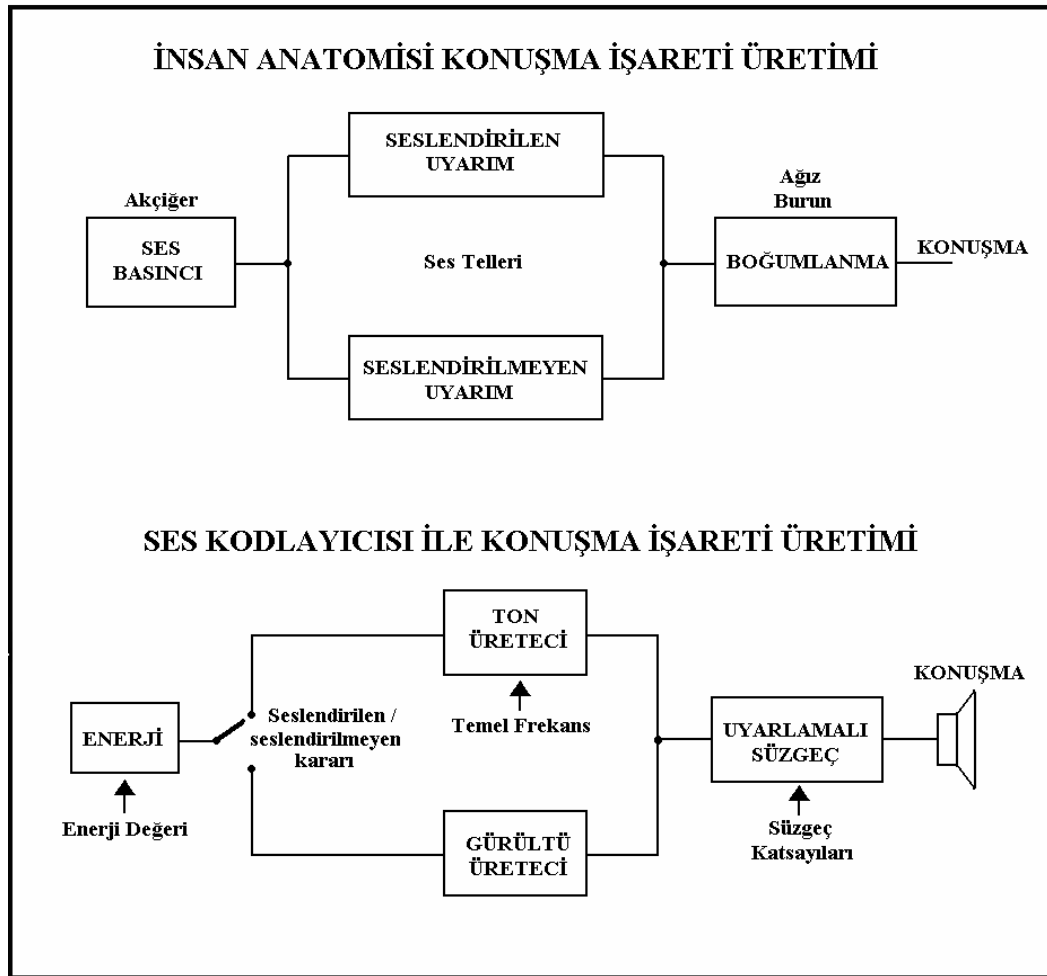
Akciğerden gelen havanın ses kaynağı, vokal sistemin filtre olarak kabul edilmesi fikrine, ses üretiminde kaynak-filtre modeli denir. Bu model LPC yönteminde konuşma

işaretinin analizinde de sentezinde de kullanılır. *A Practical Handbook for Speech Coders* (1999) isimli kitapta yazarlar Randy Goldberg ve Lance Riek modelleme kavramına şu şekilde bir açıklama getirmişlerdir; “Ses dalgaları partiküllerin titreşmesi ile havada (veya diğer ortamlarda) yayılan basınç değişimleridir. Bu dalgaların ve yayılımının modellenmesi vokal sistemin uyarıcı sinyalin frekans içeriğini nasıl şekillendirdiğini tanımlamayı sağlamak için bir iskelet oluşturur.”

2.2 LPC Modeli

LPC modeli iki temel bileşen içerir; analiz (kodlama) ve sentez (kod çözme). Analiz kısmında konuşma işareti çerçeveler halinde incelenir ve aşağıdaki soruların cevap aranır;

- Çerçeveler seslendirilen mi seslendirilmeyen mi ?
- Çerçevenin perdesi (pitch) nedir ?
- Çerçeve için vokal sistemi modelleyen filtrenin parametreleri nelerdir?



Şekil 2.2 Konuşma İşareti Üretimi İçin İnsan Anatomisi ve Ses Kodlayıcı Analojisi

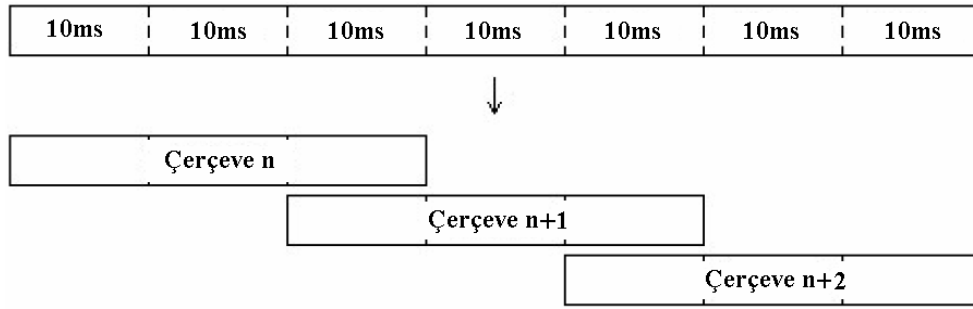
LPC model parametreleri analiz tarafında elde edilir ve sentez kısmına aktarılır. Sentez kısmında yalnızca bu parametreler kullanılarak vokal sistemi temsil eden filtre gerçekleştirir. Şekil 2.2 'de ses kodlayıcısı ile konuşma işareti üretimi ile insan anatomisinin bir analojisi verilmiştir. Akciğerden gelen hava miktarını kazançla, ses tellerini sesin perdesi ve sınıflandırmasıyla (yani uyarım işaretiyle) ve ağız-burun-yutak geometrisini de modeli oluşturan süzgeç ile bağdaştırabiliriz. Bu yapı yalnızca LPC için değil tüm ses kodlayıcılar için geçerlidir.

Tüm ses kodlayıcıları iki şey üzerine yoğunlaşmıştır; uyarım işareti ve boğumlanma. Boğumlanmadan kasıt akciğerden gelen havanın ağızdan dışarı çıkarken ağzın, dilin, yutağın yani tüm ağız boşluğunun hareket ve şekillenmesidir. Uyarım işareti filtreye veya vokal sisteme uygulanan giriş işareti, boğumlanma ise uyarım işaretinin konuşma işaretine transformasyon işlemidir.

Doğrusal öngörü tabanlı konuşma işareti kodlama yöntemlerinin pek çok türevi bulunmaktadır. Bu yöntemlerde vokal sistem modellendikten sonra genelde değişen sentez esnasında kullanılacak uyarım sinyalıdır. Ses kalitesini ve bit oranlarının asıl belirleyen de esasen budur. Örneğin RELP (Residual Excited Linear Predictor) yönteminde sesin perdesinin (pitch) ve niteliğinin (seslendirilen / seslendirilmeyen) belirlenmesine bile gerek yoktur.

Fiziksel ve anatomik yapımızdan dolayı çıkardığımız sesler çok hızlı değişmemektedir. Örneğin önce kalın ve alçak sesle bir şey söylerken birden çok ince ve yüksek sesli bir çığlık atmamız en az 100 ila 200ms sürmektedir. Bu durumda bile, ürettiğimiz işaretin korelasyonu yüksek olsa da, bu işaretin durağan olmadığı aşıkardır. Ama konuşma işaretleri kısa çerçeveler (10-30ms) halinde ele alınırsa bu durumda geniş anlamda durağan işaretler elde edilebilir. Fakat bu durumda dahi 100 ila 200ms'de bir değiştiğinden bahsettiğimiz ses özelliklerinin değişiminin tam çerçevenin ortasında gerçekleşmemesi garanti edilemez.

Bazı çerçeveler için ses özelliklerinin (perde/seslendirilen/seslendirilmeyen) tam ve doğru olarak çıkarılmaması mümkündür. Ses özelliklerinin yavaş değiştiğini söyleyip çerçeveden çerçeveye bu özelliklerin aniden değiştiğini söylemek de kendimizi yadsımak olacağından, bu geçişleri biraz yumuşatmak için çerçeveler birbirlerinin içine geçecek şekilde oluşturulurlar. Örneğin 30ms'lik çerçeveler seçildiğinde çerçeve yapısı Şekil 2.3' teki gibi olacaktır;

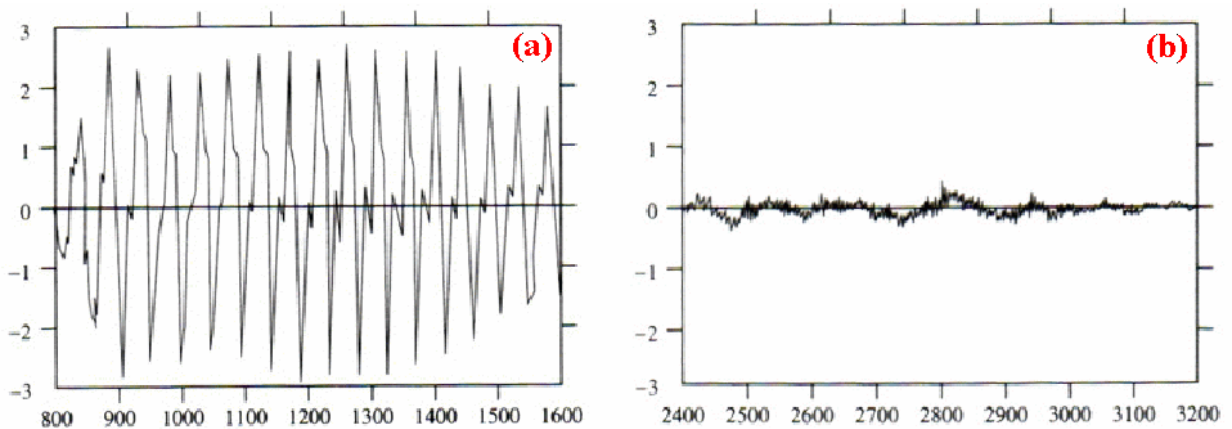


Şekil 2.3 LPC analiz ve sentez çerçevelerinin oluşturulması

Konuşma işareti çerçevelere bölündükten sonra yapılacak işlem her çerçeve için analiz parametrelerinin elde edilmesidir. Uygulamada görülmüştür ki, temel çerçeve boyutları aynı kalmakla beraber analiz ve sentezde farklı türde teşkil-örtüşme yapıları kullanmak daha iyi sonuçlar vermektedir. Bu husus ileriki bahislerde detaylandırılacaktır.

2.3 Konuşma İşaretinin Sınıflandırılması

Daha önce de belirtildiği gibi seslendirilen fonemler genelde sesli harflerdir ve periyodik dalga olarak düşünülebilir. Bu sesler yüksek ortalama enerjiye sahiptir yani genlikleri büyüktür. Ayrıca farklı rezonans (formant) frekansları vardır. Şekil 2.4a 'da test kelimesindeki 'e' harfine ait dalga şekli verilmiştir. Seslendirilmeyen fonemler genellikle sessiz harflerdir. Kaotik ve rasgele dalga şekilleri vardır. Şekil 2.4b 'de de test kelimesindeki 's' harfine ait dalga şekli verilmiştir. Şekil 2.4'ün incelenmesi seslendirilen / seslendirilmeyen sınıflandırılması için izlenebilecek yöntemler hakkında genel bir fikir vermektedir.



Şekil 2.4 Test kelimesindeki 'e' harfi (a), test kelimesindeki 's' harfi (b) *

* Bradbury J.S., (2000), "Linear Predictive Coding", Teknik Rapor, Queen's University, Ontario. Sayfa 10

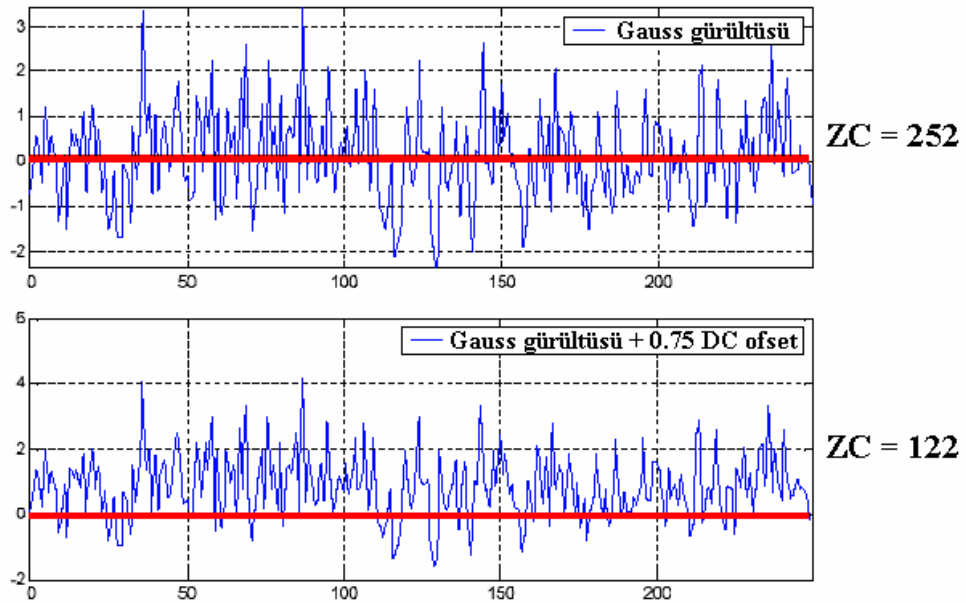
İşaretin genliği, bir anlamda enerjisi, bu sınıflandırma kararı için gözlenebilecek değişkenlerden biridir. Ne var ki gırtlak ve dudak sesleri gibi bazı özel sesler için bu yöntem daima doğru sonuç vermeyebilir. n incelenen çerçeveyi, N çerçeve genişliğini ve m de çerçevedeki örnekleri göstermek üzere çerçeve için enerji aşağıdaki gibi tanımlanabilir ;

$$E_n = \sum_{m=n-N+1}^n x^2(m) \quad (2.1)$$

Şekil 2.3 ‘ten görüleceği üzere seslendirilmeyen işaretler daha yüksek frekanstadırlar ve dolayısı ile daha yüksek oranda sıfır geçişi yapmaktadır. Bu nitelik de sınıflandırma kistası olarak kullanılabilir. Ses işareti için elde edilecek sıfır geçiş oranları da incelenen çerçevede işaretin frekansı hakkında yaklaşık bir bilgi verecektir. Örneğin salt sinüzoidal bir işaret için frekans sıfır geçiş sayısını yarısı olacaktır. n incelenen çerçeveyi, N çerçeve genişliğini ve m de çerçevedeki örnekleri göstermek üzere sıfır geçiş sayısı aşağıdaki şekilde elde edilebilir ;

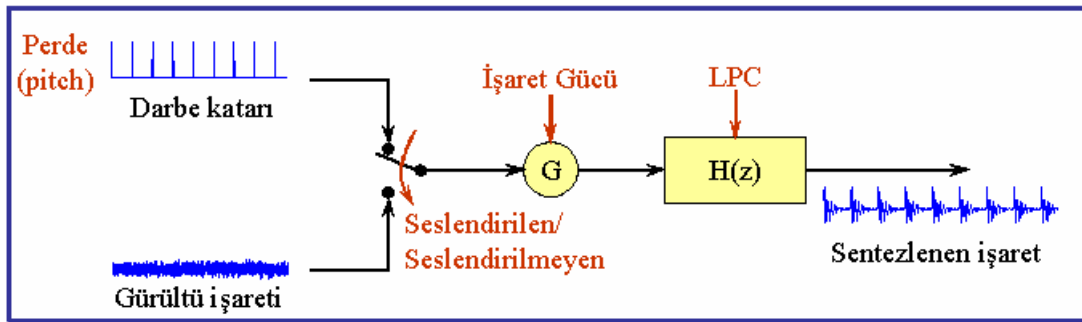
$$ZC_n = \sum_{m=n-N+1}^n |\text{sgn}[x(m)] - \text{sgn}[x(m-1)]| \quad (2.2)$$

Konuşma işaretinin sınıflandırılmasında sıfır geçiş sayısı kullanılacak ise de işaret üzerindeki DC ofsetten kurtulmak gerekir. Şekil 2.5’de yaklaşık aynı işaret için DC ofsetin etkisi gösterilmiştir.



Şekil 2.5 İşaretteki ofsetin sıfır geçiş sayısına etkisi

Ayrıca bu sınıflandırma yapılırken komşu çerçeveler de göz önünde bulundurulmalıdır. Ses parametreleri çok hızlı değişmeyeceğinden uzun bir seslendirilmeyen çerçeve gurubunun arasında bir seslendirilen çerçeveye karar kılınmış ise bu hata komşu çerçevelere bakılarak düzeltilebilir. Şu da unutulmamalıdır ki, LPC modelinin asıl amacı düşük bit oranı elde etmek olduğundan, model insanın çıkardığı tüm sesleri temsil edemez. (Burun sesleri, konuşurken oluşan soluklanma sesleri, gırtlak ve dudak sesleri vb.) Bu nedenle her çerçeve seslendirilen/ seslendirilmeyen gibi basit sınıflandırmaya tabii olmayabilir. Fakat yalın LPC model bunları düşük bit oranları uğruna göz ardı eder ve sonuç ses kalitesinde düşme olarak kendini gösterir. CELP (Code Excited Linear Prediction) gibi yöntemler ile bu sorunun üstesinden gelinebilir.



Şekil 2.6 LPC sentezi blok diyagramı

Şekil 2.6'dan da görüldüğü üzere seslendirilen / seslendirilmeyen ayrımı, tanımlanan modelde uyarım sinyalini belirlediği için önemlidir. Eğer işlenen çerçevedeki işaret seslendirilen ise, sentezde uyarım işaret olarak, çerçeve için belirlenen perdede (frekansta) darbe katarı uygulanır. İşlenen çerçevedeki işaret seslendirilmeyen ise uyarım işareti olarak gürültü işareti uygulanır. Uyarım işareti daha sonra detaylandırılacaktır.

2.4 Perde (Pitch) Tespiti

Konuşma işareti için perde genelde (temel frekans, pitch) 150 ila 400 Hz civarındadır. Çocuklar ve kadınlar için 120-500 Hz olabileceği gibi erkekler için 50 ile 250 Hz arasında değişebilir. Ses işareti temel frekans bileşeni, onun harmonikleri ve diğer frekans bileşenlerinden oluşmaktadır. Bu işaretin özilişki fonksiyonu, işaretteki benzerliklerin yüksek olduğu yerlerde, temel bileşenin harmoniklerinde, yüksek değerler alacaktır. Özilişki fonksiyonundaki bu tepe değerler takip edilerek sesin perdesi bulunabilir. Çerçeveler için kullanılacak kısa zamanlı özilişki fonksiyonu; N çerçeve boyutunu göstermek üzere aşağıdaki gibi tanımlanır;

$$r(k) = \frac{1}{N} \sum_{n=k}^{N-1} s(n) \cdot s(n-k) = \frac{1}{N} \sum_{n=0}^{N-1-k} s(n) \cdot s(n+k) \quad (2.3)$$

Baştaki $1/N$ normalizasyon katsayısı kullanılmayabileceği gibi, tepe bulmada kolaylık sağlamak amacıyla $1/(N-k)$ ile ikame edilerek genliklerdeki azalma kompanse edilebilir. $k=0$ için özilişki fonksiyonu işaretin gücünü vermekte, $k=\pm p, \pm 2p, \pm 3p \dots$ noktalarında (p temel bileşenin periyodu) ise tepe değerler oluşmaktadır. Özilişki fonksiyonuna ait elde edilen grafikleri, kullanılan bir başka yöntemi de (AMDF) daha tanıttıktan sonra karşılaştırma açısından birlikte vereceğiz.

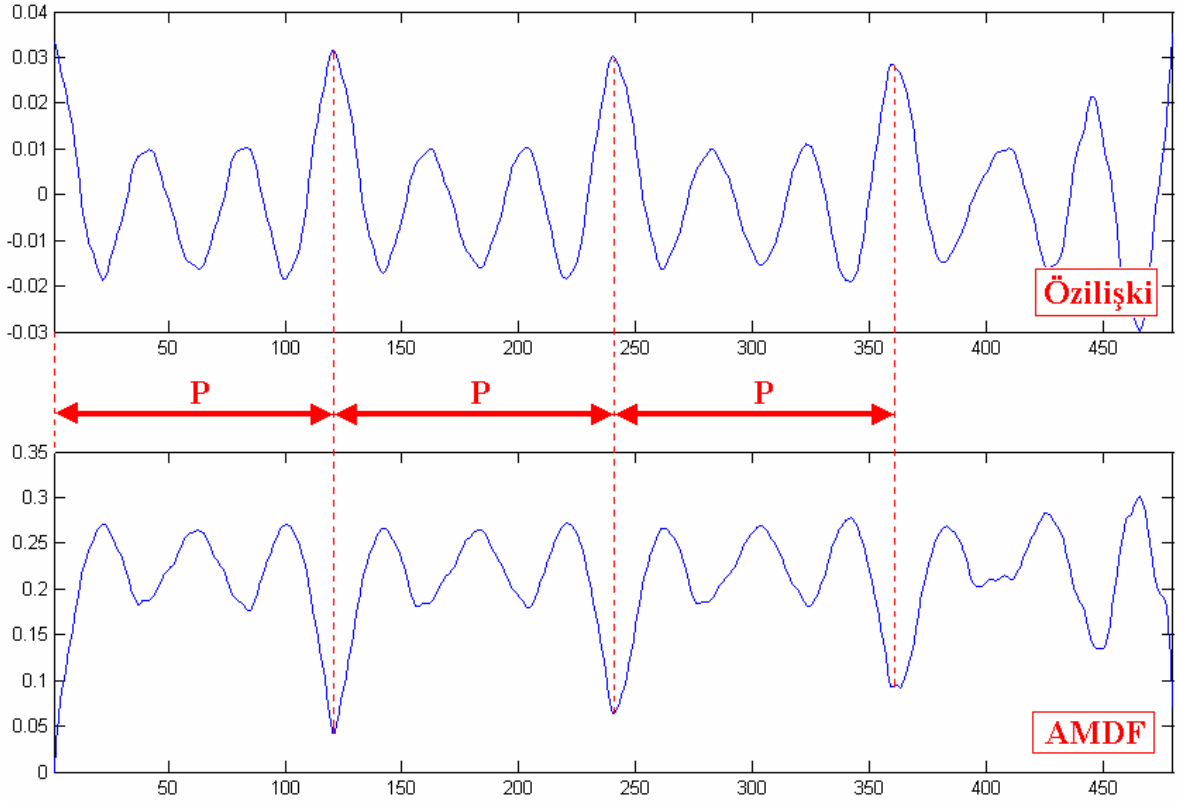
Periyodik bir işaretle, işaretin periyodu kadar aralıkla yer alan örnekler birbirine eşit olacaktır. Eğer işaret temel bileşenin yanında farklı frekans bileşenleri içeriyor ise, işaretin temel bileşenin periyodu kadar mesafedeki örnekler eşit olmasa da birbirlerine yakındır. Bu özellikten yararlanılarak tanımlan bir fonksiyon ile, Ortalama Genlik Fark Fonksiyonu (AMDF, Average Magnitude Difference Function), sesin perdesi (pitch) tayin edilebilir. N çerçeve boyutunu, n de aktif çerçeveyi göstermek üzere AMDF şu şekilde tanımlanır ;

$$AMDF_n(k) = \sum_{m=n-N+1}^n |x(m) - x(m-k)| \quad (2.4)$$

Özilişki fonksiyonunu aksine AMDF $k=0$ için minimum değerini alır ve $k=\pm p, \pm 2p, \pm 3p \dots$ noktalarında (p temel bileşenin periyodu) ise negatif tepe değerler oluşmaktadır. AMDF'nin özilişki fonksiyonuna göre tek avantajı, yalnızca toplama ve çıkarma kullanılarak hesaplanabileceğinden hızı arttırıp, işlem yükünü azaltmasıdır.

Eğer işlenen çerçevede bir periyodiklik söz konusu değil ise (seslendirilmeyen işaret ise) özilişki fonksiyonu da AMDF de periyodiklik oluşmayacaktır. Şekil 2.7 ve Şekil 2.8'de yapılan denemelerde seslendirilen ve seslendirilmeyen işaretler için elde edilen ($F_s=16000$, $N=480$, yani 30ms'lik bir çerçeve için) özilişki ve AMDF hesaplamalarından elde edilen grafikler verilmiştir.

Seslendirilmeyen ve seslendirilen işaretler için verilen özilişki fonksiyonu ve AMDF çıktıları modelde kabul edilen yaklaşımlara uygun şekilde üretilmiş ses örneklerinden elde edildiklerinden çok açık ve belirgin şekilde perde tespiti mümkün olmaktadır. Fakat ne yazık ki, defaten belirtildiği üzere konuşma işaretindeki tüm çerçeveler için bu denli ayırt edilebilir çıktılar elde etmek mümkün olmamaktadır.

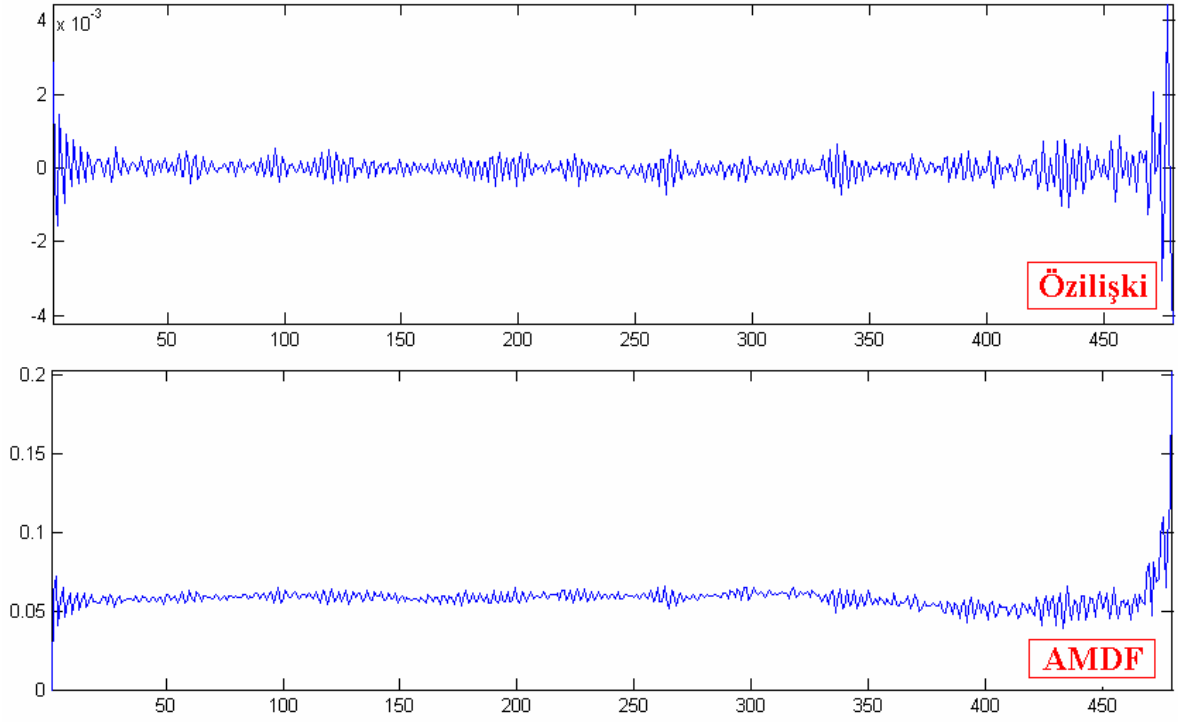


Şekil 2.7 Özilişki fonksiyonu ve AMDF ile ses perdesi tespiti

Doğrudan işlenen çerçevedeki işarete inceleme yapmak yerine, bu işareti tasarlanan doğrusal öngörü filtresinden geçirdikten sonra elde edilecek hata işareti incelenerek yapılan ses perdesi tespit yöntemleri de vardır. İleriki bahislerde matematiksel model tanıtıldıktan sonra görülecektir ki, hata işareti aslında ideal uyarım işaretidir. Bu nedenle perde tespiti ve ses sınıflandırması için hata işareti kullanımı uygun olmaktadır. Ne var ki, öngörü derecesi arttıkça, hata işareti küçüleceğinden, gerekli parametrelerin çıkarımı zor olmaktadır. Uygulamada da bu durum tecrübe edilmiştir.

Bu yöntemlerin yanında, hesap yükünü çok arttırmakla beraber, ses perdesinin tespitinde frekans domeni incelemeleri de kullanılabilir. Bunu için işaretin Fourier Dönüşümü alınıp, işaret seslendirilen ise, temel bileşen ve harmoniklerin yerlerine bakılarak ses perdesi (pitch) tespit edilebilir.

Vokal sistem modelinin diğer bir ögesi de doğrusal öngörü filtresidir. Esasen bazı yöntemler ses perdesi tespitinde vokal sistemi modelleyen doğrusal öngörücü filtreyi kullandığından, önce bu filtrenin teşkil edilmesi yoluna gidilebilir.



Şekil 2.8 Seslendirilmeyen işaret için özilişki fonksiyonu ve AMDF çıktıları

2.5 Doğrusal Öngörü Analizi

Doğrusal öngörü; rasgele bir sürecin (örneğin konuşma işaretlerinin) örnek değerlerinden oluşan bir küme verildiğinde, bu durağan ayırık zamanlı rasgele sürecin ileri veya geri yönde bir sonraki veya önceki değerinin kestirilmeye çalışılmasıdır.

$\{ u(n) \ u(n-1) \ u(n-2) \ ... \ u(n-M) \}$ durağan ayırık zamanlı rasgele bir sürecin n anına kadar ve n anından önceki toplam $M+1$ tane değerini göstermek üzere ; $\{ u(n-1) \ u(n-2) \ ... \ u(n-M) \}$ değerlerini kullanarak $u(n)$ değerini elde etmeye ($u(n-1)$ anına göre) ileri yönlü tek adımlı doğrusal öngörü, $\{ u(n) \ u(n-1) \ u(n-2) \ ... \ u(n-M+1) \}$ değerlerini kullanarak $u(n-M)$ değerini elde etmeye ($u(n-M+1)$ anına göre) geri yönlü tek adımlı doğrusal öngörü adı verilir.

Şimdi adım adım, tek adımlı ileri yönlü doğrusal öngörücü kullanarak konuşma işaretlerinin sentez ve analizinde kullanılan vokal sistem filtresinin (doğrusal öngörü katsayılarının) bulunmasını inceleyelim;

$$\hat{s}(n) \approx a_1 s(n-1) + a_2 s(n-2) + ... + a_p s(n-p) = \sum_{k=1}^p a_k s(n-k) \quad (2.5)$$

Denklem (2.5) ifadesine göre öyle a_k katsayıları tespit edilebilir ki, $s(n)$ örneği, önceki p adet örneğin doğrusal birleşimi olarak tanımlanabilir. a_k katsayılarının işlenen çerçeve boyunca sabit olması kabulü ile (sürecin durağan olması) sinyalimiz bir özbağlanımlı (AR, Auto Regressive) rasgele süreç olarak modellenebilir. Özbağlanımlı süreç, girişi beyaz gürültü olan tüm kutup süzgecin çıkışı olarak basitçe tanımlanabilir. $\{ u(n) \ u(n-1) \ u(n-2) \dots u(n-M) \}$ gibi bir zaman serisi aşağıdaki (2.6) eşitliğini sağlıyor ise, M . dereceden özbağlanımlı (AR) sürecin gerçekleşmesini gösterir. Burada a_i sabit AR parametreleri, $v(n)$ ise beyaz gürültüdür.

$$s(n) + a_1^* s(n-1) + a_2^* s(n-2) + \dots + a_M^* s(n-M) = v(n) \quad (2.6)$$

Bu durumda $s(n)$ işareti ;

$$s(n) = \sum_{k=1}^p a_k s(n-k) + Gu(n) \quad (2.7)$$

şeklinde ifade edilebilir. Burada G kazancı, $u(n)$ beyaz gürültü (uyarım işaretini) göstermektedir. Şimdi yukarıdaki ifadeyi z domeninde inceleyelim;

$$S(z) = S(z) \sum_{k=1}^p a_k z^{-k} + GU(z) \quad (2.8)$$

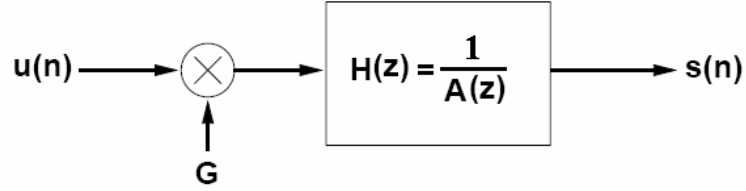
$$S(z) \cdot \left[1 - \sum_{k=1}^p a_k z^{-k} \right] = GU(z) \quad (2.9)$$

Bu sistem için transfer fonksiyonunu belirleyelim;

$$H(z) \triangleq \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{k=1}^p a_k z^{-k}} = \frac{1}{A(z)} \quad (2.10)$$

$$A(z) = 1 - \sum_{k=1}^p a_k z^{-k} \quad (2.11)$$

Şekil 2.9 'dan da görüleceği üzere, G ile ölçeklendirilen beyaz gürültü (uyarım işareti) transfer fonksiyonu $H(z)$ olan tüm kutup bir sisteme giriş olarak uygulandığında $s(n)$ işareti teşkil edilebilmektedir.



Şekil 2.9 Doğrusal öngörü sentez filtresi

Şimdi aynı sistemi doğrusal öngörü mantığı ile inceleyelim; $\hat{s}(n)$ 'yi geçmiş p örneğinin doğrusal kombinasyonu olarak tanımlayalım;

$$\hat{s}(n) = \sum_{k=1}^p a_k s(n-k) \quad (2.12)$$

Bu durumda $\hat{s}(n)$ ile $s(n)$ arasında bir fark (öngörü hatası) olacaktır. Bu fark (öngörü hatası) ;

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^p a_k s(n-k) \quad (2.13)$$

şeklinde tanımlanır. Hata işaretini de z domeninde inceleyelim;

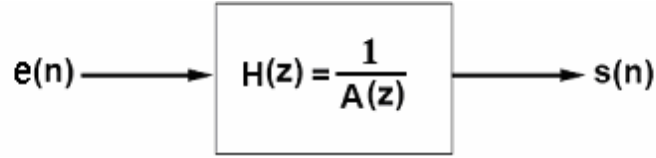
$$E(z) = S(z) \cdot \left[1 - \sum_{k=1}^p a_k z^{-k} \right] = S(z) A(z) \quad (2.14)$$

(2.14) ifadesini, hata işaretini giriş, $s(n)$ 'nin çıkış olduğu bir sistemin transfer fonksiyonunu gösterecek şekilde düzenleyelim;

$$H(z) = \frac{S(z)}{E(z)} = \frac{1}{1 - \sum_{k=1}^p a_k z^{-k}} = \frac{1}{A(z)} \quad (2.15)$$

$$A(z) = 1 - \sum_{k=1}^p a_k z^{-k} \quad (2.16)$$

Şekil 2.10'dan da görüleceği üzere, öngörü hatasının giriş olduğu, transfer fonksiyonu $H(z)$ olan tüm kutup sistem ile $s(n)$ işareti, AR yaklaşımda olduğu gibi, teşkil edilebilmektedir. *Bu açıklananlar ile, öngörü hatası $e(n)$ 'nin, $G \cdot u(n)$ ölçeklenmiş beyaz gürültüye eşit olduğu gösterilmiş oldu.*



Şekil 2.10 Doğrusal öngörü sentez filtresi

İster ses kodlamada olsun, ister ses özelliklerini ortaya çıkarmakta kullanılsın, LPC ile asıl amaçlanan ses işaretinin daha az sayısal değerle, a_k katsayılarıyla, temsil edebilmektir. İncelenen konuşma işaretine ilişkin a_k katsayıları ve gerekli uyarım işareti elde olduğu süreç, modelin elverdiği ölçüde, işaret yeniden teşkil edilebilir. Daha önceleri de açıklandığı gibi, kullanılan tüm doğrusal öngörü (LP) yöntemlerde değişen genelde uyarım sinyaliydi. Örneğin RELP (Residual Excited Linear Predictor) yönteminde hata işareti (rezidü, artık-kalan) kayıplı da olsa kodlanarak sentez kısmından analiz kısmına gönderilerek, yalın LPC (kastımız LPC-10) yönteminde göre daha iyi ses kalitesi, tabii ki artan bit oranlarına pahasına, elde edilebilmektedir. Veya CELP (Code-Excited Linear Predictor) yönteminde olduğu gibi, sesin perdesinin tespitinde işaretin kendisi yerine, hata işareti incelenebilir. Hatırlanmalıdır ki, modelimize göre, sentezde seslendirilen olarak sınıflandırılan seslerde zaten darbe katarı kullanıldığından ve hata işareti de zaten uygulanması gereken ideal uyarım işaretine eşit olduğundan, yalnızca ses perdesi belirlenmesiyle, hata işareti bir şekilde sentezciye gönderilmese dahi, oluşturulacak yaklaşık uyarım işareti ile ses işareti yeniden teşkil edilebilir.

a_k katsayıları tespit edilirken $\hat{s}(n)$ 'yi mümkün olduğunca $s(n)$ 'ye yakın tutabilmek için hata minimize edilir. a_k katsayıların olasılık dağılımı bilinmediğinden ortalama karesel hata (MSE – Mean Square Error) tanımlanır ve bu hatayı en küçük yapacak a_k katsayıları tespit edilir.

Bundan sonraki kısımda gerek ortalama karesel hata, gerekse özilişki ifadelerinin başında yer alan katsayı gösterilmemiştir. Ortalama karesel hata ifadesi sıfıra, özilişki ifadeleri birbirlerine eşitlendiğinden bu katsayıların gösterimine gerek kalmamıştır.

Konuşma işaretini çerçeveler halinde işlediğimizden, her çerçeve için LPC parametreleri ayrı ayrı hesaplanacaktır. $s_n(m)$, n anında başlayan çerçevenin m . örneğini göstermek üzere aşağıdaki tanımlamalar yapılabilir;

$$s_n(m) = s(m+n) \quad \text{ve} \quad e_n(m) = e(m+n) \quad (2.17)$$

Her çerçeve için ortalama karesel hatayı tanımlayalım;

$$E_n = \sum_m e_n^2(m) = \sum_m [s_n(m) - \hat{s}_n(m)]^2 = \sum_m \left[s_n(m) - \sum_{k=1}^p a_k s_n(m-k) \right]^2 \quad (2.18)$$

Bu ifadenin her bir a_i katsayına göre türevi alınıp sıfıra eşitlenir ise; E_n 'yi minimize eden a_i katsayıları elde edilmiş olur.

$$\frac{\partial E_n}{\partial a_i} = 0 \quad , \quad i = 1, 2, \dots, p \quad (2.19)$$

$$\frac{\partial E_n}{\partial a_i} = -2 \cdot \sum_m \left[s_n(m) - \sum_{k=1}^p a_k s_n(m-k) \right] \cdot s_n(m-i) = 0 \quad (2.20)$$

$$\sum_m s_n(m-i) s_n(m) = \sum_{k=1}^p a_k \sum_m s_n(m-i) s_n(m-k) \quad (2.21)$$

Buradan p bilinmeyenli p denklem elde etmiş olduk. $s_n(m)$ işareti için (i,k) . elemanı aşağıdaki gibi verilen kısa zaman kovaryans matrisi tanımlanabilir ;

$$\phi_n(i, k) = \sum_m s_n(m-i) s_n(m-k) \quad (2.22)$$

Bu yeni tanımla elde ettiğimiz (2.21) ifadesini yeniden yazalım;

$$\phi_n(i, 0) = \sum_{k=1}^p a_k \phi_n(i, k) \quad , \quad i = 1, 2, \dots, p \quad (2.23)$$

Sonuç olarak elde ettiğimiz (2.23) bağıntısı **Wiener-Hopf** eşitliğinin genel halidir ve aynı zamanda Yule-Walker eşitliği veya normal denklem olarak da anılır. p bilinmeyenli p eşitlik kümesi tanımlar.

Ortalama karesel hatanın kendisi ve türevinin sıfıra eşitlenmesi ile elde edilen (hatayı minimize eden katsayılar için) bağıntılar ile gösterilebilir ki toplam minimum hatanın bir sabit bileşeni, bir de doğrusal öngörü katsayılarının fonksiyonu olan bir bileşeni vardır;

$$E_n = \sum_m e_n^2(m) = \sum_m [s_n(m) - \hat{s}_n(m)]^2 = \sum_m \left[s_n(m) - \sum_{k=1}^p a_k s_n(m-k) \right]^2 \quad (2.24)$$

$$E_n = \sum_m s_n^2(m) - 2 \sum_m \sum_{k=1}^p a_k s_n(m) s_n(m-k) + \sum_m \sum_{k=1}^p \sum_{l=1}^p a_k a_l s_n(m-k) s_n(m-l) \quad (2.25)$$

$$E_n = \sum_m s_n^2(m) - 2 \sum_{k=1}^p a_k \sum_m s_n(m) s_n(m-k) + \sum_{l=1}^p a_l \left[\sum_{k=1}^p a_k \sum_m s_n(m-k) s_n(m-l) \right] \quad (2.26)$$

$$E_n = \sum_m s_n^2(m) - 2 \sum_{k=1}^p a_k \sum_m s_n(m) s_n(m-k) + \sum_{l=1}^p a_l \sum_m s_n(m) s_n(m-l) \quad (2.26)$$

$$E_n = \sum_m s_n^2(m) - \sum_{k=1}^p a_k \sum_m s_n(m) s_n(m-k) \quad (2.27)$$

$$E_n = \phi_n(0,0) - \sum_{k=1}^p a_k \phi_n(0,k) \quad (2.28)$$

Optimum öngörü katsayılarını bulmak için, öncelikle (2.23)'teki Yule-Walker eşitliğindeki $\phi_n(i,k)$ değerleri $1 \leq i \leq p$ ve $0 \leq k \leq p$ için hesaplanmalıdır. Bundan sonra yapılması gereken sadece p bilinmeyenli p eşitliğin çözülmesidir.

Görüldüğü üzere $\phi_n(i,k)$ için verilen ifadede toplam ifadelerinde n için sınırlar belirtilmemiştir. Doğrusal öngörü analizinde kullanılan iki yaklaşım (özilişki ve kovaryans) bundan sonraki bahislerde tanıtıldığında bu husus açıklık kazanacaktır.

2.5.1 Özilişki Metodu

Yukarıda irdelenen karesel öngörü hata ifadesindeki (2.18) toplamın sınırlarının belirlenmesinde kullanılan bu yaklaşımda, incelenen çerçeve ($0 \leq m \leq N-1$) dışında işaretin ($s_n(m)$ 'nin) tamamı ile sıfır olduğu kabulü vardır. $w(m)$ sonlu uzunluklu ve $0 \leq m \leq N-1$ dışında sıfır olan bir çerçeve fonksiyonu olmak üzere (örneğin Hamming, Blackman...) $s_n(m)$ işareti şu şekilde tanımlanabilir;

$$s_n(m) = s(m+n) \cdot w(m) \quad (2.29)$$

Her çerçeve için öngörü hatasını ifadesi incelenir ise bu yaklaşımdaki kabulün etkisi anlaşılır;

$$e_n(m) = s_n(m) - \hat{s}_n(m) = s_n(m) - \sum_{k=1}^p a_k s_n(m-k) \quad (2.30)$$

(2.30) ifadesinden de görüldüğü üzere $0 \leq m \leq N-I+p$ aralığında öngörü hatası sıfırdan farklı değerler alabilmektedir. Bu durumda öngörü hatasını yeniden ifade edelim;

$$E_n = \sum_{m=0}^{N+p-1} e_n^2(m) \quad (2.31)$$

Bu yaklaşımda incelenen çerçeve dışında işaretin tamamının sıfır olması kabulü vardı. Öngörü hatası ifadesi dikkatle incelenir ise, hatanın çerçevenin başında ve sonunda daha büyük olacağı anlaşılabilmektedir. Çünkü başlangıçta sıfır olan bir değerlerden, sıfır olmayan bir değer, sonda ise sıfır olmayan bir değerlerden, sıfır öngörmeye çalışılmaktadır. Bu nedenle incelenen çerçevelerdeki sıfır geçişleri yumuşatmak için işaret pencerelenir.

Şimdi daha önce toplam işleminde sınırları verilmeden tanımlanan kısa zaman kovaryans fonksiyonunu (2.22) yeniden yazalım;

$$\phi_n(i, k) = \sum_{m=0}^{N+p-1} s_n(m-i) s_n(m-k) \quad , \quad 1 \leq i \leq p \quad , \quad 0 \leq k \leq p \quad (2.32)$$

$s_n(m)$ ifadesinin $0 \leq m \leq N-I$ aralığı dışında sıfır olduğu kabulü ile yukarıdaki ifadeyi yeniden düzenleyelim;

$$\phi_n(i, k) = \sum_{m=0}^{N+p-1} s_n(m-i) s_n(m-k) = \sum_{m=-i}^{N+p-1-i} s_n(m) s_n(m+i-k) \quad (2.33)$$

$$\phi_n(i, k) = \sum_{m=0}^{N-1-(i-k)} s_n(m) s_n(m+i-k) \quad , \quad 1 \leq i \leq p \quad , \quad 0 \leq k \leq p \quad (2.34)$$

Şimdi $s_n(m)$ ifadesi için kısa zamanlı özilişki fonksiyonunu tanımlayalım :

$$R_n(k) = \sum_{m=-\infty}^{\Delta} s_n(m) s_n(m+k) = \sum_{m=0}^{N-1-k} s_n(m) s_n(m+k) \quad (2.35)$$

Görüldüğü üzere (2.34) ifadesi $R_n(i-k)$ ile aynıdır.

$$\phi_n(i, k) = R_n(i-k) \quad (2.36)$$

$R_n(i-k)$ çift bir fonksiyon olduğundan ifade şu şekilde düzenlenebilir ;

$$\phi_n(i, k) = R_n(|i-k|) \quad i = 1, 2, \dots, p \quad , \quad k = 0, 1, 2, \dots, p \quad (2.37)$$

Şimdi (2.37) ifadesini önceden elde ettiğimiz Yule-Walker (2.23) eşitliğinde ikame edelim;

$$\phi_n(i,0) = \sum_{k=1}^p a_k \phi_n(i,k) \quad , \quad i=1,2,\dots,p \quad \rightarrow$$

$$R_n(i) = \sum_{k=1}^p a_k R_n(|i-k|) \quad , \quad i=1,2,\dots,p \quad (2.38)$$

Elde ettiğimiz $R_n(i)$ ifadesi (2.38) aşağıdaki gibi matris şeklinde yeniden yazılabilir. Burada elde edilen $p \times p$ boyutlu özilişki matrisi Teoplitz özelliği göstermektedir. (Tüm diyagonal elemanları eşittir.) Bu *Teoplitz* özelliği sayesinde *Levinson-Durbin* gibi yöntemler kullanılarak denklem çözülerek a_k katsayıları kolaylıkla elde edilebilir. Bu konu ile ayrıntılara ileriki bahislerde yer verilecektir.

$$\begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \dots & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & \dots & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & \dots & R_n(p-3) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & \dots & R_n(0) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ \dots \\ a_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ \dots \\ \dots \\ R_n(p) \end{bmatrix} \quad (2.39)$$

Daha önce belirtildiği üzere ortalama karesel hatanın kendisi ve türevinin sıfıra eşitlenmesi ile elde edilen (hatayı minimize eden katsayılar için) bağıntılar ile toplam minimum hatanın bir sabit bileşeni ve doğrusal öngörü katsayılarının fonksiyonu olan bir bileşeni olduğu gösteren (2.28) ifadesinde de (2.37) eşitliği ikame edilebilir ;

$$E_n = \phi_n(0,0) - \sum_{k=1}^p a_k \phi_n(0,k) \quad \rightarrow$$

$$E_n = R_n(0) - \sum_{k=1}^p a_k R_n(k) \quad (2.40)$$

2.5.2 Kovaryans Metodu

Yukarıda irdelenen karesel öngörü hata ifadesindeki (2.18) toplamın sınırlarının belirlenmesinde kullanılan yaklaşımda, çerçeve $0 \leq m \leq N-1$ için incelenir ve çerçevenin dışındaki değerlere de erişilir. Her çerçeve için öngörü hatasını ifadesi incelenir ise bu yaklaşımdaki kabulün etkisi anlaşılır. Kabulümüze göre karesel öngörü hatayı yeniden tanımlayalım;

$$E_n = \sum_{m=0}^{N-1} e_n^2(m) \quad (2.41)$$

Şimdi daha önce toplam işleminde sınırları verilmeden tanımlanan kısa zaman kovaryans fonksiyonunu (2.22) yeniden yazalım;

$$\phi_n(i, k) = \sum_{m=0}^{N-1} s_n(m-i) s_n(m-k) \quad , \quad 1 \leq i \leq p \quad , \quad 0 \leq k \leq p \quad (2.42)$$

Bu ifadeyi farklı şekillerde düzenleyerek irdeleyelim;

$$\phi_n(i, k) = \sum_{m=-i}^{N-i-1} s_n(m) s_n(m+i-k) \quad , \quad 1 \leq i \leq p \quad , \quad 0 \leq k \leq p \quad (2.43)$$

$$\phi_n(i, k) = \sum_{m=-k}^{N-k-1} s_n(m) s_n(m+k-i) \quad , \quad 1 \leq i \leq p \quad , \quad 0 \leq k \leq p \quad (2.44)$$

Yukarıdaki ifadelerden de görüleceği üzere $0 \leq m \leq N-1$ için $\phi_n(i, k)$ hesaplanırken $s_n(m)$ çerçevesinin dışındaki değerlere de $(-p \leq m \leq N-1)$ erişilir. Bu durumda $m=0$ ve $m=N-1$ için öngörü hatasının hesaplanmasında keskin geçişler olmadığından olmayacağından incelenen çerçevenin pencerelenmesine gerek kalmamıştır. Burada bir çerçeveden bahsedildiğine göre işaretin en azından bir dikdörtgen pencerelemeye tabi tutulduğu aşıkardır. Pencerelemeden kasıt işaretin uç noktalarını yumuşatacak özel (Hamming, Blackman...) fonksiyonlar ile çarpmadan ibarettir.

Şimdi daha önceki bahislerde elde ettiğimiz Yule-Walker eşitliğini (2.23) ve matris formunu yeniden yazalım;

$$\phi_n(i, 0) = \sum_{k=1}^p a_k \phi_n(i, k) \quad , \quad i = 1, 2, \dots, p$$

$$\begin{bmatrix} \phi_n(1,1) & \phi_n(1,2) & \phi_n(1,3) & \dots & \phi_n(1,p) \\ \phi_n(2,1) & \phi_n(2,2) & \phi_n(2,3) & \dots & \phi_n(2,p) \\ \phi_n(3,1) & \phi_n(3,2) & \phi_n(3,3) & \dots & \phi_n(3,p) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \phi_n(p,1) & \phi_n(p,2) & \phi_n(p,3) & \dots & \phi_n(p,p) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ \dots \\ a_p \end{bmatrix} = \begin{bmatrix} \phi_n(1,0) \\ \phi_n(2,0) \\ \phi_n(3,0) \\ \dots \\ \dots \\ \phi_n(p,0) \end{bmatrix} \quad (2.45)$$

Korelasyon yaklaşımı için yukarıda elde ettiğimiz $\phi_n(i, k)$ ifadelerden görüldüğü üzere $\phi_n(i, k) = \phi_n(k, i)$ olduğunda $p \times p$ boyutlu matris simetriktir fakat Teopltitz özelliği göstermez. Ayrıca matrisin diyagonal elemanları arasında (2.46) ilişkisinin varlığı görülmektedir.

$$\phi_n(i+1, k+1) = \phi_n(i, k) + s_n(-i-1, -k-1) - s_n(N-1-i)s_n(N-1-k) \quad (2.46)$$

2.5.3 Doğrusal Öngörü Süzgeç Parametrelerinin Hesaplanması

Doğrusal öngörü analiz sistemini gerçeklemek için öncelikle elde edilen doğrusal eşitliklerin efektif bir yöntemle çözülmesi gerekmektedir. p bilinmeyenli p doğrusal eşitliği çözmek için kullanılan farklı etkinlikte pek çok yöntem mevcuttur. Bu çalışmada özilişki metodunu ve elde edilen eşitlikleri çözümünde de Durbin Yinelemesi yöntemini kullanacağız. Hatırlanacağı üzere öngörü katsayıları elde etmek için özilişki metodunda elde ettiğimiz ifade (2.38) aşağıdaki gibiydi ve ifadeyi matris formunda yazdığımızda elde ettiğimiz ifadedeki (2.39) katsayılar matrisi de Teopltitz özelliği göstermekteydi ;

$$R_n(i) = \sum_{k=1}^p a_k R_n(|i-k|) \quad , \quad i = 1, 2, \dots, p$$

Sistem eşitliklerini çözmek için, katsayılar matrisinin Teopltitz özelliğini kullanarak, birçok yinelemeli algoritma geliştirilmiştir. Bunların en çok bilinenleri *Levinson* ve *Robinson* algoritmaları olmasına karşın, eşitliklerin çözülmesinde kullanılan en efektif algoritma Durbin yinelemesidir.

$$\begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \dots & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & \dots & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & \dots & R_n(p-3) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & \dots & R_n(0) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ \dots \\ a_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ \dots \\ \dots \\ R_n(p) \end{bmatrix}$$

Levinson-Durbin algoritmasındaki amaç doğrusal öngörü filtresi katsayıları $a_1^{(i)}$, $a_2^{(i)}$, ..., $a_p^{(i)}$ ile ilgili öngörü hata değışintisinin yinelemeli olarak bulunmasıdır. Aşağıda Levinson-Durbin yinelemesinin adımları verilmiştir. (Bu aşamada gösterim kolaylığı nedeniyle n alt indisi kaldırılmıştır.)

- Algoritma ilklenir ;

$$E^{(0)} = R(0) \quad (2.47)$$

- PARCOR katsayıları hesaplanır ;

$$k_i = \frac{\left[R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j) \right]}{E^{(i-1)}}, \quad 1 \leq i \leq p \quad (2.48)$$

- i . adım için öngörü katsayıları hesaplanır ;

$$\begin{aligned} a_i^{(i)} &= k_i \\ a_j^{(i)} &= a_j^{(i-1)} - k_i \cdot a_{i-j}^{(i-1)}, \quad 1 \leq j \leq i-1 \end{aligned} \quad (2.49)$$

- Öngörü hatası güncellenir ;

$$E^{(i)} = (1 - k_i^2) \cdot E^{(i-1)} \quad (2.50)$$

- Yukarıdaki adımlar $i=1,2,\dots,p$ için çözülür ve sonuç aşağıdaki gibi elde edilir ;

$$a_j = a_j^{(p)} \quad (2.51)$$

Şimdi Levinson-Durbin algoritmasının anlaşılması amacıyla öngörü derecesi $p=2$ için doğrusal öngörü katsayılarının hesaplandığı bir örnek yapalım ;

$$\begin{bmatrix} R(0) & R(1) \\ R(1) & R(0) \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \end{bmatrix} \quad \text{olarak verildi ise ;}$$

$$E^{(0)} = R(0)$$

i = 1 için ;

$$k_1 = \frac{R(1)}{E^{(0)}} = \frac{R(1)}{R(0)}$$

$$a_1^{(1)} = k_1 = \frac{R(1)}{R(0)}$$

$$E^{(1)} = (1 - k_1^2) \cdot E^{(0)} = \left(1 - \left[\frac{R(1)}{R(0)} \right]^2 \right) \cdot R(0) = \frac{R^2(0) - R^2(1)}{R(0)}$$

i = 2 için ;

- $k_2 = \frac{[R(2) - a_1^{(1)}R(1)]}{E^{(1)}} = \frac{R(2)R(0) - R^2(1)}{R^2(0) - R^2(1)}$
- $a_2^{(2)} = k_2 = \frac{R(2)R(0) - R^2(1)}{R^2(0) - R^2(1)}$
- $a_1^{(2)} = a_1^{(1)} - k_2 \cdot a_1^{(1)} = \frac{R(1)R(0) - R(1)R(2)}{R^2(0) - R^2(1)}$

Sonuç olarak $a_1 = a_1^{(2)}$
 $a_2 = a_2^{(2)}$ elde edilir.

Levinson –Durbin algoritmasında hesaplanan $E^{(i)}$ parametresi, i . yineleme için öngörü hatasını göstermektedir. Böylece hesaplamının her aşamasında öngörü hatası gözlenebilir. Ayrıca özilişki katsayıları $R(i)$ 'ler normalize özilişki katsayıları ($r(k)=R(k)/R(0)$) ile ikame edilse dahi matrissel eşitliğin çözümü değişmez, yalnızca $E^{(i)}$ normalize hata olarak nitelendirilir. Eğer normalize öngörü hatası $V^{(i)}$ ile gösterilir edilir ise (2.52) şeklinde ifade edilebilir ;

$$V^{(i)} = \frac{E^{(i)}}{R(0)} = 1 - \sum_{k=1}^i a_k r(k) \quad , \quad 0 < V^{(i)} \leq 1 \quad \text{ve} \quad i \geq 0 \quad (2.52)$$

$i=p$ için normalize öngörü hatası aşağıdaki şekilde de ifade edilebilir ;

$$V^{(p)} = \prod_{i=1}^p (1 - k_i^2) \quad , \quad -1 \leq k_i \leq 1 \quad (2.53)$$

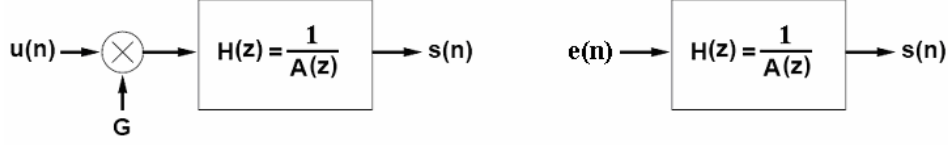
Buradaki önemli bir husus da elde ettiğimiz parametreler için süzgecin daima kararlılığının sağlanmasıdır. Daha önce elde ettiğimiz doğrusal öngörü sistemine ait transfer fonksiyonlarına (2.10 ve 2.15) bakılacak olursa $H(z)$ 'nin kararlı olması için $A(z)$ 'nin tüm köklerinin birim çember içerisinde olması gerektiği aşikardır.

$$H(z) \stackrel{\Delta}{=} \frac{1}{A(z)} \quad \Rightarrow \quad A(z) = 1 - \sum_{k=1}^p a_k z^{-k}$$

Kararlılık için gerek ve yeter koşul ; $|k_i| \leq 1 \quad , \quad \forall i \Leftrightarrow H(z)$

2.6 Öngörü Hata İşareti

Daha önceki bahislerde belirtildiği üzere modelimize göre öngörü hatası $e(n)$ 'nin, $G \cdot u(n)$ ölçeklenmiş beyaz gürültüye eşit olduğu gösterilmişti.



Şekil 2.11 Doğrusal öngörü sentez filtreleri

Aynı zamanda gösterilmiştir ki, uyarım işareti olarak öngörü hatası sisteme uygulandığında gerçekten modellenmek istenen ses işareti elde edilmektedir.

$$e(n) = s(n) - \sum_{k=1}^p a_k s(n-k) = G \cdot u(n) \quad (2.54)$$

Bu bilgiler ışığında, öngörü hatasının seslendirilen işaretler için sesin perdesinin (pitch) belirlenmesinde kullanılabileceği yorumu yapılabilir. Eğer ses işaretinde periyodiklik var ise öngörü hatasında da periyodik yükselmeler olacaktır. Yani seslendirilen işaretler için, öngörü hatasında belli eşik seviyesini aşan periyodik noktalar var ise sesin perdesi belirlenebilir. Veya öngörü hatasının özilişki analizinden yararlanılarak da ses perdesi tayini mümkündür. Uygulamalar hakkında bilgi verilirken öngörü hata işaretinin incelenmesi üzerine örnekler ile daha ayrıntılı bilgi verilecektir.

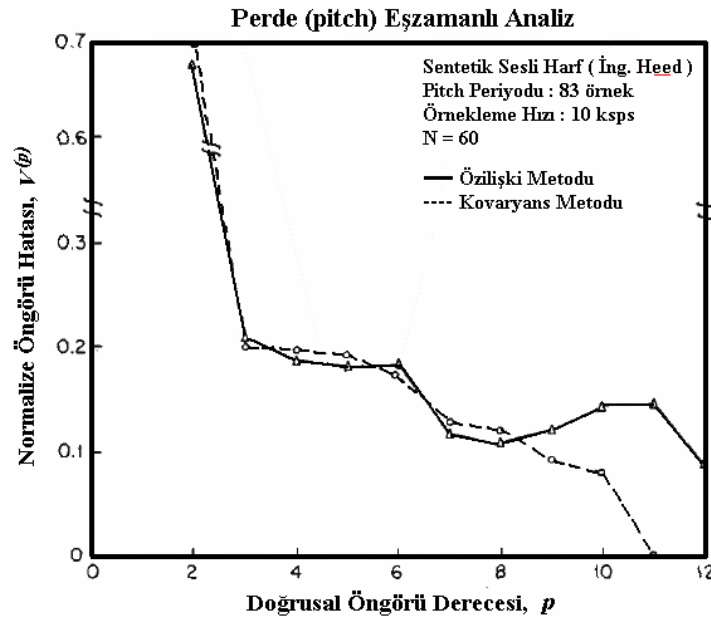
2.7 LPC Parametrelerinin Deneysel Olarak Değerlendirilmesi

Doğrusal öngörü analizinde kullanılacak olan p (öngörü derecesi) ve N (çerçeve boyutu) değerlerinin sağlıklı olarak seçilebilmesi için *Chandra* ve *Lin* (1974) bu değerlere göre normalize öngörü hatasını aşağıdaki kriterlere göre inceleyen bir çalışma yapmıştır;

- Kovaryans ve özilişki metodu
- Sentetik sesli harfler ve doğal konuşma
- Perde (pitch) eşzamanlı ve perde eşzamanlı olmayan (Eşzamanlılıktan kasıt pencere başlangıcı ile perde başlangıcının aynı nokta olarak seçilmesidir.)

Chandra ve *Lin* (1974) çalışmalarında, normalize öngörü hatası V için bizim daha önce tanımladığımız (2.52) ifadesinden farklı yöntemler kullanmışlardır. Burada

ilgilendiğimiz sonuçlar olduğundan alternatif yöntemler detaylandırılmamıştır.



Şekil 2.12 'heed' kelimesin seslendirilen kısmı için $V^{(p)}$ 'nin p 'ye göre değişimi – I *

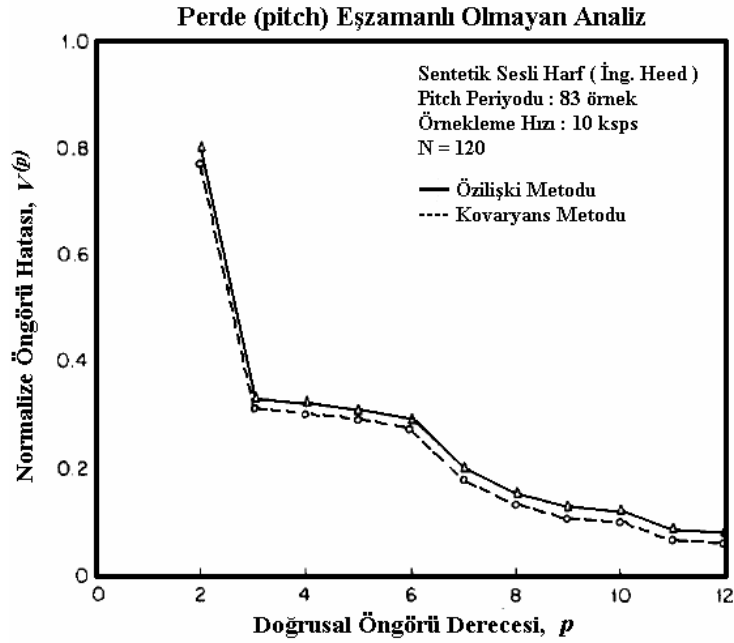
Şekil 2.12'de İngilizce 'heed' kelimesindeki seslendirilen kısım için normalize öngörü hatasının, öngörü derecesine göre değişiminin grafiği eşzamanlı analiz için verilmiştir. Görüleceği üzere, kovaryans metodu ile sentetik sesin üretilmesinde kullanılan sistemin derecesi olan $p=11$ 'e kadar monoton olarak azalmaktadır. Özilişki metodunda ise $p=7$ 'den sonra normalize öngörü hatası 0.1 gibi bir değerde kalmaktadır. Özilişki metodunda incelenen çerçeve ($0 \leq m \leq N-1$) dışında işaretin ($s_n(m)$ 'nin) tamamı ile sıfır olduğu kabulü olduğundan, p ne kadar artsa da $N=60$ gibi küçük bir çerçeveden dolayı hata azalmamaktadır.

Şekil 2.13'te İngilizce 'heed' kelimesindeki seslendirilen kısım için normalize öngörü hatasının, öngörü derecesine göre değişiminin grafiği eşzamanlı olmayan analiz için verilmiştir. Yalnız burada çerçeve boyutu $N=120$ olarak seçilmiştir. Görüleceği üzere, her iki metotla da $p=11$ için V normalize öngörü hatası 0.1'e kadar düşmektedir.

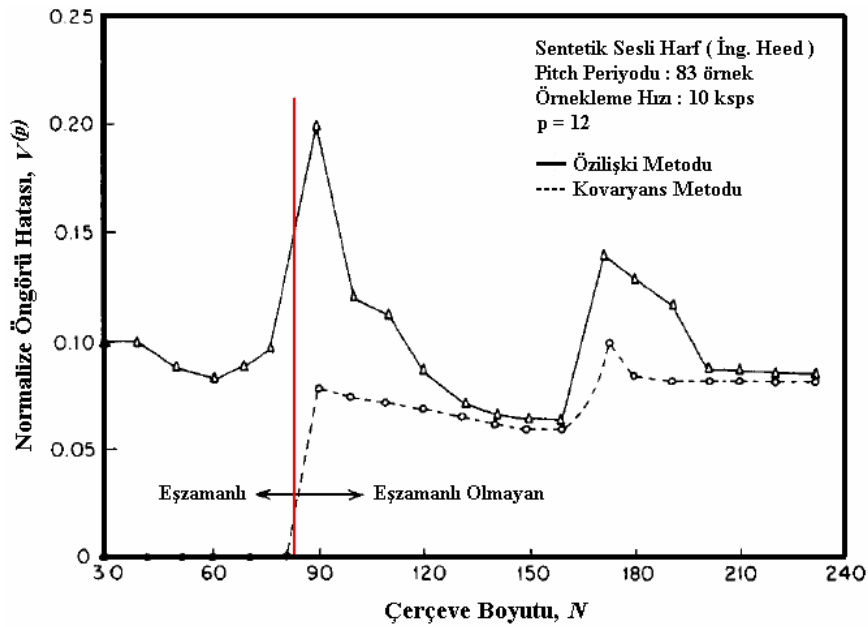
Şekil 2.14'te ise normalize öngörü hatasının, çerçeve boyutuna göre değişimi, öngörü derecesi $p=12$ için verilmiştir. Görüleceği üzere çerçeve boyutu N , perde (pitch) periyodu 83'ten küçük değerler için kovaryans metodu ile normalize öngörü hatası çok daha küçüktür. Bunu sebebi kovaryans metodunda incelenen çerçeve dışında işaretin sıfır olduğu kabul edilmemesinden dolayı çerçeve sınırlarındaki hesaplamalarda çerçeve dışındaki değerlerin

* Rabiner, L.R., Schafer R.W., (1978), Digital Processing of Speech Signals, Prentice Hall, Sayfa 427

kullanılabilir. Daha önce de belirtildiği üzere işaret seslendirilen olduğundan ve sistem perde (pitch) periyodundaki darbeler ile uyarıldığından perde periyodu ve katlarında hata artmaktadır.



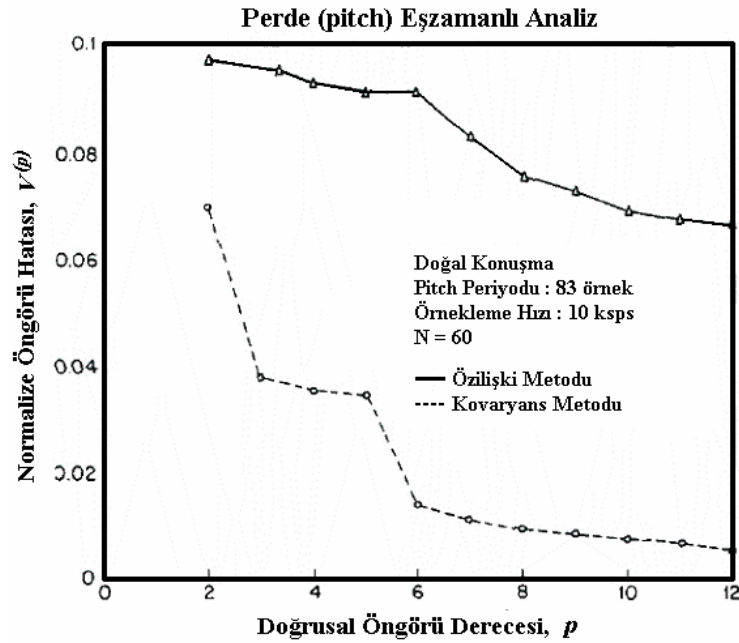
Şekil 2.13 ‘heed’ kelimesin seslendirilen kısmı için $V^{(p)}$ ’nin p ’ye göre değişimi – II *



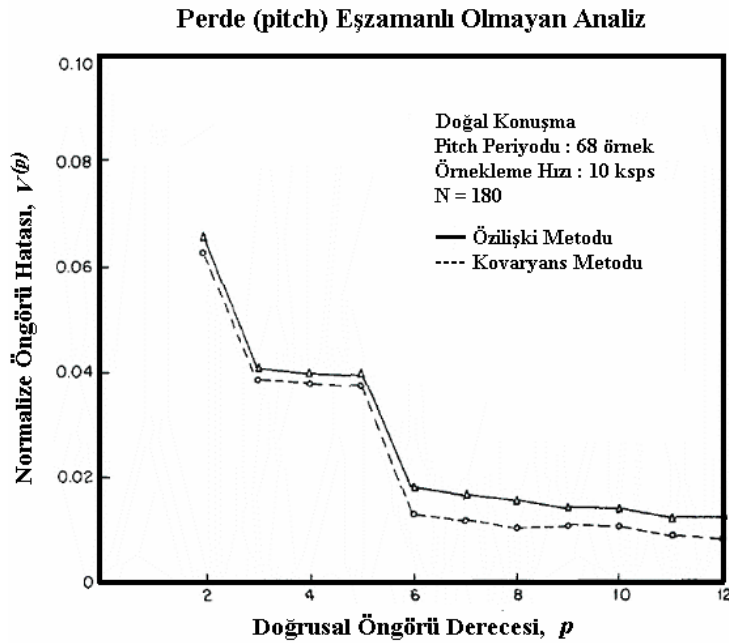
Şekil 2.14 ‘heed’ kelimesin seslendirilen kısmı için $V^{(p)}$ ’nin N ’ye göre değişimi **

* ve **, Rabiner, L.R., Schafer R.W., (1978), Digital Processing of Speech Signals, Prentice Hall, Sayfa 428

Şekil 2.15’de ise aynı denemeler seslendirilen doğal konuşma işareti için tekrarlanmıştır. Görüleceği üzere doğal konuşma işareti için kovaryans ve özilişki metotları için elde edilen normalize öngörü hatası daha da farklılaşmaktadır.



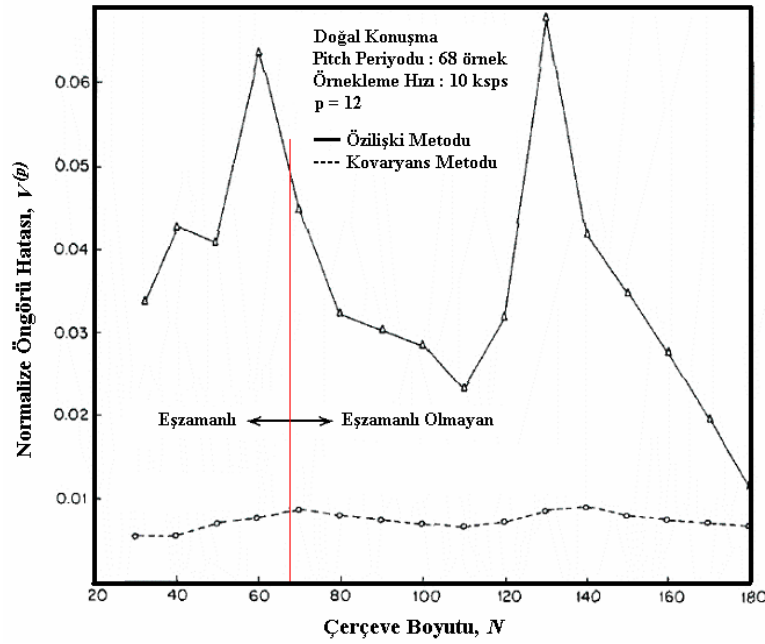
Şekil 2.15 Doğal konuşmada seslendirilen kısım için $V^{(p)}$ ’nin p ’ye göre değişimi – I *



Şekil 2.16 Doğal konuşmada seslendirilen kısım için $V^{(p)}$ ’nin p ’ye göre değişimi – II **

* ve **, Rabiner, L.R., Schafer R.W., (1978), Digital Processing of Speech Signals, Prentice Hall, Sayfa 429-430

Şekil 2.16’da da çerçeve boyutu arttıkça kovaryans ve özilişki metodu arasındaki farkın azaldığı görülmektedir. Şekil 2.17’de öngörü derecesi $p=12$ için çerçeve boyutu N ’ye göre, normalize öngörü hatasının değişimini gösteren grafikten de anlaşılabacağı üzere, doğrusal öngörü sistemine uygulana giriş darbeleri anında özilişki metodunda yüksek artışlar olmaktadır. Aynı zamanda artan çerçeve boyutu ile iki yöntemin sonuçlarının yaklaşımaya başladığı da gözlenebilir.



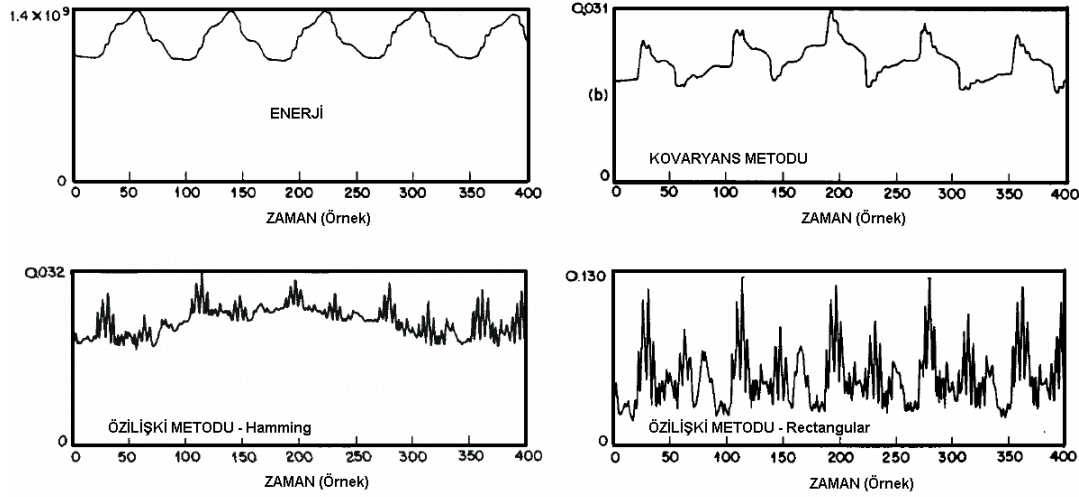
Şekil 2.17 Doğal konuşmada seslendirilen kısmı için $V^{(p)}$ 'nin N 'ye göre değişimi *

2.8 Normalize Öngörü Hatasını Çerçeve Pozisyonu İle Değişimi

Bir önceki bahiste normalize öngörü hatasının çerçeve boyutu ile ve öngörü derecesi (bir anlamda sistemin kutup sayısı) ile değişimi incelendi. Normalize öngörü hatasını etkileyen bir diğer faktör de seçilen çerçevenin pozisyonudur. Bu etki Şekil 2.18 'de gösterilmiştir. Grafikte analiz 10 Ksps 'de örneklenen 40ms'lik İngilizce /i/ sesi için yapılmıştır. İlk işaret sinyalin enerjisini göstermektedir. İkinci şekil ise öngörü derecesi $p=14$ ve 20ms'lik çerçeve ($N=200$) boyutu ile kovaryans metodu için normalize karesel öngörü hatasını göstermektedir. Üçüncü şekil Hamming penceresi kullanılan özilişki metodu için normalize karesel öngörü hatası verilmiştir. Dördüncü şekilde ise dikdörtgen pencere kullanılan özilişki metodu için normalize karesel öngörü hatası verilmiştir.

* Rabiner, L.R., Schafer R.W., (1978), Digital Processing of Speech Signals, Prentice Hall, Sayfa 431

Konuşma işareti için ortalama perde periyodu 84 örnek (8.4ms) olup 20ms'lik çerçeveler için bir çerçevede yaklaşık 2.5 periyot tamamlanmaktadır. Kovaryans metodunda çerçeve pozisyonunu değişimi ile normalize öngörü hatası belirgin değişiklikler gözlenmektedir. Bu değişimin esas nedeni öngörü hatasının perde periyodu başlarındaki büyük sapmalarındandır. Örnekten de görüleceği üzere üç perdeyi kapsayan çerçeve için normalize hata iki perdeyi kapsayan çerçeve için olan normalize hatadan büyüktür.



Şekil 2.18 Üç ayrı sistem için normalize öngörü hatası, V^*

Özilişki metodunda ise normalize öngörü hatası hem yüksek frekanslı hem de perde eşzamanlı düşük frekanslı bileşenler içermektedir. Yüksek frekanslı bileşen ilk p örnek için işaretin tam olarak öngörülememesindendir. Nedeni kolayca anlaşılacağı üzere Hamming penceresi kullanıldığında bu yüksek frekanslı bileşenin genliği küçülmektedir. (Çerçevenin başında ve sonunda işaret yumuşatılarak sıfır – sinyal geçişi yumuşatıldığından...) Düşük frekanslı bileşenin nedeni ise kovaryans metodundaki ile aynıdır.

Çerçeve pozisyonu ile değişim çoğu seslendirilen ses işareti için varolacaktır. Bu değişimin etkisini azaltmak için tüm-geçiren süzgeç ve pre-emphasis süzgeç uygulanabilir. Bu konu daha sonra detaylandırılacaktır.

2.9 Doğrusal Öngörü Analizin Frekans Domeni Yorumu

Bu noktaya kadar doğrusal öngörü analizi fark denklemleri ve ilişki fonksiyonları bakımından, zaman domeninde, incelendi. Yalnızca konunu başında doğrusal öngörü

* Rabiner, L.R., Schafer R.W., (1978), Digital Processing of Speech Signals, Prentice Hall, Sayfa 432

katsayılarının insanın ses sitemini modelleyen sistemin fonksiyonunun paydasında yer aldığını göstermiştik. Bu durumda elde edilen modelin frekans yanıtı kolaylıkla bulunabilir. $z = e^{j\omega}$ için $H(z)$ yeniden yazılır ise ;

$$H(e^{j\omega}) = \frac{G}{1 - \sum_{k=1}^p a_k e^{-j\omega k}} = \frac{G}{A(e^{j\omega})} \quad (2.55)$$

$H(e^{j\omega})$ frekansın fonksiyonu olarak çizilir ise incelenen çerçevedeki formant frekanslarda tepeler gözlenecektir. Yani doğrusal öngörü analizi kısa zamanlı spektrum hesabı olarak değerlendirilebilir. Uygulamalara kısmında bu konu analiz ve sentezde elde edilen grafikler ile detaylandırılarak incelenecektir.

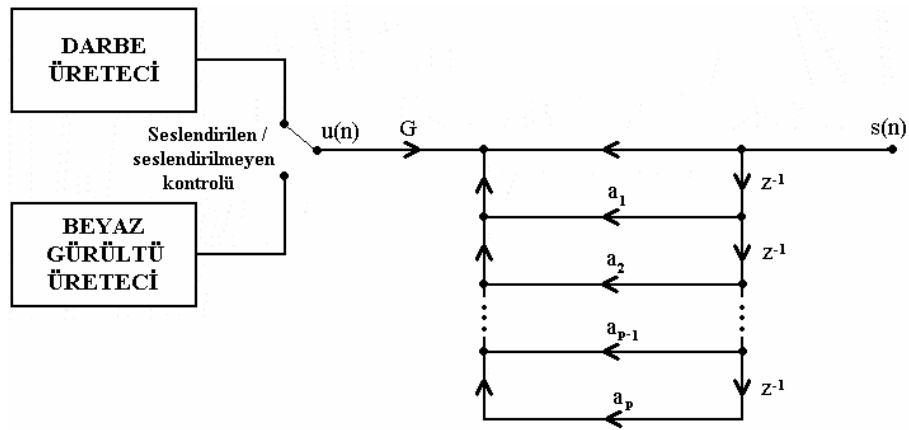
2.10 Doğrusal Öngörü Parametreleri İle Konuşma İşareti Sentezi

Konunun ilk başlangıcında doğrusal öngörü modelinin tanıtımında insan ses sistemi üzerinden yola çıkılarak bina edilen modele ait blok diyagramlar verilmişti. Burada biraz daha detaylandırılacaktır. Sentez için zamanla değişen perde periyodu, seslendirilen veya seslendirilmeyen bilgisi, kazanç ve doğrusal öngörü süzgeci katsayılar polinomu gereklidir. Seslendirilen işaretler için uyarım işareti olarak her bir perdenin başlangıcında birim genlikli darbe üretici kullanılır. Seslendirilmeyen işaretler için ise uyarım işareti olarak, ilişkisiz, üniform dağılımlı, standart sapması bir, ortalaması sıfır olan beyaz gürültü üretici kullanılır. G ile sistemin genel kazancı kontrol edilir. Sentez sonucu konuşma işareti örnekleri aşağıdaki bağıntı ile belirlenir;

$$s(n) = \sum_{k=1}^p a_k s(n-k) + Gu(n)$$

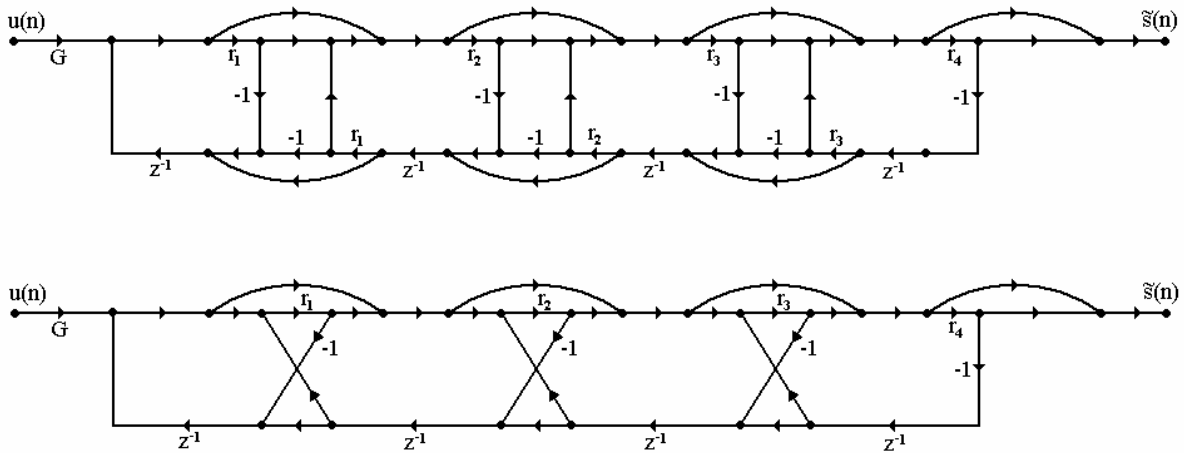
Şekil 2.19’da bu sisteme ait blok diyagram verilmiştir. Verilen blok diyagrama göre p adet toplama ve p adet çarpma gerekmektedir. Bu yapı kullanılabilecek en basit yapıdır. Şekil 2.19’da blok diyagramı verilen sistemin parametreleri zamanla güncellenmelidir. Konuşma işaretinin seslendirilen bölgelerinde parametreler düzenli aralıklarda hesaplanmasına karşın, kontrol parametreleri her periyodun başında güncellenir. Seslendirilmeyen bölgelerde ise genellikle her çerçevede bir kez parametreler güncellenir. Kontrol parametrelerin her periyodun başında güncellendiği *perde eşzamanlı sentez*, parametrelerin her çerçevede bir kez güncellendiği *perde eşzamanlı olmayan sentez* yöntemine göre daha efektifir. Bu durumda,

perde eşzamanlı sentezde, belirli aralıklarda elde edilen parametreler güncelleme esnasında interpolasyon yapılmalıdır. Bu konuda çalışmalar yapan *Atal* (1971) göstermiştir ki, kazanç ve perde interpolasyonu geometrik olarak (logaritmik skalada lineer olarak) yapılmalıdır. Fakat kararlılık kısıtlamasından dolayı doğrusal öngörü katsayıları arasında interpolasyon yapılamaz. Çünkü kararlı iki set doğrusal öngörü parametresi arasında yapılan interpolasyon sonucu kararsızlığa neden olabilecek bir set doğrusal öngörü katsayıları elde edilebilir. *Atal* (1971) bunun için bir çözüm geliştirmiştir fakat bu konuda şimdilik daha fazla irdelenemeyecektir.



Şekil 2.19 Doğrusal öngörü sentezleyicisinin blok diyagramı

Şekil 2.19’da verilen blok diyagram sadeliği ve uygulama kolaylığı nedeniyle LPC sistemlerinde en çok kullanılan yöntemdir. Dezavantajı ise yapılan işlemleri yinelemeli olduğundan katsayılardaki değişimlere yok duyarlı olmasıdır. Alternatif olarak öngörü parametreleri ile sentezde en etkili yöntem *kayıpsız tüp modelinde* yansıma katsayıları veya PARCOR katsayılarının kullanılmasıdır. Bu durumda blok diyagram aşağıdaki şekli alır;

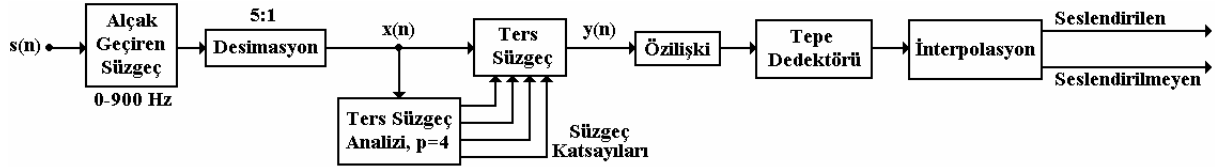


Şekil 2.20 Eşdeğer kayıpsız tüp modeli - tek ve iki çarpıcı birleşimli

Bu yapının avantajı yansıma katsayısı çarpanlar ($r_i = -k_i$) sınırlı olması ($|k_i| < 1$) ve kararlılık problemi olmaksızın iki set parametre arası interpolasyon yapılabilmesidir. Ayrıca bir önceki yapıya göre sayısal hassasiyete (kuantizasyona) daha az duyarlıdır. Uygulamamızda yer almayacağından bu husus detaylandırılmayacaktır.

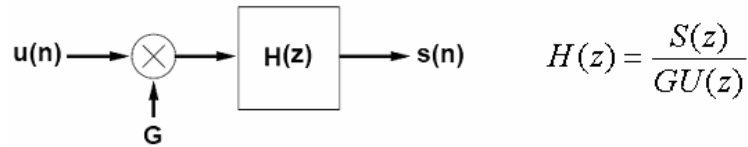
2.11 SIFT Algoritması

Öngörü hatası $e(n)$ ile ilgili daha önceki bahislerde öngörü hatasının pitch periyodunun belirlenmesinde kullanılabileceğinden bahsedilmişti. Bununla beraber seslendirilen işaretler için ses perdesi tayininde daha kompleks yöntemler de geliştirilmiştir. Bunlardan biri *Markel* (1972) tarafından (benzeri *Maksym* (1973) tarafından da geliştirilmiştir) geliştirilen **SIFT** (Simple Invers Filtering Tracking) metodudur. Şekil 2.21’de algoritmaya ilişkin blok diyagram verilmiştir



Şekil 2.21 SIFT Algoritmasına ilişkin blok diyagram

Giriş işareti önce 900Hz’lik alçak geçiren filtreden geçirilir. Daha sonra her beş örnekten biri alınarak örnekleme hızı beşte birine düşürülür (decimation). Elde edilen $x(n)$ işareti özilişki metodu kullanılarak öngörü derecesi $p=4$ için doğrusal öngörü ile analiz edilir. Dördüncü derece filtre sinyalin spektrumunu modellemek için yeterlidir çünkü 0-1 kHz aralığında sadece bir veya iki formant bulunmaktadır. Bir sonraki adımda ise $x(n)$ işareti tespit edilen parametreler ile teşkil edilen ters filtreden geçirilerek $y(n)$ elde edilir. Anlaşılabileceği üzere $y(n)$ yaklaşık düz bir spektruma sahip olacaktır. $u(n)$ uyarım işareti G de kazanç olmak üzere modelimiz için Şekil 2.22’deki gösterimi daha önce tanımlamıştık.

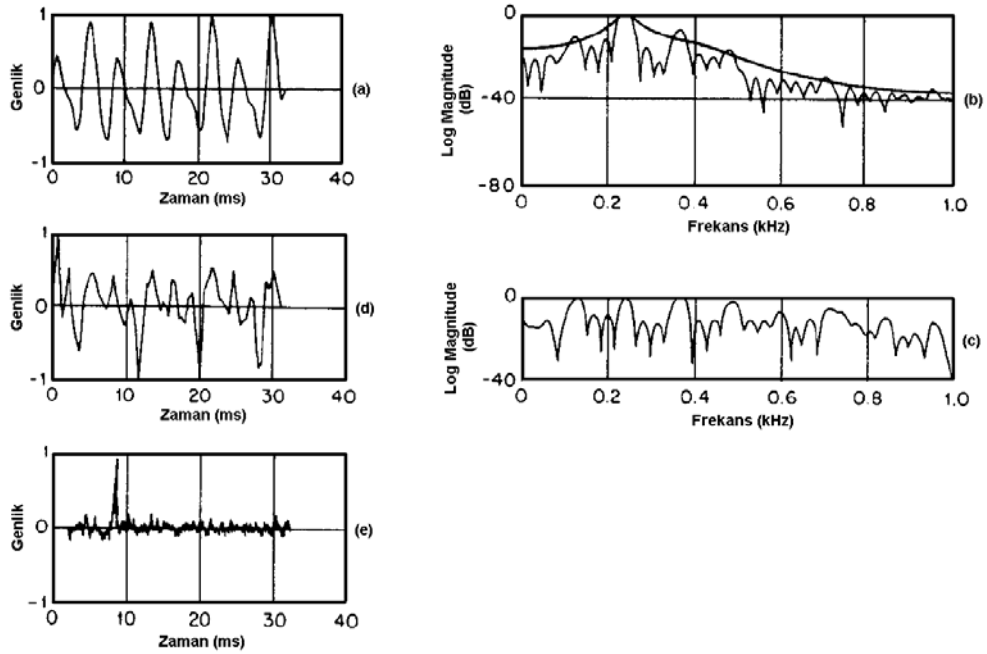


Şekil 2.22 Doğrusal öngörü sentez filtresi

$x(n)$ işaretinin yaklaşık olarak $s(n)$ olarak düşünülür ise $y(n)$ yaklaşık ölçeklenmiş uyarım işareti olduğu ve spektrumunun da ölçeklenmiş uyarım işaretine benzeyeceği

aşıkardır. Uyarım işareti ile öngörü hatası arasındaki ilişki de daha önceki bahislerde tanımlandığına göre $y(n)$ işaretinin dördüncü dereceden öngörücü için öngörü hatası olduğu anlaşılmaktadır.

Burada doğrusal öngörü kullanılmasının nedeni giriş işaretinin frekans spektrumunun düzleştirilmesidir. (yumuşatılması) Gerekli uyarım işaretini elde edebilmek için pitch tespiti gerekmektedir. Aslında $y(n)$ modelimizin uyarım şekline uymasa da kullanılabilir bir uyarım işaretidir. Ters filtreden geçirilen giriş işaretinin özilişki fonksiyonunun en yüksek değeri bulunarak pitch tespit edilmeye çalışılır. Daha hassas bir pitch değeri için elde edilen özilişki fonksiyonuna maksimum değeri civarında interpolasyon uygulanabilir. Elde edilen normalize özilişki fonksiyonu için bir eşik değeri belirlenerek seslendirilen / seslendirilmeyen ayrımı da yapılabilir.



Şekil 2.23 SIFT algoritmasındaki aşamalarındaki örnek sinyaller *

Şekil 2.23'te algoritmanın çeşitli aşamalarındaki sinyallere örnekler verilmiştir. (a)'da örnek bir giriş işareti gösterilmektedir. (b)'de giriş işaretinin ve ters süzgecin frekans spektrumu verilmiştir. Grafikten de görüleceği üzere yaklaşık 250 Hz civarında bir formant (rezonans) frekansı mevcuttur. (c) şeklinde ters süzgeç çıkışındaki, blok diyagrama göre $y(n)$, işaretin frekans spektrumu verilmiştir. (d)'de ise bu noktadaki işaretin zaman domenindeki

* Rabiner, L.R., Schafer R.W., (1978), Digital Processing of Speech Signals, Prentice Hall, Sayfa 448

dalga şekli gösterilmektedir. En son olarak (e) şeklinde de blok diyagramda gösterilen $y(n)$ işaretinin normalize edilmiş özilişki fonksiyonu verilmiştir. Yaklaşık 8ms'lik pitch (perde) periyodu grafikten kolaylıkla anlaşılabilmektedir. Her ne kadar SIFT algoritması pitch tayinindeki en iyi yöntemlerden bir olsa da, yüksek perdeden olan konuşma işaretleri için, örneğin çocuklar ve ses kanalı telefon hatları olduğunda, 0-900 Hz aralığında birden fazla formant frekansı olacağından, spektral yumuşatma-düzleştirme başarılı olamamaktadır. Bu durumda diğer konuşma işaretinin perdesinin tespitinde daha sonra yer verilecek olan diğer metotlara başvurulabilir.

Konuşma işaretinin seslendirilen kısmı için formant tespiti için doğrusal öngörü analiz metodunun kullanılmasının pek çok avantajının yanında bazı dezavantajları da vardır. Öngörü parametreleri ile formant frekansı hesaplamada iki yöntem izlenebilir. En temel yöntem öngörü polinomunun çarpanlara ayrılıp, elde edilen köklerin hangilerinin formant frekansı, hangilerinin spektrumu teşkil eden frekans bileşenleri olduğunun tespit etmeye çalışmaktır. İkinci alternatif ise sistemin frekans spektrumu elde edip, tepe izleme yöntemi ile formant frekansının tespitidir.

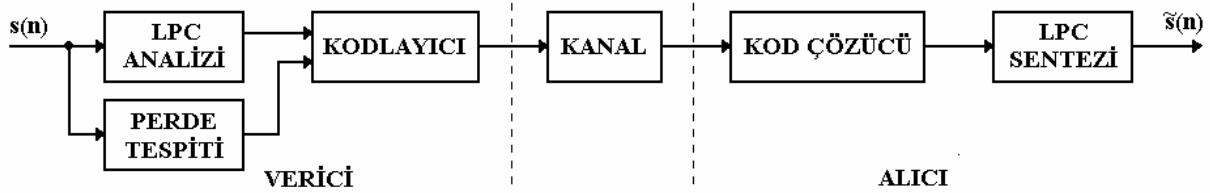
2.12 LPC Parametreleri Kullanılarak Formant Analizi

Doğrusal öngörü analizi ile formant analizinin en belirgin avantajı, formant merkez frekansının ve bant genişliğinin öngörü polinomunun çarpanlarına ayrılması ile belirlenebilmesidir. Öngörü derecesi p önceden belirlendiğinden olası maksimum kompleks eşlenik kutup sayısı $p/2$ olacaktır. Doğrusal öngörü analizi ile hangi kutupların formant frekansları oldukları diğer yöntemlere nazaran daha kolay tespit edilir, çünkü formant olmayan frekanslardaki kutupların bant genişlikleri konuşma işaretleri için beklenen bant genişliklerinden daha fazladır.

Doğrusal öngörü analizi ile formant analizinin dezavantajı ise konuşma işaretinin spektrumunun modellenmesinde tüm kutup modelin kullanılmasıdır. Model her ne kadar spektrum eşleme bakımından uygun olsa da, özellikle burun sesleri için, öngörü polinomunun köklerinin anlamı çok da açık değildir. Elde edilen kutupların burun seslerinden gelen kutuplara mı, yoksa vokal sistemin rezonans frekanslarına mı ait olduğunun tespiti pek kolay değildir. Diğer bir zorluk da, her ne kadar köklere ait bant genişliklerinin tespiti kolay olsa da, bu bant genişliklerinin formant frekansların bant genişlikleri ile bağıntıları tam olarak belirlenememektedir. Çünkü köklerin yerleri çerçeve boyutu, çerçeve yeri ve analiz metodu ile değişmektedir.

2.13 LPC Parametrelerinin Kuantizasyonu

Doğrusal öngörü ile konuşma işaretlerinin analiz ve sentezinin temel kullanım alanlarının başında düşük bit oranlarında veri iletimi ve saklanmasıdır. Temel olarak bir LPC ses kodlayıcı (voice coder – vocoder) sistemi doğrusal öngörü analizini / pitch tespiti yapıp uygun bir forma sokarak iletim kanalına ileten bir gönderici, iletim kanalı ve gelen veriyi uygun şekilde açıp konuşma işareti sentezi yapan bir alıcıdan müteşekkildir. Asıl konumuz doğrusal öngörü analiz ve sentezi olduğundan bu çalışmada iletim kanalının kayıp ve hataya neden olmayacağı kabul edilmiştir. Bu bölümde yalnızca kanalda iletilecek bit oranı üzerine açıklamalar yapılacaktır. Şekil 2.24’te tipik bir LPC ses kodlayıcı sistemine ait blok diyagram verilmiştir.



Şekil 2.24 Temel bir LPC ses kodlayıcı blok diyagramı

Kanaldan iletilmesi gereken temel parametreler ; p adet doğrusal öngörü katsayısı, pitch periyodu, seslendirilen/seslendirilmeyen bilgisi ve kazançtır. Uygulamada genel olarak sabit veya kayan noktalı olarak hesaplanan doğrusal öngörü katsayıları 6 bit olarak kuantalanır. Seslendirilen / seslendirilmeyen bilgisi 1 bit ve kazanç da logaritmik skalada 5 bit olarak kodlanır.

Daha önceki bahislerde kutup noktalarının yerlerinin değişmemesi ve kararsızlık durumunun ortaya çıkmaması için, öngörü katsayılarının sayısal hassasiyete olan duyarlılıklarından dolayı, yüksek sayısal çözünürlükte (8 veya 10) bit ile kodlama gerektiğinden bahsedilmişti. Bu nedenle doğrusal öngörü katsayıların doğrudan kuantizasyon yapılmasından kaçınılmalıdır. Bu nedenle kanaldan doğrudan doğrusal öngörü katsayıları yerine öngörü polinomunun köklerinin veya yansıma katsayılarının (PARCOR) iletilimi düşünülmüştür. Öngörü polinomunun kökleri kararlılık sıkıntısı olmadan kuantize edilebilir. Çünkü birim kökler birim çember içerisinde oldukça kararlılık garantilenmiş demektir. Atal ‘ın (1971) bu konu ilgili yaptığı çalışmaya göre öngörü polinomunun kökleri için 5 bitin ve bant genişliğini temsil etmek üzere 5 bitin kullanılması ile kuantalamadan kaynaklanan fark hissedilemez hale gelmektedir.

Bu tür bir kodlama yöntemi için iletim veya veri saklamada gereken bit oranı saniyede $72 \cdot F_s$ bit olarak ifade edilebilir. Burada F_s saniyedeki iletilen çerçeve sayısını temsil etmektedir. F_s için belirlenebilecek tipik değerler 100, 67 veya 33 sırasıyla 7200, 4800 veya 2400 bit/saniyelik hızlara tekabül etmektedir.

Yansıma katsayılarının iletildiği yaklaşımda da kararlılığın sağlanması garanti edilebilmektedir. Çünkü kararlılık için gereken $|k_i| < 1$ koşulu bozulmayacak şekilde kolaylık ile kuantalama yapılabilir. Makhoul ve Viswanathan (1975) tarafından yapılan çalışma sonucu göstermiştir ki, yansıma katsayılarının sabit bit sayısı ile optimum olarak iletilebilmesi için bir dönüşüme ihtiyaç duyulmaktadır. Uygulamamızda yer almayacağından bu dönüşümün ayrıntılarına girilmeyecektir, çalışmamızda yer alan kuantalama işlemleri ile ilgili ayrıntılar uygulamalar kısmında ayrıntıları ile verilecektir.

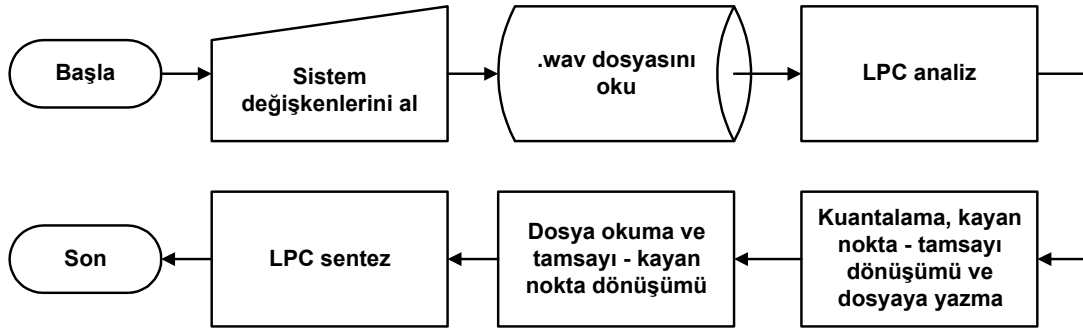
3. UYGULAMA

Bu tez çalışmasında elde edilmek istenen sonuç Mutlak Kategori Test'ine (ACR) göre orta ve üstünde ses kalitesine sahip, 2.4 kb/s mertebesinde sıkıştırma yapabilen bir ses kodlayıcının MATLAB programı üzerinde gerçekleştirilmesidir. Uygulama gerçek zamanlı olmayıp kayıtlı .wav uzantılı konuşma örnekleri dosyalarının doğrusal öngörü ile işlenerek, tanımladığımız .LPC uzantılı dosya formatına dönüştürülmesinden ve bazı sesli / görsel animasyonlardan müteşekkildir.

Bu çalışmada örnek konuşma işaretleri olarak UEKAE (Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü) MTRD (Multimedya Teknolojileri Değerlendirme, Araştırma ve Geliştirme) Laboratuvarı'nda hazırlanan TURTEL kütüphanesinin bir alt kümesi kullanılmıştır. Test kümemize telefon üzerinden kayıt edilen beş erkek ve beş kadın konuşmacının on beş ayrı cümlesi yer almaktadır. Kütüphanedeki cümleler anlamlı ifadelerdir, bununla beraber cümle ve kelime seçimlerindeki asıl gaye dilimizdeki tüm seslere ve yapılara mümkün mertebe yer verebilmektir.

Program çalıştırıldığında ilk olarak kullanıcıdan öngörü derecesi, çerçeve boyutu ve işlenecek .wav dosyasının adı gibi bilgiler istenmektedir. Animasyon kontrol , ön vurgulama kontrol, örnekleme frekansı, dosyadan okunacak azami veri süresi gibi değişkenler ise programın başında belirli ilk değerler ile, sonradan değiştirilebilecek şekilde tanımlanmıştır.

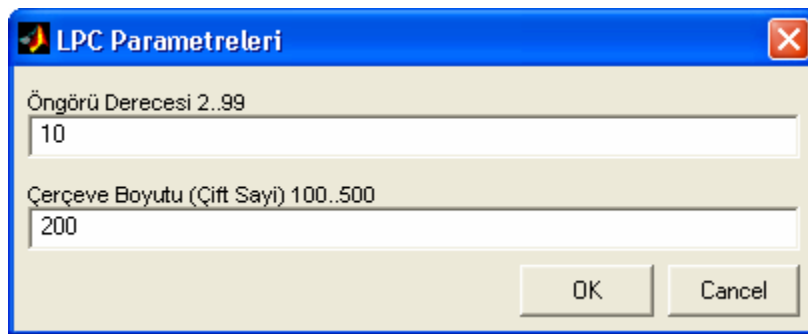
Şekil 3.1'de program akışı ile ilgili akış diyagramı verilmiştir. Bu bölümde her bir adım detaylandırılacaktır.



Şekil 3.1 MATLAB uygulaması genel alış diyagramı

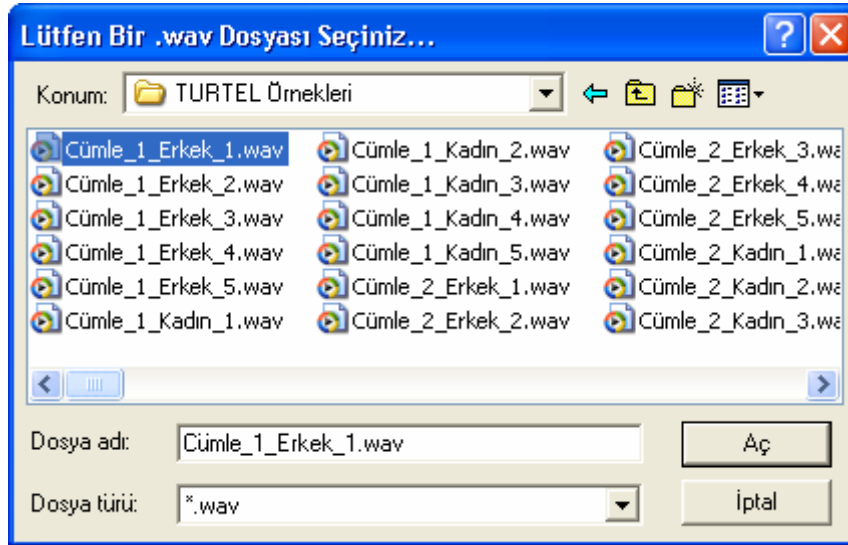
3.1 Sistem Değişkenlerinin Girilmesi ve Giriş İşaretinin Seçimi

Hazırlanan MATLAB yazılımı her ne kadar birkaç grafiksel kullanıcı ara yüzü (GUI) içerse de programı daha anlaşılır ve basit kılmak amacıyla yazılım tamamı ile MATLAB GUI araçları üzerine bina edilmemiştir. Programın başında ilk değer atamaları yapılmıştır. Giriş işaretinin örnekleme hızı ne olur ise olsun, giriş işareti belirlenen hızda (8 ksps) yeniden örneklenmektedir. (F_s) Örnekleme hızının değişimi dahili MATLAB komutları kullanılarak gerçekleştirilmiştir. Giriş işaretinin kaç kanal olursa olsun yalnızca ilk kanal kullanılmaktadır. Ayrıca çalışmamız gösterim ve eğitim amaçlı olduğundan hız ve bellek kullanımı gibi optimizasyonlar yapılmadığından, MATLAB programının uzun sürecek çalışma süresini sınırlamak için giriş işaretinin işlenecek uzunluğu belirli bir süre (10sn) ile sınırlandırılmıştır. (*Azami_Sure*) Kullanılan yapıda, işlenecek dosyaların tamamının belleğe okunması yerine, dosyaların kısa parçalarının okunup tedricen işlenmesi gibi bir tedbirle dahi bellek kullanımı ve çalışma hızı optimizasyonu mümkün olsa da, programın anlaşılabilirliği açısından bu yola gidilmemiştir.



Şekil 3.2 MATLAB GUI - sistem parametrelerinin alınması

Program ilk çalıştığında kullanıcıdan Şekil 3.2'deki diyalog penceresi ile öngörü derecesini ve kullanılacak pencere boyutunu belirlemesi istenir. Öngörü derecesi varsayılan değeri 10 olmak üzere 2 ile 99 arasında sınırlandırılmıştır. Alt sınır matematiksel açıdan olurlu en küçük değer olarak belirlenmiştir. Üst sınır ise iki nedenle sınırlandırılmıştır. Birincisi öngörünün derecesi en küçük pencere boyutundan (100'den) küçük olamaz, çünkü en az öngörü derecesi kadar örneğe ihtiyaç vardır. İkincisi doğrusal öngörü ile konuşma işareti kodlamada amaç bit oranının düşürülmesi olduğundan, artan öngörü derecesi artan parametre sayısı anlamına geldiğinden sıkıştırma anlamını yitirecektir. Ayrıca ses perdesinin belirlenmesinde öngörü hatasından yararlanıldığından, artan öngörü derecesi ile hata işareti çok küçülmekte, böylece kullandığımız ses perdesi belirleme algoritması olurlu sonuç vermemektedir. Ses perdesini belirleyen yöntem, LPC algoritmasında en sık kullanılan öngörü dereceleri (10~14) göz önüne alınarak belirlendiğinden 40 ve üzeri öngörü derecelerindeki yetersizliği kabul edilebilir. Pencere boyutu varsayılan değeri 200 örnek olmak üzere 100 ile 500 örnek arasında sınırlandırılmıştır. Alt sınır konuşma işaretinin özelliklerinin sağlıklı olarak çıkarılmasında kullanılabilecek en küçük değer olması hasebiyle, üst sınır ise çerçeve için konuşma işaretinin durağan sayılabileceği en üst sınır olması nedeniyle seçilmiştir. LPC analiz ve sentezin algoritmalarında %50 iç içe geçen (örtüşen) çerçeveler kullanıldığından çerçeve boyutunun çift sayı olması gerekmektedir. Bu husus LPC analiz uygulaması kısmında detaylandırılacaktır.



Şekil 3.3 MATLAB GUI – giriş işaretinin seçilmesi

Sistemin ana parametreleri alındıktan sonra Şekil 3.3'deki diyalog penceresi ile işlenecek .wav uzantılı dosya seçilecektir. Her ne kadar Windows işletim sistemi .wav uzantılı

tüm dosya biçimlerini (A-low, μ -low vb.) çalabilse de, MATLAB .wav uzantılı dosyaların yalnızca PCM formatlı olanlarını açabilmektedir. Açılmayan bir dosya formatı seçildiğinde kullanıcı uyarılmaktadır.

Son olarak Şekil 3.4'deki diyalog penceresi ile kullanıcıdan, program işlerken grafik animasyon isteyip istemediği bilgisi alınır. Animasyondan kasıt LPC analizinde giriş işareti çerçeveler halinde incelenirken, aktif pencere için giriş işaretinin frekans spektrumunun, öngörü süzgecinin frekans yanıtının, öngörü hatasının kısa zaman özilişki fonksiyonunun ve programın sonunda giriş ve çıkış işaretlerinin dalga şekli ile frekans spektrumlarının gösterilmesidir.



Şekil 3.4 MATLAB GUI – animasyon sorgusu

3.2 LPC Analizi

Bölüm 2.8'de öngörü hatasının çerçeve pozisyonu ile değişiminden bahsederken, çerçeve konumundan kaynaklanan hatanın giderilebilmesi için ön-vurgulama (pre-emphasize) süzgeçten yararlanılabileceğini ifade etmiştik. Temel olarak ön-vurgulama işaretin frekans spektrumunun düzeltilmeye çalışılmasıdır. Fourier Dönüşümü'nden de bildiğimiz üzere işaretin içerdiği harmonikler, temel bileşenine göre frekans eksenini boyunca sönümlenir. Ayrıca konuşma işaretinin temel frekans bileşenleri düşük seviyelerde olup, artan frekanslardaki spektral genlikler düşük olmaktadır. Buraya kadar anlatılanlardan anlaşılacağı üzere ön-vurgulama süzgecinin frekans yanıtının alt frekanslarda küçük ve üst frekanslara doğru artan bir spektral genliğe sahip olacağı kolayca anlaşılmaktadır. LPC süreçleri boyunca yapılan tüm işlemler doğrusal olduğuna göre, lineer sistem yaklaşımına göre, son olarak elde edilen işarete ön-vurgulama süzgecinin tersi uygulanır ise başlangıçta yapılan değişimin etkisi ortadan kalkmış olacaktır. Öyleyse vurgu kaldırma (de-emphasize) süzgecinin ön-vurgulama süzgecinin tersi olduğu aşıkardır. Burada filtrelerin nasıl tasarlandığına ver verilmeyip, yalnızca matematiksel ifadeleri verilecektir. (3.1) ifadesinde ön-vurgulama süzgeci fark denklemi ile tanımlanmıştır.

$$y(n) = x(n) - k \cdot x(n-1) \quad (3.1)$$

(3.1) ifadesinin z-dönüşümünü alalım;

$$Y(z) = X(z) - k \cdot z^{-1} \cdot X(z) \quad (3.2)$$

Buradan da transfer fonksiyonunu elde edelim;

$$H_1(z) = \frac{Y(z)}{X(z)} = 1 - k \cdot z^{-1} \quad (3.3)$$

Böylece (3.3) ifadesi ile ön-vurgulama süzgecini tanımlamış olduk. Vurgu kaldırma süzgeci de bunu tersi olduğuna göre fark denklemini elde edebiliriz.

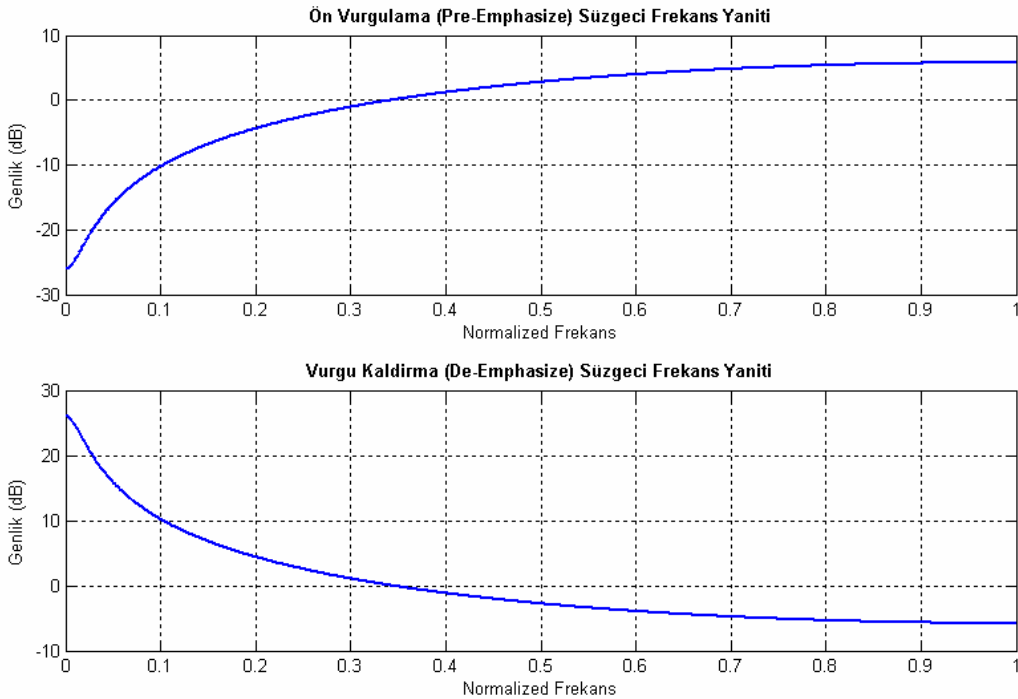
$$H_2(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - k \cdot z^{-1}} \quad (3.4)$$

(3.2) ifadesinin ters z-dönüşümünü alalım;

$$Y(z) - k \cdot z^{-1} \cdot Y(z) = X(z) \quad (3.5)$$

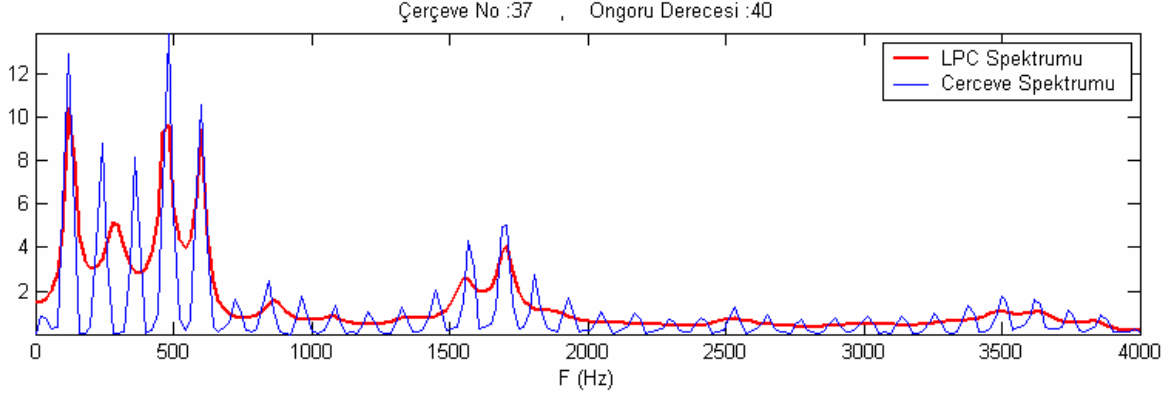
Buradan da fark denklemini elde edelim;

$$y(n) = x(n) + k \cdot y(n-1) \quad (3.6)$$



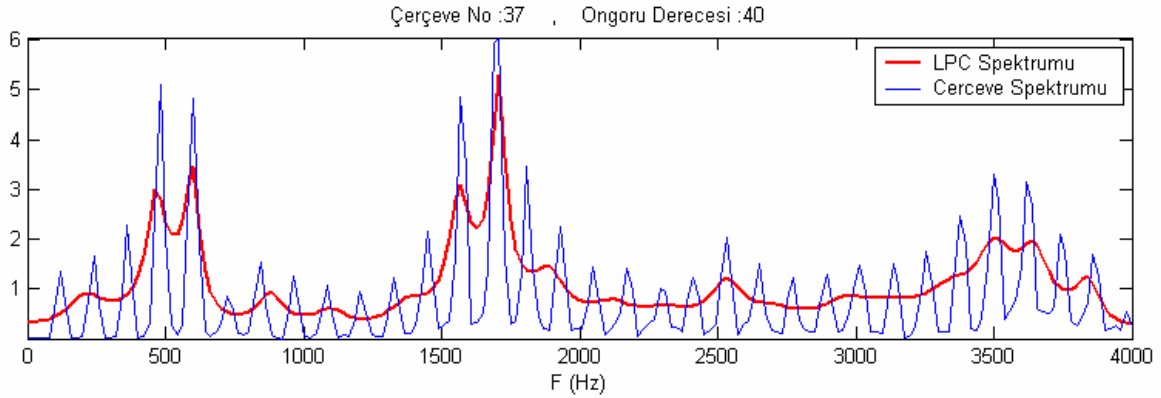
Şekil 3.5 Ön-vurgulama ve vurgu kaldırma süzgeçlerinin frekans yanıtları

Şekil 3.5’de $k=0.95$ için tasarlanan süzgeçlerin frekans yanıtları verilmektedir. Şekilden de görüleceği üzere, sistem teorisinin gereği olarak iki süzgecin frekans yanıtları çarpılacağından süzgeçlerin etkisi birbirini dengelemektedir.



Şekil 3.6 Ön-vurgulama süzgeci uygulanmayan işlem çerçevesi için frekans spektrumları

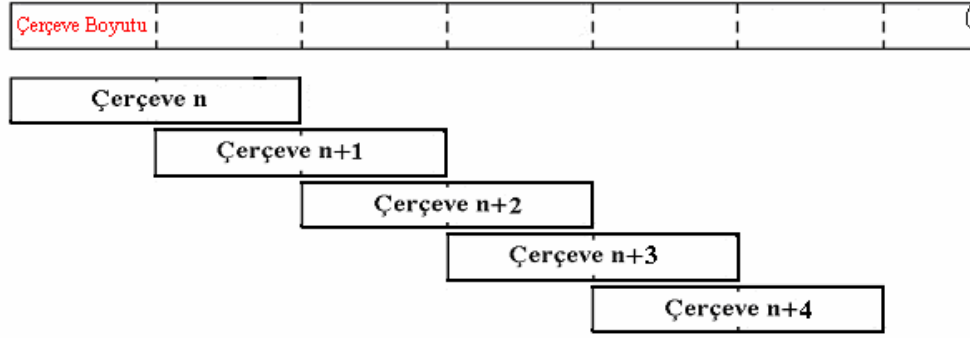
Ön vurgulama ve vurgu kaldırma süzgeçlerinin kullanımının kontrolü analiz ve sentez fonksiyonlarının birer giriş parametresi (*PreEmphasize*) olarak tanımlanmıştır. Bu parametre kullanılarak süzgeçlerin etkisi Şekil 3.6 ve Şekil 3.7’de, özellikle spektral düzleşme anlamında açıkça görülmektedir.



Şekil 3.7 Ön-vurgulama süzgeci uygulanan işlem çerçevesi için frekans spektrumları

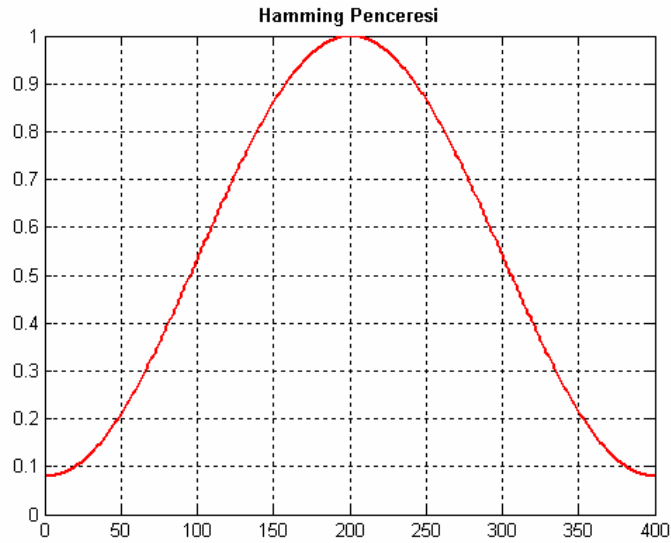
Giriş işareti vurgulama süzgecinden geçirildikten sonra çerçeveler halinde analiz parametreleri hesaplanmak üzere işlenir. İnsan anatomik yapısı nedeniyle konuşma işaretinin özelliklerinin yavaş değiştiğinden Bölüm 2’de defaten bahsetmiştik. Bu nedenle bir çerçevedeki özelliklerin diğerinde birdenbire değiştiğini veya çerçeve sınırlarının tam olarak ses özelliklerinin değişiminin olduğu yerlere denk geldiğini iddia etmek daha önceki açıklamalar ile çelişeceği mukimdir. Bu nedenle analiz çerçeveleri birbirleri üzerine örtüşecek

şekilde belirlenmiştir. Şekil 3.8’de çerçeve analiz yapısı grafiksel olarak gösterilmiştir. Burada analiz için kullanılan örneklerin uzunluğu belirlenen çerçeve boyutunun (*Çerçeve_Boyutu* değişkeni) iki katı olması başlangıçta yanlış anlaşılmalara sebebiyet verebilir. Çerçeve boyutunun iki katı miktarında örnek kullanılarak elde edilen parametreler yalnızca çerçeve boyutu kadar örneği temsil etmek üzere kullanılacaktır. Sentez işleminde kullanılan çerçeve yapısına bakılarak bu husus daha iyi anlaşılabilir.



Şekil 3.8 Uygulamada analiz için kullanılan çerçeve yapısı

Buraya kadar işlenecek çerçeveyi elde etmiş olduk. Doğrusal öngörü analizinde Bölüm 2.5.1’de tanıtilen özilişki yaklaşımı kullanılmıştır. Bu yaklaşıma göre işlem çerçevesi dışındaki tüm örnekler sıfırdır. Çerçeve dışındaki örneklere erişemediğimize göre çerçeve sınırlarında süreksizlikler oluşmaktadır. Tabii ki bu durum bize çerçeve sınırlarındaki geçişleri yumuşatacak çerçeve kullanımını çağırıştırır. Uygulamamızda (3.7) ifadesi ile tanımlanan *Hamming* penceresi kullanılmıştır. Şekil 3.9’de $N=400$ için *Hamming* penceresine ait grafik verilmiştir.

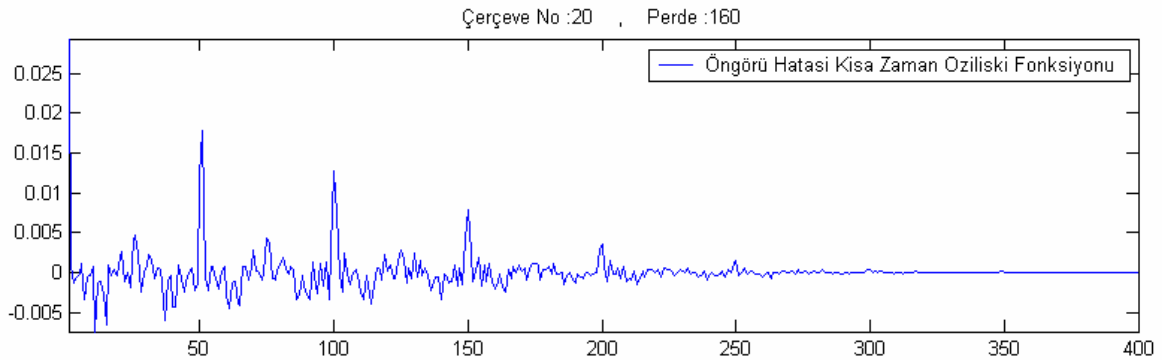


Şekil 3.9 Hamming penceresi

$$w(k) = 0.54 - 0.46 \cos\left(2\pi \frac{k-1}{N-1}\right), \quad k = 1, 2, \dots, N \quad (3.7)$$

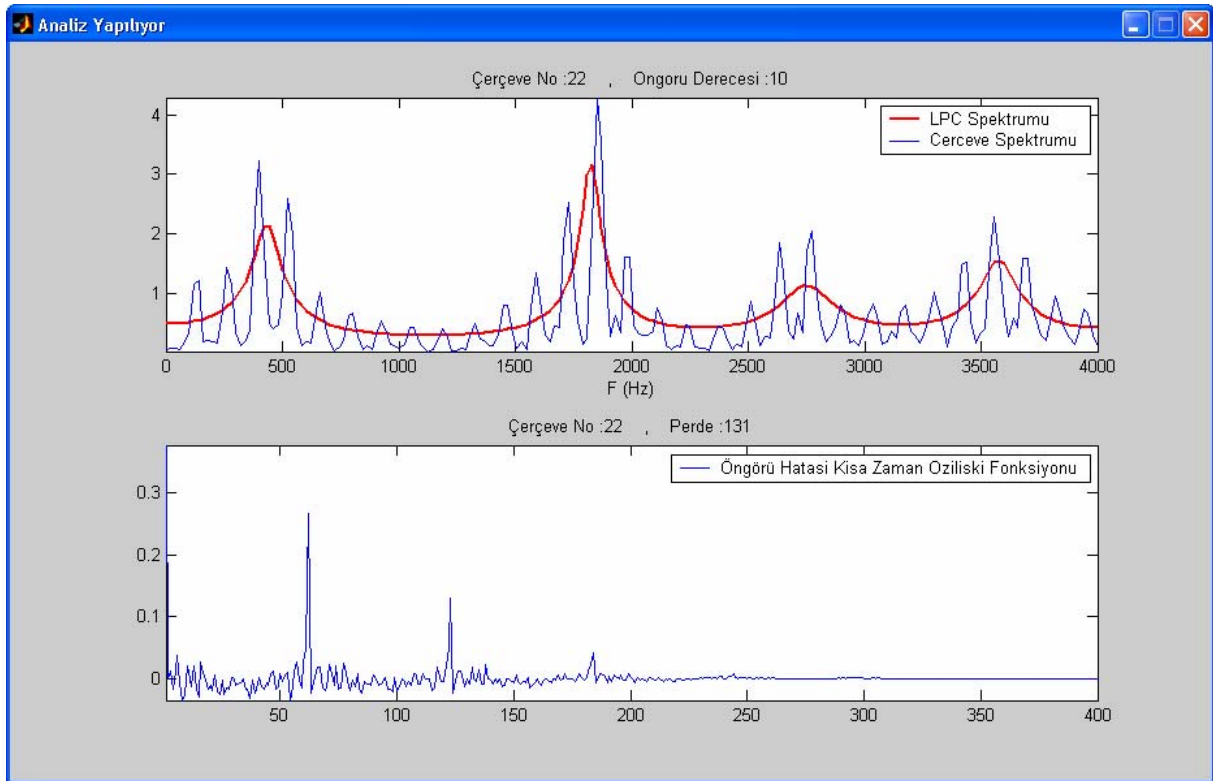
Şimdi elimizde doğrusal öngörü analizi yapabileceğimiz bir çerçeve teşkil edilmiş oldu. LPC analizinde Bölüm 2.5.1’de tanıtilan özilişki yaklaşımı kullandığımız için öncelikle Yule-Walker eşitliğini ve Teoplitz matrisinin elemanlarını tespit etmek için kısa zaman özilişki fonksiyonu hesaplanmaktadır. Sonra Levinson-Durbin yinelemesi kullanılarak doğrusal öngörü parametreleri hesaplanmaktadır. Bölüm 2.5.3’de de açıklandığı üzere Levinson–Durbin yinelemesinin her aşamasında karesel öngörü hatası elde edilmektedir. Bu özellik çerçeve için kazanç parametresinin hesaplanmasında kolaylık sağlamaktadır. Çünkü LPC yönteminde bahsi geçen kazanç uyarım işaretinin genliği, bir anlamda da enerjisi, anlamına geldiği ve ideal uyarım işaretinin de öngörü hata işareti olduğu göz önüne alındığında, kullanılacak uyarım işaretinin enerjisinin, öngörü hatasının enerjisine eşit olması gereği anlaşılabilmektedir. Algoritmamızda kazanç doğrudan karesel öngörü hatasını karekökü alınarak elde edilmektedir.

Doğrusal öngörü parametreleri ve kazanç elde edildiğine göre uyarım işaretinin teşkil edilebilmesi için kalan adımlar seslendirilen/seslendirilmeyen ayırımı ve perde tayinidir. Defaten bahsedildiği üzere ideal uyarım öngörü hata işaretidir. Öyle ise tasarladığımız model gereği teşkil edeceğimiz uyarım öngörü hata işareti incelenerek belirlenebilir. Bu amaçla işlenen çerçeve elde edilen doğrusal öngörü süzgecine uygulanarak öngörü hata işareti elde edilmektedir. Şekil 2.6’da verilen LPC sentezi blok diyagramında da görüldüğü üzere eğer çerçevemiz seslendirilen olarak sınıflandırılır ise uyarım işareti periyodik darbe katarından müteşekkil idi. Burada asıl amacımız öngörü hata işaretine benzeyen bir uyarım elde etmek ise, öncelikle öngörü hata işaretinde bir periyodiklik olup olmadığına bakılabilir. Böylece bir periyodiklik var ise hem seslendirilen/seslendirilmeyen ayırımı yapılabilir hem de işaretin perdesi tespit edilebilir.



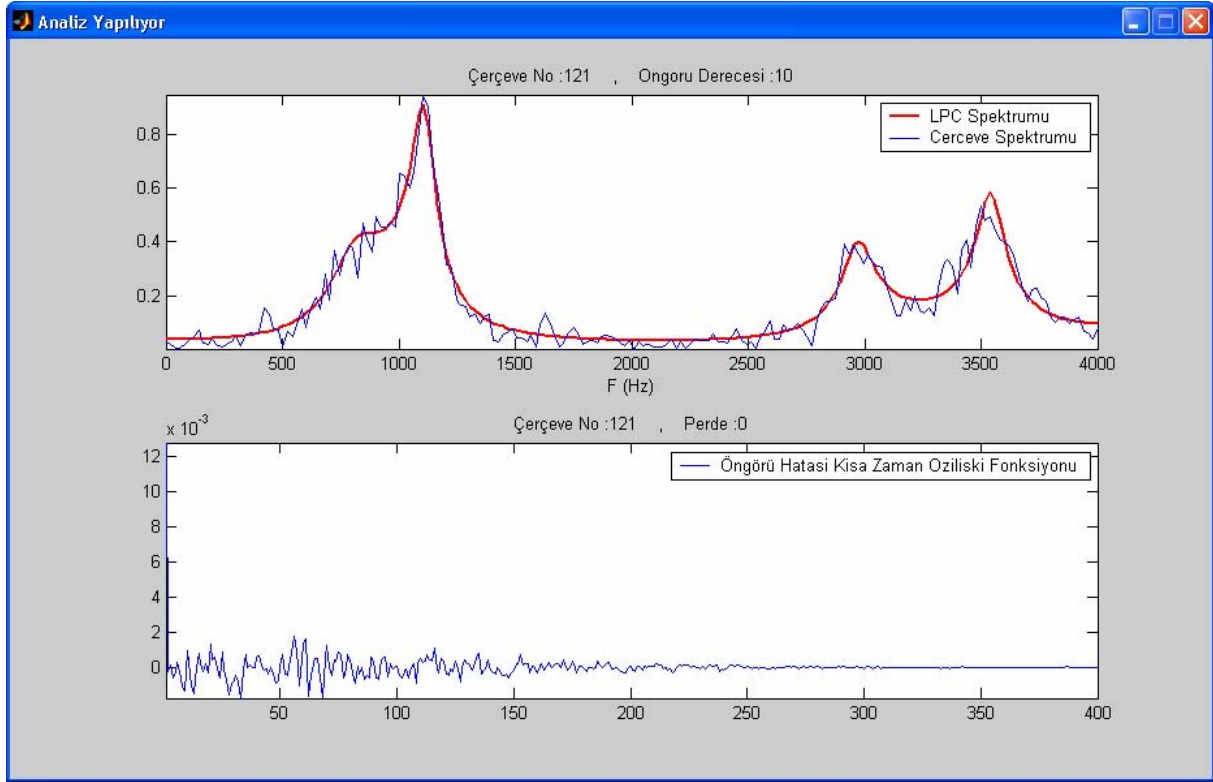
Şekil 3.10 Kısa zaman özilişki fonksiyonu

Kısa zaman özilişki fonksiyonu bir işaret içindeki periyodikliğin tespitinde de kullanılabilir. Eğer işaret sinüzoidlerin toplamı olarak düşünülür ise ve işaretle harmonikler mevcut ise işaretin kısa zaman özilişki fonksiyonunda harmoniklerin bulunduğu noktalarda temel bileşen ile ve diğer harmonikler ile olan benzerlik arttığından tepe noktalar oluşacaktır. Bu tepe noktalar arası ölçülerek işaretin perdesi tespit edilebilir. Ayrıca eğer belirgin bir periyodiklik var ise, bu harmoniklerde kısa zaman özilişki fonksiyonunun aldığı değerler, merkezde aldığı değerle mukayese edilebilir olmalıdır. Bu özellik de seslendirilen/ seslendirilmeyen ayrımında kullanılmaktadır. Örneğin uygulamada “kısa zaman özilişki fonksiyonunda gözlenen periyodik tepe değerleri merkezdeki değerin $\frac{1}{4}$ 'ünden büyük ise çerçeve seslendirilendir” şeklinde eşik belirlenmiştir. Şekil 3.10'da örnek bir kısa zaman özilişki fonksiyonu grafiği verilmiştir. Çerçeve boyutu 200 örnek (analiz penceresi 400 örnek) ve örnekleme hızı 8 ksps için verilen grafikte görüleceği üzere 50 örneklik aralıklar ile gözlenen periyodiklik mevcuttur. Bu durumda perde $8000/50=160$ olarak belirlenir. İlk harmoniğin genliği de eşiği aştığına göre seslendirilen sınıflandırılması yapılabilir. Şekil 3.10'da genlikleri daha küçük olmakla beraber farklı frekanslarda bileşenler de gözlenmektedir. Ne var ki belirlenen LPC modeli yalnızca tek ve formant frekansını uyarım olarak kabul etmektedir. Ayrıca durum her zaman Şekil 3.10'daki kadar belirgin olmayıp, belirlenen eşiğin çok az altında kalabilen, tespiti zor işaretler ile de karşılaşılmaktadır.



Şekil 3.11 Analiz animasyon örneği - I

LPC analizi MATLAB uygulamasına ilişkin son adım eğer isteniyor ise görsel animasyon yapılmasıdır. Şekil 3.11 ve Şekil 3.12’de bu grafiklere örnekler verilmiştir. Grafiklerin üst kısımlarında işlenen çerçevenin frekans spektrumu ve parametreleri elde edilen doğrusal öngörü süzgecinin frekans yanıtları, alt kısımlarında da öngörü hata işaretinin kısa zaman özilişki fonksiyonları verilmektedir. Grafiklerin üstlerinde de işlenen çerçevenin numarası, öngörü derecesi ve belirlenen perde yazılmaktadır.



Şekil 3.12 Analiz animasyon örneği - II

3.3 LPC Parametrelerinin Dosyaya Yazılması ve Dosyadan Okunması

MATLAB uygulamamızın veri sıkıştırma sonuçlarını gösterebilmek amacıyla elde edilen doğrusal öngörü parametreleri özel tanımlı bir dosyaya yazılıp okunmaktadır. Fakat amaçlardan biri de veri sıkıştırma olduğundan, parametreler hesaplandıkları gibi 64 bit kayan noktalı sayılar ile kayıt edilmemiştir. Aynı husus verinin bir kanaldan iletiminde de mevzu bahistir. Bu nedenle parametreler kuantalanarak daha düşük bit sayıları ile ifade edilebilir hale getirilmektedir. Ayrıca Bölüm 2.13’de belirtildiği üzere doğrusal öngörü filtresi katsayıları, a_k ’lar, doğrudan kuantalandıklarında, sayısal çözünürlüklerden dolayı kutup yerleri değişebilmekte ve sistem kararsız hale gelebilmektedir. a_k katsayıları sayısal çözünürlüğe fazlaca duyarlı olduklarından kuantizasyonda yansıma katsayılarının (PARCOR)

kullanılması tercih edilmiştir. MATLAB uygulamasını anlaşılabilirliği amacıyla , analiz kısmından sentez kısmına halihazırda elde edilmiş olan yansıma katsayıları transfer edilecek olsa da analiz kısmında doğrusal öngörü süzgeç katsayıları da elde edilip, verilerin yazılması esnasından yansıma katsayıları yeniden elde edilmektedir.

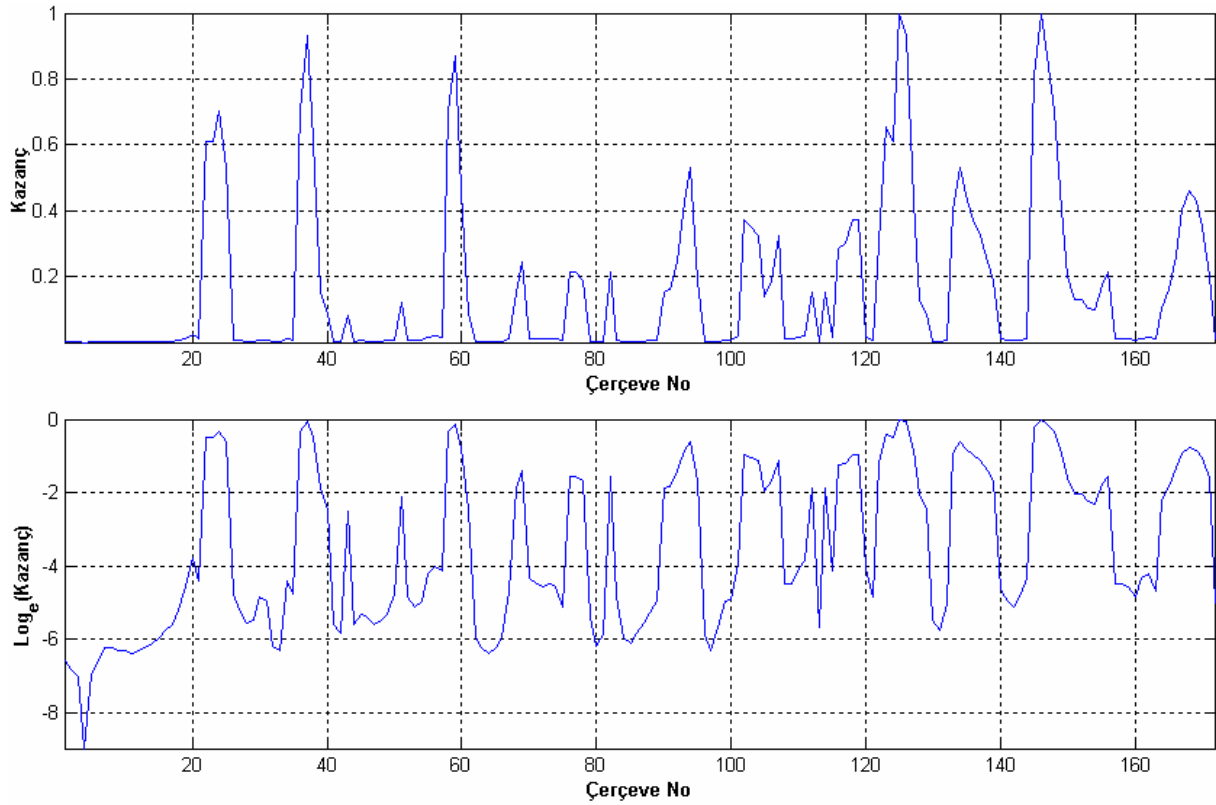
Verilerin yazıldığı özel tanımlı dosyanın işlenirken kontrol edilebilmesi amacıyla dosyaya başlık ve sonlu ifadeler ('*LPC*') eklenmiştir. Dosyaya yazılan bazı parametreler veri transferinde veya saklamada yalnızca bir kez yer alacağından bit oranı hesabında kullanılmayacaktır. Örneğin örnekleme frekansı, öngörü derecesi ve çerçeve boyutu gibi parametreler analiz ve sentez kısımlarında mutabakat ile belirlemiş ise transferi gereksizdir. Çizelge 3.1'de özel tanımlı dosya yapısı verilmiştir.

Çizelge 3.1 Özel tanımlı dosya yapısı

Başlık, <i>LPC</i>	8 bit karakter	3 adet
Örnekleme frekansı, F_s	16 bit tamsayı	1 adet
Öngörü derecesi, p	8 bit tamsayı	1 adet
Çerçeve boyutu, N	9 bit tamsayı	1 adet
Çerçeve adedi, i	16 bit tamsayı	1 adet
Yansıma katsayıları	8 bit kuantalanmış 64 bit kayan nokta	$i \cdot p$ adet
Perde	8 bit kuantalanmış 64 bit kayan nokta	i adet
Kazanç	8 bit kuantalanmış 64 bit kayan nokta	i adet
Maks. yansıma katsayı	32 bit kayan nokta	p adet
Maks. Perde	16 bit tamsayı	1 adet
Maks. Kazanç	32 bit kayan nokta	1 adet
Sonluk, <i>LPC</i>	8 bit karakter	3 adet

Çizelge 3.1'de görüldüğü üzere başlık, sonluk, örnekleme frekansı, öngörü derecesi, çerçeve boyutu ve çerçeve adeti dosyaya hiçbir uyarlama yapılmaksızın yazılmaktadır. Yansıma katsayıları ise MATLAB dahili komutları kullanılarak öngörü süzgeci katsayılar polinomundan elde edilmektedir. 64 bit kayan nokta hesaplanan yansıma katsayıları, kazanç ve perde yine MATLAB dahili komutları kullanılarak doğrusal olarak 8 bit kuantize edilmektedir. MATLAB kuantizasyon fonksiyonu azami $\pm A$ arasında değerler alan işareti belirlenen bit sayısı ile ifade edilen tamsayı aralığına karşılık getirmektedir. Örneğin -1000 ila 1000 arasında değer alan bir işaret 8 bit ile kuantize edildiğinde , -1000 değeri 0'a, 1000 değeri de 255'e karşılık gelmektedir. Anlaşılacağı üzere dosyadan veriler okunduğunda kuantize edilen işaretin yeniden elde edilmesi için bu işaretin azami değerine de ihtiyaç duyulacaktır. Bu nedenle yansıma katsayıları, kazanç ve perde parametreleri için azami

değerler de dosyaya yazılmaktadır. Esasen perde de kayan nokta sayılarla hesaplanmaktadır, ne var ki kulağımız 1 Hz frekansı çözünürlükte bile olmadığından tamsayıya yuvarlanmıştır. Bu nedenle diğer parametrelerin azami değerleri kayan nokta iken perdeninki tamsayıdır. Her ne kadar tüm parametreler doğrusal kuantizasyona tabii tutulsa da, kazanç parametresinin logaritması alınarak kuantizasyonun logaritmik olması sağlanmıştır. Daha önce de defaten belirtildiği üzere konuşma işaretinin kazancı seslendirilen kısımlarda, seslendirilmeyen kısımlara göre nispeten yüksektir. Öyle ki, 8 bit ile doğrusal kuantizasyon yapıldığında seslendirilen kısımlar için kazanç sıfır dışında bir değer almasına karşın, seslendirilmeyen kısımlar için çoğu zaman sıfır veya çok küçük değerler almaktadır. Yani seslendirilmeyen kısımlar için kazanç parametresinin çözünürlüğü çok düşmektedir. Şekil 3.13'teki kazanç parametresinin ve logaritmasının grafiğinden kazancın sıfır gibi gözüktüğü çerçevelerde bile logaritma ile kazancın ayırt edilebilir olduğu görülebilmektedir.



Şekil 3.13 Kazanç parametresi ve logaritması

Dosyadan okuma işleminde ise başlık ve sonluk kontrol edilerek dosya yapısı doğrulandıktan sonra yapılan işlem, yazma kısmındakini tersinden ibarettir, kuantalanmış verilerin kodları çözülür, kazancın anti logaritması alınır ve yansıma katsayılarından doğrusal öngörü süzgeci katsayıları elde edilir.

Bu tez çalışmasının amaçlarından biri de 2.4 kb/s mertebesinde sıkıştırma yapabilen bir ses kodlayıcının gerçekleştirilmesi idi. Şimdi 8 kps örnekleme hızı ve 25ms'lik (200 örnek) çerçeve boyutu için bit oranına hesaplayalım. Daha önce de belirtildiği üzere bit oranı hesaplanırken başlık, sonluk, örnekleme frekansı, öngörü derecesi ve çerçeve boyutu analiz ve sentez kısımları mutabık olduklarında kullanılmayabileceğinden, azami değerler de yalnızca bir kez transfer edileceğinden hesaba katılmayacaktır. Çizelge 3.2'de görüleceği üzere çerçeve başına 96 bit kullanılmaktadır.

Çizelge 3.2 Çerçeve başına bit miktarı- I

Yansıma katsayıları	8 bit kuantalanmış 64 bit kayan nokta	10 adet	80 bit
Perde	8 bit kuantalanmış 64 bit kayan nokta	1 adet	8 bit
Kazanç	8 bit kuantalanmış 64 bit kayan nokta	1 adet	8 bit
			96 bit

Bir çerçeve süresi (25ms) için 96 bit gerekiyor ise, bit oranımız $96/0.025=3.84$ kb/s olarak hesaplanmaktadır. Standartlaşmış LPC algoritması olan FS1015' de, 22.5ms olan çerçeve için farklı kuantizasyon kullanıldığından 54 bit gerekmekte ve $54/0.0225=2.4$ kb/s gibi bir bit oranı elde edilmektedir. Alternatif olarak Çizelge 3.3'te verilen kuantizasyon değerleri ile yapılan denemede bit oranımız $64/0.025=2.56$ kb/s olarak elde edilmiştir fakat dikkatli dinlendiğinde sesin sentetikliğinde fark edilebilecek düzeyde bir artış fark edilebilmektedir. MATLAB uygulamasında kullanılacak kuantizasyon yöntemi *Kuantizasyon* adlı parametre ile kontrol edilebilmektedir. Elde edilen 2.56 kb/s sıkıştırma oranı ile hedef olarak belirlenen 2.4 kb/s oranına yaklaşabilmiştir. FS1015 standardında kullanılan kuantizasyon yapısı eklerde verilmiştir.

Çizelge 3.3 Çerçeve başına bit miktarı- II

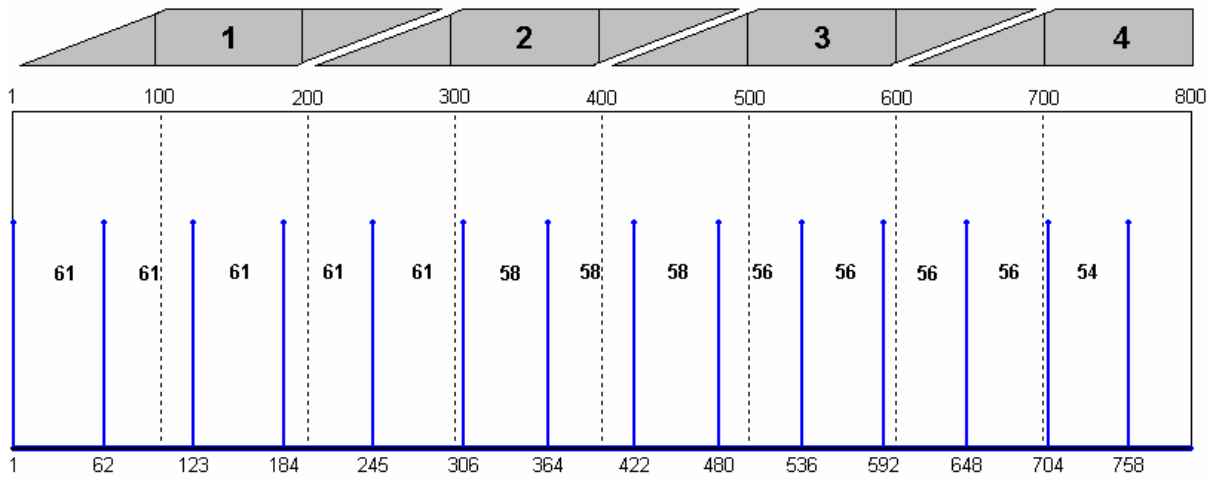
Yansıma katsayıları	5 bit kuantalanmış 64 bit kayan nokta	10 adet	50 bit
Perde	7 bit kuantalanmış 64 bit kayan nokta	1 adet	7 bit
Kazanç	7 bit kuantalanmış 64 bit kayan nokta	1 adet	7 bit
			64 bit

3.4 LPC Sentezi

LPC sentez işlemi modelimizin de öngördüğü üzere Şekil 2.6'da verilen blok diyagramın gerçekleştirilmesinden ibarettir. Not edilmesi gereken tek farklılık çerçeveler arasındaki geçişi yumuşatmak için kullanılan çerçevelerin örtüştürülmesi işlemidir, yani

sistem Şekil 2.6'dakine ek olarak örtüşürmenin gerçekleşmesi için bellek de içermektedir.

Sesin perdesini ifade eden değişkenin sıfır olması, çerçevenin seslendirilmeyen olmasını ifade etmektedir ve bu durumda uyarım işareti olarak MATLAB dahili fonksiyonları kullanılarak normal dağılımlı (standart sapması 1 olan, sıfır ortalamalı) rasgele işaret ile teşkil edilmektedir. Sıfırdan farklı bir perde belirlenmesi durumunda ise uyarım işareti perde ile belirlenen periyotta darbe katarı kullanılarak teşkil edilmektedir. Yalnız bu durumda çerçeveler arası geçişi yumuşatmak için, çerçevenin bir önceki çerçeve ile örtüşecek kısmında, bir önceki çerçeveye ait perde kullanılmaktadır. Böylece çerçeveler arasında perdenin değişimindeki geçiş yumuşatılmaya çalışılmaktadır.



Şekil 3.14 Seslendiren çerçeveler için uyarım işareti teşkili

Şekil 3.14'de ardı ardına gelen dört seslendirilen çerçeveye ilişkin uyarım işareti gösterilmektedir. Çerçeveler için tespit edilen perdeler sırasıyla 131, 138, 143 ve 148 frekanslarındadır ve 8 ksps örnekleme hızı için perde periyotları sırasıyla 61, 58, 56 ve 54 örnek olmaktadır. Grafiğin üst kısmında da uyarım işaretlerinin doğrusal öngörü süzgecine uygulanmasından sonra elde edilen işaretlerin nasıl örtüşecek şekilde birleştirildikleri gösterilmektedir. Örneğimiz için çerçeve genişliği 200 örnektir ve %50'lik bir örtüşme yapısı kullanılmaktadır. Şekilden de görüleceği üzere ilk çerçeve için darbeler 61 örnek aralıklardır. İkinci çerçevede ise ilk çerçeve ile örtüşecek kısımlarda darbe aralıkları bir önceki perde periyodunda yani 61 örnek aralıktır. İkinci çerçevede geri kalan darbeler 58 örnek aralıktır. Üçüncü çerçevede ise yine bir önceki çerçeve ile örtüşecek kısımda, önceki çerçeveye ait perde kullanılmış sonra gerçek perde değeri ile devam edilmiş ve bu durum seslendirilmeyen çerçeveye gelinceye kadar devam etmektedir.

Esasen seslendirilen kısımda teşkil edilen uyarım işareti yalnızca darbelerden müteşekkil değildir. Çünkü uyarım işaretinin teşkilinde amaç öngörü hata işaretine mümkün mertebe yaklaşmaktır. Öngörü hata işaretinin yalnızca darbelerden müteşekkil olmadığı aşikardır. Bu nedenler uyarım işaretine çok küçük genlikli normal dağılımlı gürültü de eklenmektedir. Bu yaklaşık ses kalitesinde olumlu yönde hissedilir değişikliğe neden olmaktadır. Uyarım işareti teşkil edildikten sonra belirlenen doğrusal öngörü katsayıları ve kazanç ile tasarlanan süzgece uygulanmakta ve çıkışlar Şekil 3.14'ün üst kısmında gösterilen yapıda birbirleri ile birleştirilerek çıkış işareti elde edilir.

4. SONUÇ

Bu tez çalışmasında elde edilmek istenen sonuç Mutlak Kategori Test'ine göre orta ve üstünde ses kalitesine sahip, 2.4 kb/s mertebesinde sıkıştırma yapabilen bir ses kodlayıcının MATLAB programı üzerinde gerçekleştirilmesi idi. Belirlediğimiz farklı kuantizasyon yapıları ile 3.84 kb/s ve 2.56 kb/s sıkıştırma oranları elde edilmiştir. Testlerde ses kalitesi daha tatminkar görülen 3.84 kb/s bit oranına sahip kodlayıcının sonuçları kullanılmıştır.

Çizelge 3.4 Mutlak Kategori Test Dereceleri

5	Mükemmel
4	İyi
3	Orta
2	Zayıf
1	Kötü

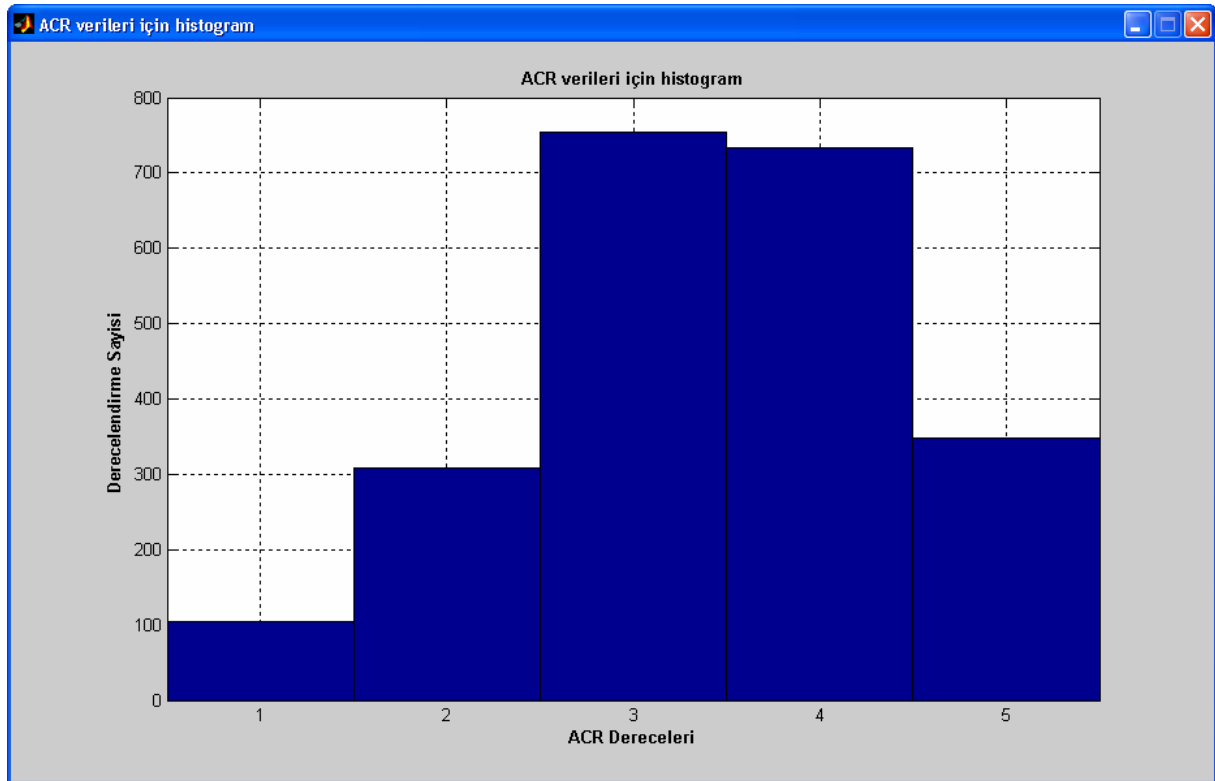
Orijinal ses örnekleri ve ses kodlayıcı çıktıları bir CD ile ve bir Internet haberleşme grubu ile katılımcılara dağıtılarak Çizelge 3.4'e göre Mutlak Kategori Test'i yapılmıştır. Tasarlanan ses kodlayıcısının kalitesi, yaş ortalamaları 31 olan 8 erkek ve 7 kadın dinleyici ile yapılan bu testte 3.4 olarak derecelendirilmiştir. Orijinal ses örnekleri stüdyo ortamında değil de telefondan kayıt edildiği için ve kodlayıcının amacı orijinal sesi elde etmek olduğundan, karşılaştırmalı değerlendirme yapımları için test katılımcılarına “*Orijinal sesi 5 ile derecelendirir iseniz sentetik olana kaç verirdiniz ?*” sorusu yöneltilmiştir. Sonuçları daha iyi değerlendirebilmek için test verileri her bir dinleyiciye göre, ses örneklerindeki konuşmacılara göre ve ses örneklerindeki cümlelere göre incelenmiştir.

Şekil 4.1'de ACR test verileri için histogram verilmiştir. Ses kalitesi yapılan derecelendirmeler görüldüğü üzere 3 ve 4 üzerinde yoğunlaşmıştır. Grafikten test sonuçlarının Gauss dağılımı gösterdiği söylenebilir. Bu da testin doğrallığının ve güvenilirliğinin bir ölçüsü

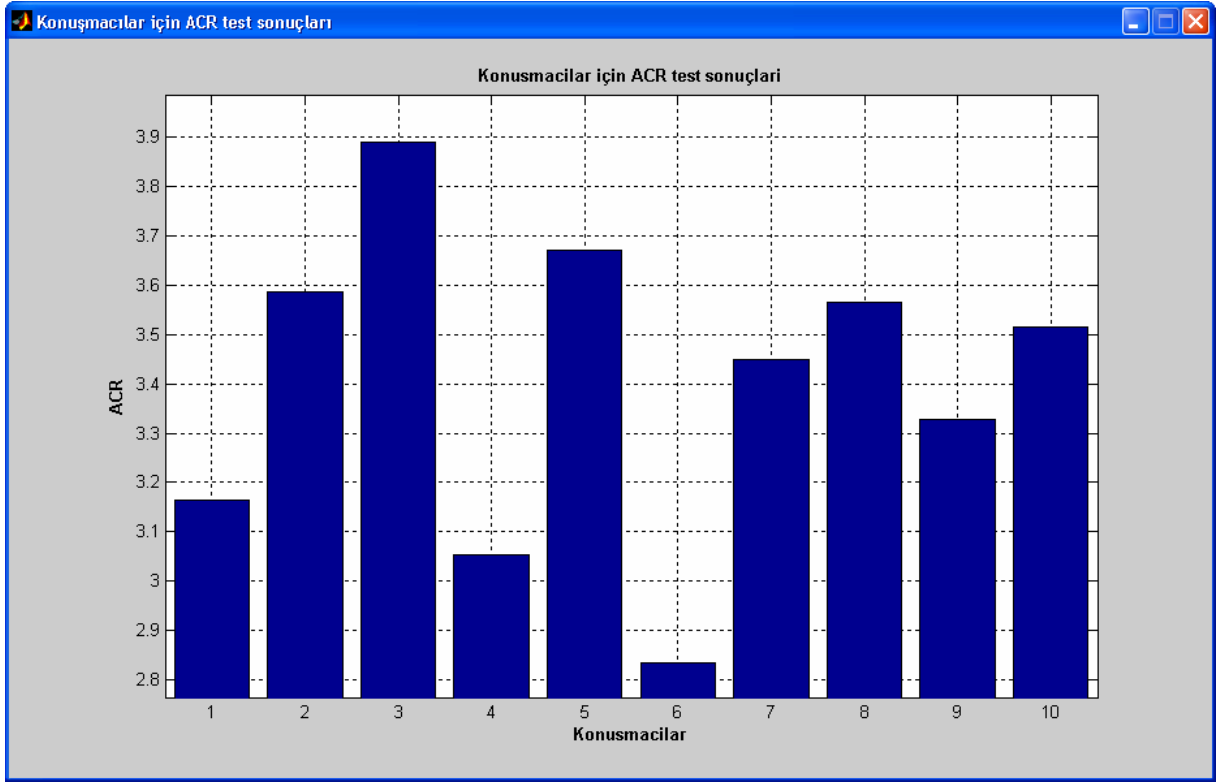
olarak düşünülebilir.

Şekil 4.2’de her bir konuşmacı için test sonuçları verilmiştir. İlk 5 konuşmacı erkekler sonraki 5 konuşmacı kadınlardır. 1. ve 4.erkek konuşmacı ile 1. kadın konuşmacı diğerlerine nispeten oldukça düşük dereceler almışlardır. Orijinal kayıtlar üzerinde yapılan karşılaştırmalardan bu konuşmacıların, sesleri tam olarak çıkarmadıkları, yutkunarak boğazlarını temizlemeden hırıltılı konuştukları, kelimeler arasında “eee”, “ııı” gibi sesler çıkardıkları ve aksanlarının diğer konuşmacılar kadar iyi olmadığı gözlenmiştir. İlk üç neden model göz önünde bulundurulduğunda kabul anlaşılır niteliktedir. Yalnız aksan ile ilgili son nedenin test katılımcılarının ses kalitesini değerlendirmek yerine algısal olarak ses içeriği hakkında derecelendirme yapmasından kaynaklandığı düşünülmektedir.

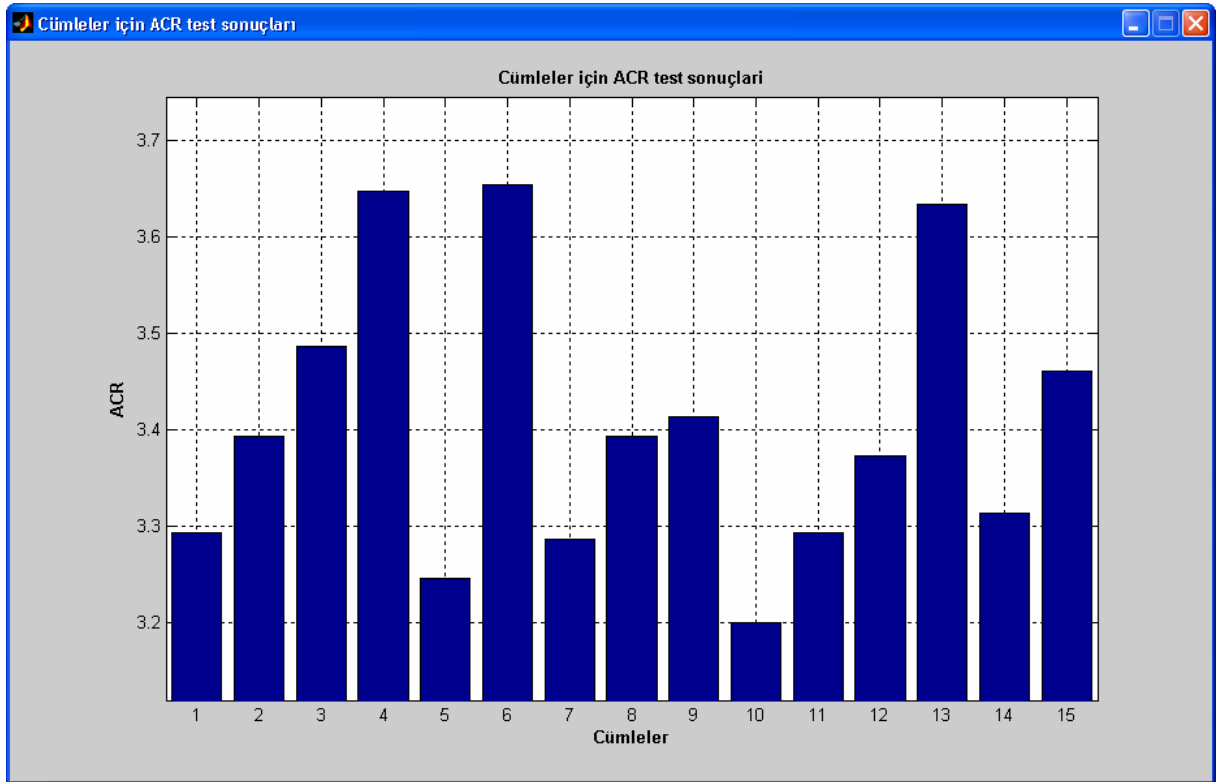
Şekil 4.3’te her bir cümle için ACR test sonuçları verilmiştir. Her ne kadar dudak ve burun sesi olan cümleler nispeten düşük dereceler almış olsa da gözlemlenen istisnalardan dolayı düşük ve yüksek derece alan cümleler arasında sağlıklı bir sınıflandırma yapılamamıştır. Şekil 4.4’te ise testte katılan dinleyicilerin ACR test sonuçları verilmiştir. Burada da derecelendirmeler arasındaki bu denli farkı açıklayan bir yorum geliştirilememiştir.



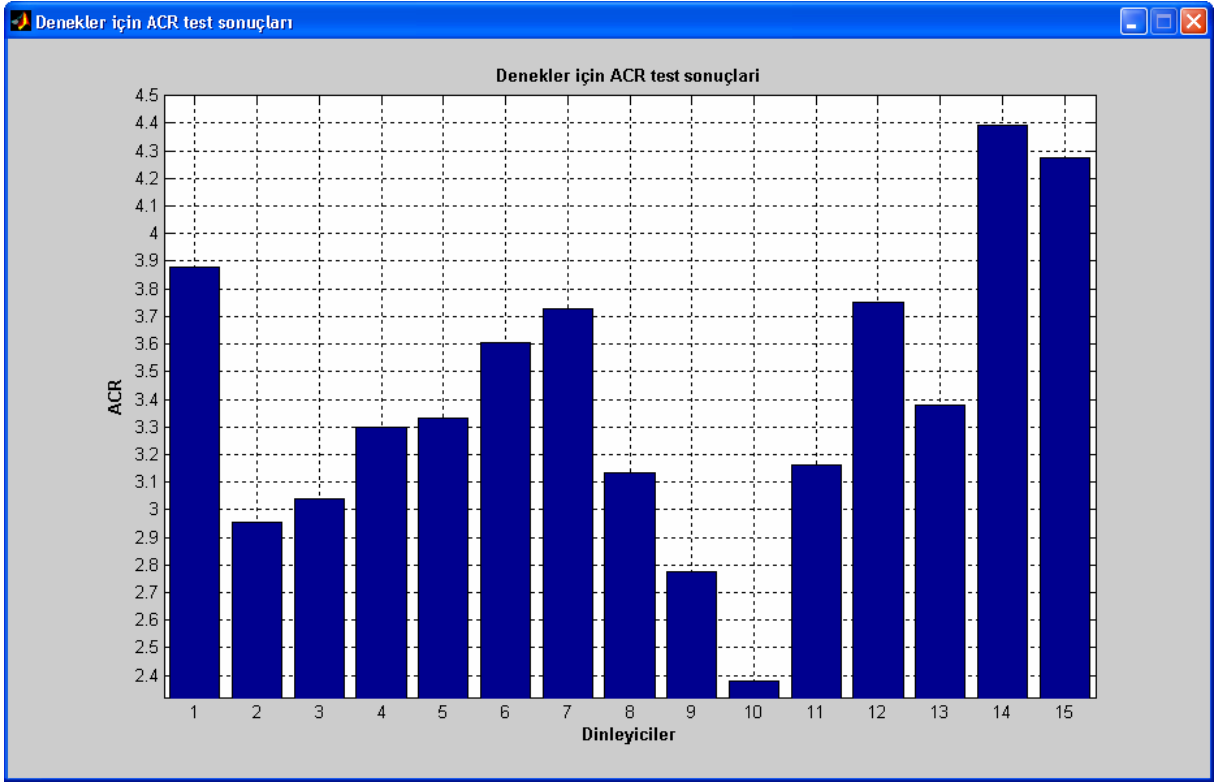
Şekil 4.1 ACR test verileri için histogram



Şekil 4.2 Konuşmacılar için ACR test sonuçları



Şekil 4.3 Test cümleleri için ACR test sonuçları



Şekil 4.4 Denekler için ACR test sonuçları

KAYNAKLAR

Bae, S.H., (2004), “LPC10, 2.4kbps Federal Standard in Speech Coding”, Ders Notu, Georgia Institute of Technology, Atlanta.

Bradbury J.S., (2000), “Linear Predictive Coding”, Teknik Rapor, Queen’s University, Ontario.

Kularatna, N. ve Dias, D., (2004), Essentials of Modern Telecommunications Systems, Artech House Publishers, Norwood

Lemmetty, S., (1999), “Review of Speech Synthesis Technology”, Yüksek Lisans Tezi, Helsinki University of Technology, Helsinki.

Park, S. (2005), “Linear Predictive Speech Processing”, Ders Notu, Texas A&M University, Texas.

Patin, F. ve Violette A., (2005), “Vocodeur Paramétrique à Deux Sources”, Yüksek Lisans Tezi, Speléc University.

Rabiner, L.R. ve Schafer, R.W., (1978), Digital Processing Of Speech Signals, Prentice Hall, New Jersey.

Rahim, M., (2004), “Signal Processing Methods for Speech Analysis”, Ders Notu, AT&T Labs., New Jersey.

Smith, J.O., (1998), “Linear Predictive Coding”, Ders Notu, Stanford University, Stanford.

EKLER

- Ek 1 TURTEL kütüphanesi örnek cümleleri
- Ek 2 FS1015 standart LPC-10 algoritması kuantizasyon yapısı
- Ek 3 MATLAB uygulaması, LPC ana program
- Ek 4 MATLAB uygulaması, LPC analizi
- Ek 5 MATLAB uygulaması, LPC parametrelerinin özel formatlı dosyaya yazılması
- Ek 6 MATLAB uygulaması, LPC parametrelerinin özel formatlı dosyadan okunması
- Ek 7 MATLAB uygulaması, LPC sentezi
- Ek 8 Ses kodlama tekniklerinin karşılaştırılması

Ek 1 TURTEL kütüphanesi örnek cümleleri

1. Yağmurdan sonra güneş çıkarsa gökkuşağı oluşur.
2. Rezervasyon için bilet almak gerekir mi ?
3. Projenin faturası bana çıkartıldı.
4. Verdiğin çiçekte örümcek var.
5. Askerler depremde kurtarma çalışmalarına katıldılar.
6. Ümit demektir yağmur çiftçiler için.
7. Sinemaya bilet bulabildin mi ?
8. Ayvalıkta güneşin batışını seyretnmelisin.
9. Yemek için rezervasyon yaptırdım.
10. Acı bir iz bıraktı deprem zihinlerimizde.
11. Dün manifaturacıya gittim.
12. Dünyanın en harika dokumacılarıdır örümcekler.
13. Fıkranın konusu bir çiçek.
14. Kafanda nasıl projeler var.
15. Demek büyüyünce asker olacaksın.

Ek 2 FS1015 standart LPC-10 algoritması kuantizasyon yapısı

FS1051 standardında perde periyodu 60 farklı değer alabilecek şekilde 6 bit ile logaritmik kuantizasyona tabii tutulur. Kazanç ise 5 bit logaritmik kuantizasyona tabii tutulur.

Eğer çerçeve seslendirilen ise 10. dereceden öngörücü kullanılır ve 10 adet yansıma katsayısı gerektirir, çerçeve seslendirilmeyen ise 4. dereceden öngörücü kullanılır ve 4 adet yansıma katsayısı gerektir.

Her ne kadar yansıma katsayıları doğrusal öngörü süzgeci katsayılar polinomu göre sayısal çözünürlüğe daha az duyarlı ise de ilk iki yansıma katsayı bir yakın olduğundan yine de bir tedbir alınmalıdır. Bu nedenle k_1 ve k_2 yansıma katsayıları aşağıda verilen dönüşüm ile yeniden uyarlanır ve beş bit kuantizasyona tabii tutulur.

$$g_i = \frac{1+k_i}{1-k_i} \quad (B.1)$$

Yansıma katsayıları k_3 ve k_4 doğrudan beş bit kuantizasyona tabii tutulur. Seslendirilen çerçeveler için k_5 , k_6 , k_7 , k_8 dört bit, k_9 üç bit ve k_{10} iki bit ile kuantizasyona tabii tutulur. Seslendirilmeyen çerçeveler için k_5 'ten k_{10} 'a kadar olan katsayıların kullandığı 21 bit hata kontrol işlemleri için kullanılır ve sabit çerçeve yapısı ve bit hızı elde edilmiş olur. Tabii ki bu durumda bir bit de seslendirilen / seslendirilmeyen ayırımı için kullanılır.

Standarda göre 8 ksp/s örnekleme hızı ile 180 örneklik (22.5ms) çerçeveler için 54 bit gerekmektedir. Bu durumda da bit oranı $54 / 0.025 = 2.4$ kb/s olarak elde edilmektedir.

Çizelge Ek2.1 FS1015 kuantizasyon yapısı

1 bit	Seslendirilen/seslendirilmeyen bayrağı
6 bit	Perde periyodu
10 bit	Yansıma katsayıları k_1 ve k_2
10 bit	Yansıma katsayıları k_3 ve k_4
16 bit	Yansıma katsayıları k_5 , k_6 , k_7 ve k_8
3 bit	Yansıma katsayısı k_9
2 bit	Yansıma katsayısı k_{10}
5 bit	Kazanç
1 bit	Senkronizasyon
54 bit	Toplam

Ek 3 MATLAB uygulaması, LPC ana program

% LPC analiz ve sentezi demosu

close all; clear all; clc

Fs = 8000 ; *% Belirlediğimiz örnekleme hızı, örnek/saniye, azami 65535*
Azami_Sure = 10 ; *% Belirlenen azami işaret süre, saniye*
PreEmphasize = 1 ; *% 1 ise pre-emphasize süzgeç kullanılır.*
Kuantizasyon = 1 ; *% 1 ise yansıma katsayıları, perde ve kazanç 8 bit kuantalanır.*
% 0 ise yansıma katsayıları 5, perde ve kazanç 7 bit kuantalanır.

Sorular = { 'Öngörü Derecesi 2..99' , 'Çerçeve Boyutu (Çift Sayı) 100..500' } ;

Varsayılanlar = { '10' , '200' } ;

while 1

Parametreler = inputdlg(Sorular, 'LPC Parametreleri', 1, Varsayılanlar, 'Off') ;

if size(Parametreler) == [0 0]

clear all, return

else

Ongoru_Derecesi = str2double(Parametreler(1)) ;

Cerceve_Boyutu = str2double(Parametreler(2)) ;

if Ongoru_Derecesi<2 || Ongoru_Derecesi>99 || isnan(Ongoru_Derecesi)...

|| Cerceve_Boyutu<100 || Cerceve_Boyutu>500 || isnan(Cerceve_Boyutu)...

|| mod(Cerceve_Boyutu,2)==1

beep

questdlg('Lütfen uygun değerler giriniz....', 'Hatalı giriş !!!', 'TAMAM', 'TAMAM') ;

else

break

end

end

end

clc, clear Sorular Varsayılanlar Parametreler

% İşlenecek işaret wav formatındaki dosyadan okunuyor.

Calisma_Dizini = cd ;

while 1

[DOSYA_ADI, DOSYA_YOLU] = uigetfile('*.wav', 'Lütfen Bir .wav Dosyası Seçiniz...') ;

if ischar(DOSYA_ADI) == 0 || ischar(DOSYA_YOLU) == 0

clear all, return

end

cd(DOSYA_YOLU) ;

try

[GIRIS, Fg] = wavread(DOSYA_ADI) ;

break

catch

beep

questdlg('Lütfen geçerli bir .wav dosyası seçiniz...', 'Geçersiz dosya !!!', 'TAMAM', 'TAMAM') ;

end

end

cd(Calisma_Dizini) ;

clear Calisma_Dizini DOSYA_ADI DOSYA_YOLU

% İki veya daha fazla kanallı işaretin yalnızca bir kanalı alınır.

GIRIS = GIRIS(:,1) ;

% Örnekleme hızı için gerekli düzenleme yapılıyor.

if Fg ~= Fs

GIRIS = resample(GIRIS, Fs, Fg) ;

GIRIS = GIRIS / max (abs(GIRIS)) ;

end

clear Fg ;

% İşaretin 10 saniyeden fazla olan kısmı kesiliyor.

if length(GIRIS) > Fs*Azami_Sure

GIRIS = GIRIS(1:Fs*Azami_Sure) ;

beep

questdlg('Azami Süre değişkeni ile belirlenen süreden uzun olan giriş işareti kırpıldı !', ...
'Uyarı', 'TAMAM', 'TAMAM') ;

end

clear Azami_Sure

% Animasyon yapılsın mı ?

SECIM = **questdlg**('Animasyon yapılsın mı ?', 'Lütfen seçiminizi yapınız.', 'EVET', 'HAYIR', 'HAYIR') ;

if strcmp(SECIM, 'EVET')

Animasyon = 1 ;

elseif strcmp(SECIM, 'HAYIR')

Animasyon = 0 ;

elseif strcmp(SECIM,")

clear all, **return**

end

% LPC analizi

[Katsayılar, Perde, Kazanc] = **LPC_Analizi**(GIRIS, Fs, Ongoru_Derecesi, Cerceve_Boyutu, ...
PreEmphasize, **Animasyon**) ;

try

if Kuantizasyon == 1

% Belirlenen formatta LPC.LPC dosyası yazılıyor.

LPC_Dosyasi_Yaz(Fs, Cerceve_Boyutu, Katsayılar, Perde, Kazanc, 'LPC.LPC') ;

% LPC.LPC dosyasında veri okunuyor.

[Katsayılar, Perde, Kazanc, Fs, Cerceve_Boyutu] = **LPC_Dosyasi_Oku**('LPC.LPC') ;

else

% Belirlenen formatta LPC.LPC dosyası yazılıyor.

LPC_Dosyasi_Yaz_1(Fs, Cerceve_Boyutu, Katsayılar, Perde, Kazanc, 'LPC.LPC') ;

% LPC.LPC dosyasında veri okunuyor.

[Katsayılar, Perde, Kazanc, Fs, Cerceve_Boyutu] = **LPC_Dosyasi_Oku_1**('LPC.LPC') ;

end

catch

return

end

% LPC sentezi

CIKIS = **LPC_Sentezi**(Katsayılar, Perde, Kazanc, Fs, Cerceve_Boyutu, **PreEmphasize**) ;

% Ses çalınırken ve kayıt edilirken oluşacak kırpılma önleniyor.

if max(CIKIS) >= 1

CIKIS = CIKIS / max(CIKIS) * 0.9999 ;

end

wavwrite(CIKIS, Fs, 'LPC_CIKIS.wav')

if Animasyon == 1

EkranBoyutu = **get**(0, 'ScreenSize') ;

figure('Position', [50 100 EkranBoyutu(3)-100 EkranBoyutu(4)-200] , ...

'Name', 'GİRİŞ ve ÇIKIŞ işaretlerinin dalga şekilleri', ...

'NumberTitle', 'off', 'MenuBar', 'none')

subplot(2,1,1)

plot(GIRIS(1:length(CIKIS)))

title('\bfOrijinal Konusma Isareti'), **axis tight**

subplot(2,1,2)

plot(CIKIS)

title('\bfLPC Ile Modellenen Konusma Isareti'), **axis tight**

```

figure( 'Position', [ 50 100 EkranBoyutu(3)-100 EkranBoyutu(4)-200 ] , ...
        'Name', 'GİRİŞ ve ÇIKIŞ işaretlerinin spektrogramları', ...
        'NumberTitle', 'off', 'MenuBar', 'none' )

subplot(2,1,1)
specgram( GIRIS(1:length(CIKIS)), 512, Fs, kaiser(500,5), 475 )
xlabel( '\bfOrişinal Konusma Isaretinin Spektrogramı' )
ylabel( '\bfFrekans' )

subplot(2,1,2)
specgram( CIKIS, 512, Fs, kaiser(500,5), 475 )
xlabel( '\bfLPC Ile Modellenen Konusma Isaretinin Spektrogramı' )
ylabel( '\bfFrekans' )
end

% Ses kalitesi dinleyicinin deęerlendirmesine sunuluyor.
while 1
    SECIM = questdlg( 'Lütfen seçiminizi yapınız.', 'Sonuçları karşılaştıralım...', ...
        'ORİJNAL', 'SENTEZLENEN', 'Çıkış', 'Çıkış' ) ;
    if strcmp( SECIM, 'ORİJNAL' )
        sound( GIRIS, Fs )
    elseif strcmp( SECIM, 'SENTEZLENEN' )
        sound( CIKIS, Fs )
    elseif strcmp(SECIM,'Çıkış')
        clear SECIM, break
    end
end
end

```

Ek 4 MATLAB uygulaması, LPC analizi

% LPC Analizi

**function [Katsayılar, Perde, Kazanc] = LPC_Analizi(GIRIS, Fs, Ongoru_Derecesi, Cerceve_Boyutu, ...
PreEmphasize, Animasyon) ;**

% Pre-emphasize süzgeç uygulanıyor

if PreEmphasize == 1

GIRIS = filter([1 -0.95], [1], GIRIS) ;

end

if Animasyon == 1

EkranBoyutu = get(0, 'ScreenSize') ;

**figure('Position', [50 100 EkranBoyutu(3)-100 EkranBoyutu(4)-200], 'Resize', 'off', ...
'Name', 'Analiz Yapılıyor', 'NumberTitle', 'off', 'MenuBar', 'none')**

else

% İlerleme çubuğunu başlat

İlerleme_Cubugu = waitbar(0, 'Lütfen bekleyiniz...', 'Name', 'Analiz Yapılıyor') ;

İlerleme_Cubugu_Maks = ceil((length(GIRIS)-2*Cerceve_Boyutu+1) / Cerceve_Boyutu) ;

end

Cerceve_Indeksi = 1 ;

for i = 1 : Cerceve_Boyutu : length(GIRIS)-2*Cerceve_Boyutu+1

% İşlenen çerçeve Hamming penceresine tabi tutuluyor.

CERCEVE = GIRIS(i : i-1+2*Cerceve_Boyutu) .* hamming(2*Cerceve_Boyutu) ;

% Kısa zamanlı özilişki fonksiyonu hesaplanıyor.

% M...İşlem çerçeve boyutu (2 x Cerceve_Boyutu) , 0 ≤ k ≤ Ongoru_Derecesi

R = zeros(1 , Ongoru_Derecesi+1) ;

for k = 0 : Ongoru_Derecesi

for m = 0 : 2*Cerceve_Boyutu-1-k

R(k+1) = R(k+1) + CERCEVE(m+1)*CERCEVE(m+1+k) ;

end

end

% Levinson yinelemesi ile doğrusal öngörü katsayıları hesaplanıyor.

k(1) = R(2)/ R(1) ;

a(1,1) = k(1) ;

Karesel_Ongoru_Hatasi(2) = R(1) - R(2)*R(2)/R(1) ;

for m = 2 : Ongoru_Derecesi

Toplam = 0 ;

for n = 1 : m-1

Toplam = Toplam + a(m-1,n) * R(m-n+1) ;

end

k(m) = (R(m+1) - Toplam) / Karesel_Ongoru_Hatasi(m) ;

a(m,m) = k(m) ;

for n = 1 : m-1

a(m,n) = a(m-1,n) - k(m) * a(m-1,m-n) ;

end

Karesel_Ongoru_Hatasi(m+1) = (1 - k(m)*k(m)) * Karesel_Ongoru_Hatasi(m) ;

end

Katsayılar(Cerceve_Indeksi,:) = [1 -a(Ongoru_Derecesi,:)] ;

Karesel_Ongoru_Hatasi = abs(Karesel_Ongoru_Hatasi(Ongoru_Derecesi+1)) ;

% Hata işareti ideal uyarım işareti olduğuna göre hata işaretinin enerjisi uyarım işaretinin enerjisi ile

% aynı olmalıdır. Karesel hata enerji olarak yorumlanır ise uyarım işaretinin genliği hesaplanabilir.

Kazanc(Cerceve_Indeksi) = sqrt(Karesel_Ongoru_Hatasi) ;

% Öngörü hata işareti hesaplanıyor.

Ongoru_Hatasi = filter(Katsayılar(Cerceve_Indeksi,:), 1 , CERCEVE) ;

% Öngörü hata işareti kullanılarak segmentin perdesi tespit ediliyor. Bu amaçla öngörü hata işaretindeki

% periyodiklik araştırılıyor. Kısa zaman özilişki fonksiyonu hesaplanıyor.

*% M....İşlem çerçeve boyutu (2 x Cerceve_Boyutu) , 0 <= k <= 2*Cerceve_Boyutu-1*

R = zeros(1 , 2*Cerceve_Boyutu) ;

for k = 0 : 2*Cerceve_Boyutu-1

for m = 0 : 2*Cerceve_Boyutu-1-k

R(k+1) = R(k+1) + Ongoru_Hatasi(m+1)*Ongoru_Hatasi(m+1+k) ;

end

end

[R_Sirali , Indeks] = sort(R) ;

if R_Sirali(2*Cerceve_Boyutu-1) > 0.25 * R_Sirali(2*Cerceve_Boyutu)

Perde(Cerceve_Indeksi) = Fs / abs(Indeks(2*Cerceve_Boyutu) - Indeks(2*Cerceve_Boyutu-1)) ;

if Perde(Cerceve_Indeksi) >= 4000

Perde(Cerceve_Indeksi)=Fs/abs(Indeks(2*Cerceve_Boyutu)-Indeks(2*Cerceve_Boyutu-2)) ;

end

else

Perde(Cerceve_Indeksi) = 0 ;

Kazanc(Cerceve_Indeksi) = Kazanc(Cerceve_Indeksi) / sqrt(Cerceve_Boyutu);

end

if Perde(Cerceve_Indeksi) > 600

Perde(Cerceve_Indeksi) = 0 ;

Kazanc(Cerceve_Indeksi) = Kazanc(Cerceve_Indeksi) / sqrt(Cerceve_Boyutu);

end

% Aktif çerçeve için orijinal ve LPC spektrumları hesaplanıp gösteriliyor.

if Animasyon == 1

if strempi(get(gcf,'Resize'),'on')

**set((gcf, 'Position', [50 100 EkranBoyutu(3)-100 EkranBoyutu(4)-200], 'Resize', 'off', ...
'Name', 'Analiz Yapılıyor', 'NumberTitle', 'off', 'MenuBar', 'none')**

end

LPC_SPEKTRUM = freqz(sqrt(Karesel_Ongoru_Hatasi),...

Katsayılar(Cerceve_Indeksi,:), Cerceve_Boyutu) ;

CERCEVE_SPEKTRUM = abs(fft(CERCEVE,2*Cerceve_Boyutu)) ;

w = 0:0.5*Fs/(Cerceve_Boyutu-1):Fs/2 ;

subplot(2,1,1), plot(w, abs(LPC_SPEKTRUM), 'r', 'LineWidth',2), hold on

plot(w, CERCEVE_SPEKTRUM(1:Cerceve_Boyutu),'b'), axis tight, hold off

title(['Çerçeve No : ' int2str(Cerceve_Indeksi) ' , Ongoru Derecesi : ' int2str(Ongoru_Derecesi)])

legend('LPC Spektrumu', 'Cerceve Spektrumu', xlabel('F (Hz)')

subplot(2,1,2), plot(1:2*Cerceve_Boyutu, R), axis tight

title(['Çerçeve No : ' int2str(Cerceve_Indeksi) ' , Perde : ' int2str(Perde(Cerceve_Indeksi))])

legend('Öngörü Hatasi Kısa Zaman Özilişki Fonksiyonu', drawnow

else

% İlerleme çubuğunu güncelle

waitbar(Cerceve_Indeksi/İlerleme_Cubugu_Maks)

end

Cerceve_Indeksi = Cerceve_Indeksi + 1 ;

end

if Animasyon ~= 1

% İlerleme çubuğunu kapat

close(İlerleme_Cubugu)

end

Ek 5 MATLAB uygulaması, LPC parametrelerinin özel formatlı dosyaya yazılması

% LPC formatlı dosya yazma

function LPC_Dosyasi_Yaz(Fs, Cerceve_Boyutu, Katsayilar, Perde, Kazanc, DOSYA_ADI) ;

Ongoru_Derecesi = size(Katsayilar) ;
Cerceve_Adedi = Ongoru_Derecesi(1,1) ;
Ongoru_Derecesi = Ongoru_Derecesi(1,2)-1 ;

*% Öngörü filtresi polinomu yansıma katsayılarına dönüştürülüyor. Eğer doğrudan öngörü filtresi
 % polinomu (katsayıları) kuantize edilir ise, bu parametrelerin sayısal çözünürlüğe olan yüksek
 % bağımlılıklarından dolayı filtre kararsız olabilir.*

for i = 1:Cerceve_Adedi
 PARCOR(i,:) = poly2rc(Katsayilar(i,:))' ;
end

% Perde tamsayıya yuvarlanıyor.

Perde = round(Perde) ;

% Kazanç logaritmik quantizasyondan için hazırlanıyor.

for i=1:Cerceve_Adedi
 if Kazanc(i) == 0
 Kazanc = 0.001;
 end
end
Kazanc = log(Kazanc);

Max_Katsayilar = max(abs(PARCOR)) ;
Max_Perde = max(1, max(Perde)) ;
Max_Kazanc = max(abs(Kazanc)) ;

for i = 1:Ongoru_Derecesi
 PARCOR(:,i) = uencode(PARCOR(:,i), 8, Max_Katsayilar(i)) ;
end
Perde = uencode(Perde, 8, Max_Perde) ;
Kazanc = uencode(Kazanc, 8, Max_Kazanc) ;

% Parametreler dosyaya yazdırılıyor.

DOSYA = fopen(DOSYA_ADI, 'w+') ;

if DOSYA == -1
 beep, questdlg('Dosyasi Olusturulamadi !', 'Dosya Hatası', 'TAMAM', 'TAMAM') ; return
end

fwrite(DOSYA, 'LPC', 'uchar') ;
fwrite(DOSYA, Fs, 'uint16') ;
fwrite(DOSYA, Cerceve_Boyutu, 'ubit9') ;
fwrite(DOSYA, Cerceve_Adedi, 'uint16') ;
fwrite(DOSYA, Ongoru_Derecesi, 'uint8') ;
fwrite(DOSYA, PARCOR, 'uint8') ;
fwrite(DOSYA, Perde, 'uint8') ;
fwrite(DOSYA, Kazanc, 'uint8') ;
fwrite(DOSYA, Max_Katsayilar, 'float32') ;
fwrite(DOSYA, Max_Perde, 'uint16') ;
fwrite(DOSYA, Max_Kazanc, 'float32') ;
fwrite(DOSYA, 'LPC', 'uchar') ;

if fclose(DOSYA) == -1
 beep, questdlg('Dosyasi Kapatilamiyor !', 'Dosya Hatası', 'TAMAM', 'TAMAM') ; return
end

Ek 6 MATLAB uygulaması, LPC parametrelerinin özel formatlı dosyadan okunması

% LPC formatlı dosyayı okuma

function [Katsayilar, Perde, Kazanc, Fs, Cerceve_Boyutu] = LPC_Dosyasi_Oku(DOSYA_ADI) ;

% Parametreler dosyadan okunuyor.

DOSYA = fopen(DOSYA_ADI, 'r') ;

if DOSYA == -1

beep

questdlg('Dosyasi Acilamadi !', 'Dosya Hatası', 'TAMAM', 'TAMAM') ;

return

end

% Dosya formatı kontrol ediliyor.

if strcmp(fread(DOSYA,3,'uchar=>char') , 'LPC') == 0

beep

questdlg('Uygunsuz dosya formatı !', 'Dosya Hatası', 'TAMAM', 'TAMAM') ;

return

end

Fs = fread(DOSYA, 1, 'uint16') ;

Cerceve_Boyutu = fread(DOSYA, 1, 'ubit9') ;

Cerceve_Adedi = fread(DOSYA, 1, 'uint16') ;

Ongoru_Derecesi = fread(DOSYA, 1, 'uint8') ;

PARCOR = fread(DOSYA, [Cerceve_Adedi, Ongoru_Derecesi], 'uint8=>uint8') ;

Perde = fread(DOSYA, Cerceve_Adedi, 'uint8=>uint8') ;

Kazanc = fread(DOSYA, Cerceve_Adedi, 'uint8=>uint8') ;

Max_Katsayilar = fread(DOSYA, Ongoru_Derecesi, 'float32') ;

Max_Perde = fread(DOSYA, 1, 'uint16') ;

Max_Kazanc = fread(DOSYA, 1, 'float32') ;

% Dosya formatı kontrol ediliyor.

if strcmp(fread(DOSYA,3,'uchar=>char') , 'LPC') == 0

beep

questdlg('Uygunsuz dosya formatı !', 'Dosya Hatası', 'TAMAM', 'TAMAM') ;

return

end

if fclose(DOSYA) == -1

beep

questdlg('Dosyasi Kapatilamiyor !', 'Dosya Hatası', 'TAMAM', 'TAMAM') ;

return

end

for i = 1:Ongoru_Derecesi

PARCOR_fp(:,i) = udecode(PARCOR(:,i), 8, Max_Katsayilar(i)) ;

end

Perde = udecode(Perde, 8, Max_Perde) ;

Kazanc = exp(udecode(Kazanc, 8, Max_Kazanc)) ;

% Yansıma katsayıları öngörü filtresi polinomuna dönüştürülüyor.

for i = 1:Cerceve_Adedi

Katsayilar(i,:) = rc2poly(PARCOR_fp(i,:)) ;

end

Ek 7 MATLAB uygulaması, LPC sentezi

% LPC Sentezi

function CIKIS = LPC_Sentezi(Katsayılar, Perde, Kazanc, Fs, Cerceve_Boyutu, PreEmphasize);

Oteleme = 0 ;

Yeni_Oteleme = 0 ;

Ortusen_Parca = zeros(1 , Cerceve_Boyutu/2) ;

Cerceve_Adedi = size(Katsayılar) ;

Cerceve_Adedi = Cerceve_Adedi(1) ;

% İlerleme çubuğunu başlat

İlerleme_Cubugu = waitbar(0, 'Lütfen bekleyiniz...', 'Name','Sentez Yapılıyor') ;

for i = 1 : Cerceve_Adedi

if Perde(i) == 0 *% Seslendirilmeyen çerçeve ise*

 Uyarim_Isareti = randn(1 , 1.5*Cerceve_Boyutu) ;

 Oteleme = 0 ;

else *% Seslendirilen çerçeve ise*

if (i-1)<1 || Perde(i-1)==0

 Periyod_1 = floor(Fs / Perde(i)) ;

else

 Periyod_1 = floor(Fs / Perde(i-1)) ;

end

 Periyod_2 = floor(Fs / Perde(i)) ;

 Uyarim_Isareti = 0.001*randn(1 , 1.5*Cerceve_Boyutu + Oteleme) ;

 Periyod_Guncelle = 0 ;

 Periyod = Periyod_1 ;

 n = 1 - Periyod ;

while n <= (1.5*Cerceve_Boyutu + Oteleme - Periyod_2)

 n = n + Periyod;

if (n > Cerceve_Boyutu + Oteleme) && (Periyod_Guncelle==0)

 Yeni_Oteleme = Cerceve_Boyutu + 1 + Oteleme + Periyod - n ;

 Periyod_Guncelle = 1;

end

if n > Cerceve_Boyutu / 2

 Periyod = Periyod_2 ;

end

 Uyarim_Isareti(n) = Uyarim_Isareti(n) + min(1,Periyod_2/100) ;

end

end

% Uyarım işareti modeli teşkil eden süzgeçten geçiriliyor.

CERCEVE = filter(Kazanc(i) , Katsayılar(i,:) , Uyarim_Isareti) ;

for k = 1 : Cerceve_Boyutu/2

 Olcek = k / (Cerceve_Boyutu/2) ;

 CIKIS(k+(i-1)*Cerceve_Boyutu+Cerceve_Boyutu/2)=CERCEVE(k+Oteleme+Cerceve_Boyutu/2);

 CIKIS(k+(i-1)*Cerceve_Boyutu)=CERCEVE(k+Oteleme)*Olcek + Ortusen_Parca(k)*(1-Olcek) ;

 Ortusen_Parca(k) = CERCEVE(k+Oteleme+Cerceve_Boyutu);

end

 Oteleme = Yeni_Oteleme ;

 waitbar(i/Cerceve_Adedi) *% İlerleme çubuğunu güncelle*

end

% Pre-emphasize süzgeç etkisi dengeleniyor.

if PreEmphasize == 1

 CIKIS = filter([1], [1 -0.95], CIKIS) ;

end

close(İlerleme_Cubugu) *% İlerleme çubuğunu kapat*

Ek 8 Ses kodlama tekniklerinin karşılaştırılması

Çizelge Ek8.1 Ses kodlama tekniklerinin karşılaştırılması

Algoritma	Bit Oranı (kbit/s)	MOS	Komplekslik (MIPS)	Çerçeve Boyutu (ms)
PCM G.711	64	4.3	0.01	0
ADPCM G.726	32	4.1	2	0.125
SBC G.722	48/56/64	4.1	5	0.125
LD-CELP G.728	16	4.0	30	0.125
CS-ACELP(-A) G.729	8	4.0 (3.8)	20 (11)	10
MPC-MLQ G.723.1	6.3/5.6	4.0/3.7	11	10
GSM HR VSELP	6.3	3.4	14	20
IS-54 VSELP	8	3.5	14	20
IS-96 QCELP	1.2/2.4/4.8/9.6	3.3	15	20
Inmarsat-B APC	9.6/12.8	3.1/3.4	10	20
MELP	2.4	3.2	40	22.5
FS1016 - CELP	4.8	3.2	16	30
FS1015 - LPC10	2.4	2.3	7	22.5

ÖZGEÇMİŞ

Doğum tarihi 07.04.1978

Doğum yeri Kars

Lise 1992-1996 Tuzla Anadolu Meslek Lisesi

Lisans 1998-2002 Uludağ Üniversitesi Mühendislik Mimarlık Fak.
Elektronik Mühendisliği Bölümü

Lisans 2002-2004 Yıldız Üniversitesi Elektrik-Elektronik Fak.
Elektronik ve Haberleşme Mühendisliği Bölümü

Yüksek Lisans 2004-2007 Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Elektronik ve Haberleşme Müh. Anabilim Dalı,
Elektronik Programı

Çalıştığı kurumlar

2003-2004	Enda Endüstriyel Elektronik Ltd.
2004-2005	Bilgi Teknolojileri Enstitüsü – Tübitak MAM
2006-2007	SVS Telekomünikasyon Hizmetleri Ltd.