

DÜŞÜK BİT HIZLARINDA KONUŞMA KODLAMA VE UYGULAMALARI

DOKTORA TEZİ

Y.Müh. Tarık AŞKIN

100661

Tezin Enstitüye Verildiği Tarih : 21 Ocak 1998

Tezin Savunulduğu Tarih : 7 Ocak 1999

Tez Danışmanı : Prof. Dr. Günsel DURUSOY

G. Duruso 23.2.1999
G. Duruso

Diğer Jüri Üyeleri : Prof. Dr. Eşref ADALI

Prof. Dr. Emin ANARIM

Emin Anarim
Emin Anarim

Prof. Dr. Ömer CERİD

Doç. Dr. Ümit AYGÖLÜ

Ümit AYGÖLÜ 23.2.1999

OCAK 1999

ÖNSÖZ

Bu konuda çalışmamı sağlayan sayın Prof. Dr. Günsel Durusoy'a, sağladığı sonsuz destek ve motivasyondan dolayı Simko Tic. ve San. AŞ'ne ve bana katkıları, ayrıca bitmek tükenmek bilmeyen özverileri ile beni teşvik eden yakınlarıma ve aileme teşekkür ederim.

Ayrıca doktora çalışması için beni daima teşvik etmiş, değerli insan, rahmetli hocam sayın Yük. Müh. Yücel Durusoy'a teşekkürü bir borç bilirim. Ruhu şad olsun.

Tarık Aşkın
Ocak 1998



İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	ii
ŞEKİL LİSTESİ.....	vii
TABLO LİSTESİ.....	xii
ÖZET	xiii
SUMMARY.....	xvi
BÖLÜM 1 GİRİŞ.....	1
1.1 Düşük Bit Hızlarında Konuşma Kodlama.....	2
1.2 Tez ile Getirilen Yenilikler	4
BÖLÜM 2 KONUŞMANIN FİZYOLOJİK OLARAK İNCELENMESİ.....	5
2.1 Konuşmanın Yapısı.....	5
2.2 Konuşma Üretiminin Modellenmesi	7
BÖLÜM 3 KONUŞMA KODLAMA;GEREKSİNİMLER VE UYGULAMALAR.....	11
3.1 Konuşma Kodlama Prensipleri	11
3.2 Konuşma Kodlamanın Temelleri	12
3.3 Konuşma Kodlama Uygulamaları.....	14
3.3.1 Kamu Telefon Şebekesi	14
3.3.2 Sayısal Hücreli Mobil Radyo Sistemleri	15
3.3.3 Mobil Uydu Haberleşmesi	16
3.3.4 Kişisel Haberleşme Sistemleri (<i>Personal Communication Systems-PCS</i>)	18
3.3.5 Çok Küçük Açıklıklı Terminal (VSAT) Şebekeleri.....	18
3.3.6 Askeri Haberleşme	19
3.4 Algoritma Seçiminde Önemli Unsurlar.....	20
3.4.1 Kalite ve Kapasite	21
3.4.2 Kodlama Gecikmesi	22
3.4.3 Bağışıklık (<i>Robustness</i>).....	22
3.4.4 Karmaşıklık ve Maliyet	23
3.4.5 Kaskat Bağlantı ve Yeniden Kodlama	24
3.4.6 Ses Bandında Veri İletimi.....	24

BÖLÜM 4 KODLAMA YÖNTEMLERİ VE STANDARTLARI..... 25

4.1 Kodlama Sistemlerinin Sınıflandırılması	25
4.2 Kodlama Süreci.....	29
4.2.1 Nicemleme	29
4.2.1.1 Doğrusal Nicemleme	30
4.2.1.2 Logaritmik Nicemleme	30
4.2.1.3 Düzgün Olmayan Nicemleme	32
4.2.1.4 Vektör Nicemleme.....	32
4.2.2 Zaman Bölgesinde Dalga Şekli Kodlama.....	36
4.2.2.1 Darbe Kod Modülasyonu (<i>Pulse Code Modulation-PCM</i>)	36
4.2.2.2 Delta Modülasyonu (<i>Delta Modulation-DM</i>).....	37
4.2.2.3 Ayırık Darbe Kod Modülasyonu (<i>Differerantial PCM-DPCM</i>)...38	
4.2.2.4 Adaptif Ayırık Darbe Kod Modülasyonu (<i>Adaptive DPCM-ADPCM</i>).....	40
4.2.3 Frekans Bölgesinde Dalga Şekli Kodlama	42
4.2.3.1 Adaptif Öngörülü Kodlama (<i>Adaptive Predictive Coding-APC</i>).....	42
4.2.3.2 Alt Band Kodlama (<i>Subband Coding</i>)	43
4.2.4 Kaynak Kodlayıcılar (<i>Source Coders</i>)	45
4.2.4.1 Kanal Ses Kodlayıcı (<i>Channel Vocoder</i>).....	46
4.2.4.2 Homomorfik Ses Kodlayıcı (<i>Homomorphic Vocoder</i>).....	48
4.2.4.3 Rezonans Ses Kodlayıcı (<i>Formant Vocoder</i>).....	49
4.2.4.4 Doğrusal Öngörülü Ses Kodlayıcı (<i>Lineer Predictive Vocoder</i>) ..50	
4.2.5 Hibrit Kodlayıcılar	51
4.2.5.1 Artık İşaret Uyarmalı Doğrusal Öngörülü Kodlayıcı (<i>RELP</i>)	54
4.2.5.2 Çoklu Darbe ile Uyarılan Doğrusal Öngörülü Kodlayıcı (<i>MPLPC</i>)	55
4.2.5.3 Düzenli Darbe ile Uyarılan Doğrusal Öngörülü Kodlayıcı (<i>RPLPC</i>)	57
4.2.5.4 Kod ile Uyarılan Doğrusal Öngörülü Kodlayıcı (<i>CELP</i>)	58
4.2.5.5 Karışık Uyarmalı Doğrusal Öngörülü Kodlama (<i>MELP</i>)	60
4.2.5.6 Sinüzoidal Modelleme (<i>STC</i>)	61
4.2.5.7 Çoklu-Band Uyarma (<i>MBE</i>).....	62
4.2.6 Hece (Fonetik) Kodlayıcılar	63

BÖLÜM 5 DÜŞÜK BİT HIZLI KONUŞMA KODLAMA ARAÇLARI..... 64

5.1 Konuşma İşareti	64
5.2 Kısa Süreli Spektrum Analizi	65
5.2.1 Pencerelemin Katkısı.....	66
5.2.2 <i>Fourier</i> Dönüşümü Yaklaşımı	68
5.3 Konuşma İşareti Spektrumu Zarf Öngörüsü.....	74
5.3.1 Maksimum Benzerlik Yöntemi ile Spektrum Zarfı Öngörüsü	75
5.3.1.1 Tam Kutuplu Modelleme ile Spektral Öngörü	79
5.3.1.2 Özyinelemeli Maksimum Benzerlik (<i>RML</i>) Algoritması.....	81
5.3.2 Zaman Bölgesinde Doğrusal Öngörü Analizi.....	83
5.3.2.1 Özilişki Yöntemi	85

5.3.2.2 LPC Modelinin Derecesi.....	88
5.4 Perde Öngörüsü.....	91
5.4.1 Frekans Bölgesinde Perde Öngörüsü	92
5.4.1.1 Harmonik Perde Öngörüsü	93
5.4.1.2 Spektrum Temelli Perde Öngörüsü	94
5.4.2 Zaman Bölgesinde Perde Öngörüsü.....	98
5.4.2.1 Özilişki Esaslı Perde Öngörüsü.....	98
5.4.3 Perde Pekleştirme.....	100
5.4.3.1 Yanlılık Giderme.....	101
5.4.3.2 Perde İzleme.....	103
5.4.3.3 Perde Astkat Kontrolü	105
5.4.3.4 Perde Öngörüsünde Duyarlık Artışı	106
BÖLÜM 6 SENTEZLEYEREK ANALİZ.....	108
6.1 Sentezleyerek Analiz Yaklaşımı	108
6.2 Genel AbS Kodlaması	110
6.2.1 Zamanla Değişen Filtreler	113
6.2.2 Algılama Esaslı Minimizasyon İşlemi.....	114
6.2.3 Uyarma İşareti.....	117
6.2.3.1 Kod Kitabıyla Uyarma.....	118
6.2.3.2 Kendinden Uyarma.....	119
6.2.3.3 Çoklu Darbeli ve Düzenli Uyarma.....	120
6.2.4 En Uygun Uyarma Dizisinin Tesbiti	122
6.2.5 AbS-LPC Yapılarının Özellikleri.....	127
6.2.5.1 CELP	127
6.2.5.2 SELP	130
6.2.5.3 MPLPC	130
6.2.5.4 RPELPC.....	134
6.3 Çoklu Darbeli LPC (MPLPC).....	135
6.3.1 En Uygun Genlik Uyarmalı MPLPC.....	138
6.3.2 Uzun Dönemli Öngörülü MPLPC.....	142
6.3.3 Darbe Yeri Kodlaması.....	146
6.3.4 Darbe Genlik Kodlaması	147
6.3.5 Ortak Darbe Genliği Nicemlemesi (JPAQ)	151
6.4 Kod ile Uyarılan Doğrusal Öngörülü Kodlama (CELP)	154
6.4.1 Kısa Dönemli Öngörü (STP)	156
6.4.2 Uzun Dönemli Öngörü	158
6.4.2.1 Açık Çevrim LTP	159
6.4.2.2 Kapalı Çevrim LTP	161
6.4.2.3 Kesirli Gecikmeli LTP.....	166
6.4.2.4 LTP Yöntemlerinin Performans Karşılaştırması:.....	171
6.4.3 İkincil Uyarmanın Hesaplanması.....	176
6.4.3.1 Standart (<i>Gauss</i> tipi) Uyarma:	177
6.4.3.2 Vektör Toplamı Uyarma (VSELP):	182
6.4.3.3 Adaptif Perdeli Karışık Uyarma (PAME)	187
6.4.4 Değişik İkincil Uyarmalı CELP'in Performansı.....	191
6.4.5 Ortak LTP ve İkincil Uyarma Hesabı.....	200

BÖLÜM 7 CELP KODLAYICILARIN GERÇEK ZAMAN UYGULAMALARI.....	204
7.1 Gerçek Zamanlı Süreçler	204
7.2 Sayısal İşaret İşlemciler.....	206
7.2.1 Sabit-Noktalı Uygulama	207
7.2.2 Kayan-Noktalı Uygulama.....	209
7.3 Gerçek Zamanlı Algoritmalar ve Yazılım Geliştirme.....	211
7.4 CELP Gerçekleme Yaklaşımları	214
7.4.1 Gerçeklenen 4.8 kbit/san CELP Kodlayıcının Yapısı	214
7.4.1.1 Giriş/Çıkış.....	216
7.4.1.2 Spektrum Analizi.....	217
7.4.1.3 Perde ve Kod Kitabı Arama.....	217
7.4.2 TMS320C50 Sabit Noktalı Sayısal İşaret İşlemci.....	219
7.4.3 TMS320C50 Esaslı DSK Donanımı.....	220
7.4.4 Etkin DSP Programlama Teknikleri.....	222
7.4.5 CELP Karmaşıklık İndirgeme Stratejileri	223
7.4.5.1 Konvolüsyonun Karmaşıklık İndirgemesi	224
7.4.5.2 Özilişkinin Karmaşıklık İndirgemesi.....	225
7.4.5.3 Çapraz-İlişkinin Karmaşıklık İndirgemesi.....	226
7.4.6 CELP Kodlayıcının AbS Çevrimi için Önerilen Geliştirmeler.....	228
7.4.6.1 Sentez Filtresi Parametrelerinin Hesabı.....	230
7.4.6.2 Uyarma Parametrelerinin Hesabı	233
BÖLÜM 8 SONUÇLAR VE ÖNERİLER	238
8.1 Giriş	238
8.2 Konuşma Kodlama	238
8.2.1 Konuşma Sıkıştırması için Süregelen Gereksinim	239
8.2.2 Konuşma Kodlamanın Günümüzdeki Yeterliliği	240
8.2.3 Teknolojik Hedefler	241
8.2.4 Gelecekteki Yönelim	241
8.2.4.1 Kodlama, İletişim ve Bilgisayar Ağları:	242
8.2.4.2 Konuşma Üretimi ve Algılama:.....	242
8.2.4.3 Sayısal İşaret İşleme:	243
8.3 Sonsöz.....	243
KAYNAKLAR.....	245
EKLER.....	249
ÖZGEÇMİŞ.....	253

ŞEKİL LİSTESİ

Şekil 2.1 İnsanın Ses Üretme Aracı	6
Şekil 2.2 İnsan Sesi Üretim Mekanizması Modeli	8
Şekil 2.3 Sesli Harf Zaman İşareti ve Spektrumu.....	9
Şekil 2.4 Sessiz Harf Zaman İşareti ve Spektrumu	9
Şekil 2.5 Erkek ve Kadın Ses Spektrumlarının Karşılaştırılması.....	10
Şekil 3.1 Çeşitli Karakterdeki Ses İşaretlerinin Band Genişlikleri.....	13
Şekil 3.2 Tipik Konuşma Kodlama (Sıkıştırma) Uygulamaları	14
Şekil 4.1 Konuşma Kodlama Sistemlerinin Sınıflandırılması.....	26
Şekil 4.2 Kodlama Yöntemlerinin Kalite Açısından Karşılaştırılması	28
Şekil 4.3 μ ve A yasasına ait transfer karakteristikleri	31
Şekil 4.4 İki boyutlu alanda bir vektör nicemleme örneği	34
Şekil 4.5 Delta Modülatörü ve Demodülatörü	37
Şekil 4.6 DPCM Modülatör ve Demodülatör Yapısı.....	39
Şekil 4.7 DPCM-AQF ve DPCM-AQB Modülasyon ve Demodülasyon.....	40
Şekil 4.8 Alt Band Kodlayıcı ve Kod Çözücü (iki band için).....	43
Şekil 4.9 Kanal Ses Kodlayıcı Blok Şeması (İki Kanal için).....	47
Şekil 4.10 Perde bileşeninin gecikmesini gösteren tipik bir konuşma kepstrumu	48
Şekil 4.11 Homomorfik Ses Kodlayıcı; (a) Verici, (b) Alıcı	49
Şekil 4.12 Rezonans Ses Kodlayıcı	50
Şekil 4.13 <i>AbS</i> Kodlayıcı ve Kodçözücü Modeli	52
Şekil 4.14 Konuşmanın ters filtrelenerek, LPC filtreden geçirilmesinin etkisi.....	54
Şekil 4.15 MPLPC Verici ve Alıcı	55
Şekil 4.16 CELP Sentezleyici Blok Diyagramı.....	59
Şekil 4.17 MELP Kodlayıcı Blok Şeması.....	61
Şekil 5.1 Pencere özellikleri: (a) impuls cevabı, (b) dikdörtgen pencere cevabı ve (c) Hamming penceresi cevabı	67
Şekil 5.2 Konuşma işaretinin, pencereleme yöntemiyle hemen hemen durağan çerçevelere bölünmesi.....	68

Şekil 5.3 Ünlü bir konuşma parçasının 27,5 msan'lik bir pencere kullanılarak yapılan spektral analizi (a ve c, zaman dalga şekilleri, b ve d bunlara karşılık gelen spektrumlar).....	72
Şekil 5.4 Ünsüz bir konuşma parçasının 27,5 msan'lik bir pencere kullanılarak yapılan spektrum analizi (a ve c, zaman dalga şekilleri, b ve d bunlara karşılık gelen spektrumlar).....	73
Şekil 5.5 Ünlü bir konuşma parçasının 5 msan'lik bir pencere kullanılarak yapılan spektrum analizi (a ve c, zaman dalga şekilleri, b ve d bunlara karşılık gelen spektrumlar).....	73
Şekil 5.6 Maksimum benzerlik spektrum eşleme kistası	79
Şekil 5.7 Özilişki ve özyinelemeli maksimum benzerlik yöntemleriyle elde edilen konuşma spektrumları ve bunların zarfları : (a) ünlü konuşma için, (b) ünsüz konuşma için, (c) normalize hata	88
Şekil 5.8 Çeşitli LPC modeli dereceleri için konuşma spektrumu ve LPC spektral zarfları.....	89
Şekil 5.9 LPC modeli dizisinin bir işlevi olarak normalize doğrusal öngörü hatası ..	90
Şekil 5.10 Harmonik perde öngörüsü metodu.....	94
Şekil 5.11 Spektrum temelli perde öngörü tekniğinin blok şeması	95
Şekil 5.12 Optimum perde frekansı için orijinal ve yeniden oluşturulmuş spektrumlar.....	97
Şekil 5.13 Normalize ortalama karesel hata ile perde periyodu ilişkisi.....	97
Şekil 5.14 21. kademe FIR filtresinin frekans cevabı.....	100
Şekil 5.15 Perde öngörüsü için gerekli yanlılık düzeltmesi	102
Şekil 5.16 Perde öngörüsü için yanlılık düzeltmenin etkisi	102
Şekil 5.17 Adaptif eşik fonksiyonu.....	106
Şekil 5.18 Bir çerçeveden diğerine perde periyodu değişimi.....	107
Şekil 6.1 AbS kapalı çevrim analizi blok yapısı.....	110
Şekil 6.2 AbS-LPC yönteminin blok yapısı	112
Şekil 6.3 Tipik ağırlıklandırma filtresi spektrumları ve orijinal konuşma zarfıyla karşılaştırılması.....	115
Şekil 6.4 AbS-LPC kodlama yapısının değiştirilmiş kısmı (ağırlıklandırma filtresi, hata minimizasyon yordamının iki koluna taşınmış).....	117

Şekil 6.5 AbS-LPC kodlayıcının değişik uyarma tiplerini gösteren genel blok yapısı.....	118
Şekil 6.6 Tipik bir RPELPC darbe pozisyonu yapısı.....	122
Şekil 6.7 CELP için kod kitabı indisine ilişkin hata grafiği (ayırddedici vektör).....	127
Şekil 6.8 Orijinal ile karşılaştırılmış sentezlenmiş en iyi kod kitabı girişleri (ayırddedici vektör).....	128
Şekil 6.9 CELP için kod kitabı indisine ilişkin hata grafiği (daha az ayırddedici vektör).....	128
Şekil 6.10 Orijinal ile karşılaştırılmış sentezlenmiş en iyi kod kitabı girişleri (daha az ayırddedici vektör).....	129
Şekil 6.11 SELP için LTP1 ve LTP2 gecikmelerinin dağılımı	131
Şekil 6.12 MPLPC darbe genliklerinin dağılım grafiği	132
Şekil 6.13 80 örneklik alt çerçeve boyulu MPLPC darbe konumlarının dağılım grafiği.....	134
Şekil 6.14 Değişik yapıdaki RPELPC kodlayıcıların işaret gürültü oranları.....	135
Şekil 6.15 Basit bir MPLPC kodlayıcının blok diyagramı.....	136
Şekil 6.16 MPLPC kodlayıcısı dalga şekilleri (a) orijinal konuşma, (b) çoklu darbe uyarması, (c) sentezlenmiş konuşma ve (d) hata işareti.	138
Şekil 6.17 Bölütlenmiş işaret gürültü oranı ile darbe sayısının değişimi. her darbeden sonra genlik yeniden optimize ediliyor, (b) tüm darbelerden sonra bir kere genlik yeniden optimize ediliyor ve (c) genliğin yeniden optimizasyonu olmaksızın ardışıl MPLPC.....	141
Şekil 6.18 LTP'li MPLPC AbS yordamı blok diyagramı.....	143
Şekil 6.19 MPLPC'nin (a) LTP'li ve (b) LTP'siz performansı	145
Şekil 6.20 Uyarma dalga biçimleri (a) LTP uyarması ve (b) Çoklu darbeli uyarma.....	145
Şekil 6.21 (a) Bağımsız ve (b) bileşik darbe pozisyonu kodlamasının karşılaştırılması.....	147
Şekil 6.22 İlk genlik normalizasyonu için dağılım grafikleri.....	149
Şekil 6.23 İlk darbe dışında normalize edilmiş darbe genliklerinin dağılım grafikleri.....	150

Şekil 6.24 (a) yüksek enerjili (b) alçak enerjili konuşma bölgeleri için darbe hesaplama sırasına göre darbe genlikleri	152
Şekil 6.25 λ değerinin histogramı	153
Şekil 6.26 Hesap sırasına göre orijinal ve ortak nicemlenmiş darbe genlikleri	153
Şekil 6.27 Standard CELP algoritması blok diyagramı	156
Şekil 6.28 (a) yarım çerçeve aradeğerleme gecikmeli ve (b) gecikmesiz LPC analiz pencereleri.....	158
Şekil 6.29 Değiştirilmiş açık devre (<i>Modified OLM</i>) LTP analizi blok diyagramı.	159
Şekil 6.30 Kapalı Çevrim (<i>Closed Loop-CL</i>) LTP analizinin blok diyagramı.....	162
Şekil 6.31 Optimizasyon aralığından daha küçük gecikmeler için perde tekrarını gösteren blok diyagram	166
Şekil 6.32 Aradeğerlemenin gerçekleşmesi için uygulanan çok fazlı yapı	168
Şekil 6.33 $N = 65$ iken alçak geçiren aradeğerleme filtresinin frekans cevabı	170
Şekil 6.34 Toplam SNR'ye katkıda bulunan CELP alt sistemlerinin açınımı	172
Şekil 6.35 MOL3-LTP yapısı için SNR değerlerinin açınımı.....	174
Şekil 6.36 HMOL3-LTP yapısı için SNR değerlerinin açınımı	174
Şekil 6.37 CL3-LTP yapısı için SNR değerlerinin açınımı.....	175
Şekil 6.38 Tipik bir üst üste binen kod kitabı yapısı.....	179
Şekil 6.39 Tek kaydırmalı üst üste binen kod kitabı araması sistemiği	181
Şekil 6.40 VSELP kodlayıcı blok diyagramı	187
Şekil 6.41 (a) orijinal, (b) üst üste binen merkezi konuşma kesmeli Gauss uyarmalı CELP çıktısı ve (c) PAME uyarmalı CELP çıktılarının konuşma dalga biçimleri	190
Şekil 6.42 PAME ve spektrum şekillendirmeli CELP'in blok diyagramı.....	192
Şekil 6.43 (a) orijinal ve (b) şekillenmiş PAME uyarma dalga biçimlerinin grafiği.....	192
Şekil 6.44- Tipik CELP dalga biçimleri. (a) Kod kitabı katkısı, (b) LTP (perde) katkısı, (c) STP belleği katkısı, (d) toplam çıktı, (e) orijinal konuşma ve (f) kalan hata.....	198
Şekil 6.45 Bir konuşma alt çerçevesi için LTP (perde) ve kod kitabı ilişkisi	199
Şekil 7.1 Sabit ve Kayan Noktalı DSP'lerin Dinamik Aralığı	207
Şekil 7.2 CELP Kod Çözücü	215
Şekil 7.3 CELP Kodlayıcı	216

Şekil 7.4 Kod Kitabı Arama Blok Diyagramı217

Şekil 7.5 Üstüste Binmiş Kod Kitabı Yapısı218

Şekil 7.6 TMS 320C50 DSK Blok Şema221

Şekil 7.7 DSK Yazılım Geliştirme Akışı222

Şekil 7.8 CELP Kodlayıcı Prensip Şeması.....229

Şekil 7.9 Güncellemenin SNR'ye Etkisi233

Şekil 8.1 Çeşitli Bit Hızları için Konuşma Kalitesi MOS Değerleri240



TABLO LİSTESİ

Tablo 3.1 Konuşma Kodlama Uygulamaları ve Şebeke Gereksinimleri	21
Tablo 4.1 Konuşma Kodlama Uygulamaları	27
Tablo 6.1 Aradeğerleme filtresinin dürtü cevabı	169
Tablo 6.2 LTP karşılaştırma testi için CELP kodlayıcısı yapısı	172
Tablo 6.3 Değişik LTP'lerin SegSNR (bölütlenmiş SNR) değerlerinin açılımı	173
Tablo 6.4 Karşılaştırma testi için CELP kodlayıcı parametreleri tanımı	180
Tablo 6.5 Dört standard kod kitabının performansları	180
Tablo 6.6 Yaklaşık 8 kbit/san'lik CELP yapılarının bit ataması	193
Tablo 6.7 1 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı	193
Tablo 6.8 5 msan'de güncellenmiş 3 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı.....	193
Tablo 6.9 10 msn'de güncelleştirilmiş 1 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı... ..	194
Tablo 6.10 Yaklaşık 5 kbit/san hızında CELP yapılarının bit ataması	194
Tablo 6.11 1 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı ..	194
Tablo 6.12 3 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı ..	195
Tablo 6.13 4.7 kbit/san'de örnek bir CELP konfigürasyonu	197
Tablo 6.14 Uyarma vektör kazançları için düzgün olmayan 5 bit nicemleyici	197
Tablo 6.15 LTP kazançları için düzenli olmayan 5 bit nicemleyici	199
Tablo 7.1 Sabit Noktalı Sayısal İşaret İşlemci Örnekleri.....	208
Tablo 7.2 Kayan Noktalı Sayısal İşaret İşlemci Örnekleri	210
Tablo 7.3 4.8 kbit/san CELP Parametreleri	216
Tablo 7.4 Çeşitli Kod Kitabı Arama Yöntemlerinin Performansı ve Karmaşıklığı.....	237

ÖZET

Günümüzde insanoğlu iletim ortamından gittikçe daha fazla kapasite beklemektedir. Teknolojik yenilikler ile ne kadar büyük bandgenişlikleri sağlansa da, iletilecek işaretlerin bandgenişlikleri de üstel olarak artmaktadır. Bu nedenle sayısal iletişimin kanal kapasitesinin artması gereksinimi, haberleşme kanalının etkin olarak kullanılması problemini başlıca araştırma konularından biri haline getirmiştir.

Böylelikle sayısallaştırılmış konuşma işaretlerinin iletimi ve kaydı için etkin kodlayıcılar oluşturulması gereksinimi doğmuştur (örneğin, mobil haberleşme, uydu haberleşmesi, ses kaydetme ve gönderme sistemleri). Günümüzde, konuşma kalitesini düşürmeden hızı 4,8 kbit/san ve altına çekmek için çeşitli teknikler uygulanmaktadır. Konuşma işaretinde bulunan mevcut fazlalıkları atarak düşük bit hızlarında kodlanma mümkün olmaktadır. Ayrıca, insanın işitme sistemi değişik frekanslardaki bozulmalara karşı eşit derecede hassas değildir ve sınırlı bir dinamik aralığı vardır. Uygulanan konuşma kodlama teknikleri, bit hızını düşürmek için konuşmanın hem üretimi, hem de algılanmasıyla ilgili özelliklerden yararlanır. Bütün bu karmaşık algoritmaların uygulanması, bu yöntemlerin gerçek zamanlı uygulamalarını mümkün kılan VLSI/DSP teknolojisindeki hızlı ilerlemeler sayesinde olmuştur.

CELP yöntemi 1985 yılında *Schroeder* ve *Atal* tarafından ilk ortaya atıldığında gerçekleştirme için 333 MIPS'lik bir işlem kapasitesi, yani o zamanki Cray-1 bilgisayarında 1 sn. konuşma işaretini işlemek için 125 sn. CPU süresi gerektirmekteydi. Buradan hareketle bu tezde düşük bit hızlarında konuşma kodlama teknikleri ve sayısal işaret işlemciler (DSP) ile gerçekleştirme yöntemleri araştırılmış; bu tekniklerden hareketle, 4.8 kbit/san civarındaki bit hızları için DSP tabanlı gerçekleştirmeler yapılmıştır. Çalışmada daha düşük hızlara inmek değil eşdeğer hızlarda kaynakları (DSP imkanlarını) daha verimli kullanıp, daha kaliteli kodlayıcılar tasarlama amacı güdülmüştür. Bu amaçla CELP kodlayıcılarının DSP tabanında gerçekleştirilmesi için gerekli gerçek zaman şartları, sabit noktalı ve kayan noktalı yaklaşımlar irdelenmiştir. Tipik bir CELP konuşma kodlama algoritmasında işlem süresinin çoğu kapalı çevrim LTP ve kod kitabı arama için harcanmaktadır. Bu işlemlerde oluşan karmaşıklık konvolüsyon, çaprazilişki ve özilişki hesaplamaları sonucunda oluşmaktadır. Bu nedenle bu işlemlerin daha etkin yapılabilmesi için sadeleştirme yöntemleri önerilmiştir. Ayrıca kodlayıcının sentezleyerek analiz çevrimi incelenmiş ve uyarma parametreleri belirlendikten sonra kısa dönemli sentez filtresi parametrelerinin güncellenmesi için yöntemler araştırılmıştır. Gösterilmiştir ki sentez filtresi parametrelerini bandgenişliği genişletme (*bandwidth expansion*) ve aradeğerleme (*interpolation*) yaklaşımlarında olduğu gibi güncellemek performansta önemli gelişmeler sağlamaktadır. Bundan başka bütün önerilen geliştirmelerin kullanıldığı bir CELP kodlayıcısı, yüksek seviyeli yazılım dili C kullanılarak gerçekleştirilmiş ve sabit noktalı bir sayısal işaret işlemciye uygulanmıştır. Elde edilen sonuçların sayısal bir dahili santralin söz bellek (voice mail) modülünde kullanılması hedeflenmektedir.

LOW BIT RATE SPEECH CODING AND APPLICATIONS

In an age where the word gigabit became common when talking about channel or disk capacity, the aim of compression or low bit rate speech coding is not clear to everyone and one needs to justify it by describing the myriad of new applications demanding less and less bits per second and the rapidly expanding corpora.

Speech is produced when air is forced from the lungs through the vocal cords and along the vocal tract. The vocal tract extends from the opening in the vocal cords (called the glottis) to the mouth, and in an average man is about 17 cm long. It introduces short-term correlations (of the order of 1 ms) into the speech signal, and can be thought of as a filter with broad resonances called formants. The frequencies of these formants are controlled by varying the shape of the tract, for example by moving the position of the tongue. An important part of many speech codecs is the modelling of the vocal tract as a short term filter. As the shape of the vocal tract varies relatively slowly, the transfer function of its modelling filter needs to be updated only relatively infrequently (typically every 20 ms or so).

The vocal tract filter is excited by air forced into it through the vocal cords. Speech sounds can be broken into three classes depending on their mode of excitation:

- Voiced sounds are produced when the vocal cords vibrate open and closed, thus interrupting the flow of air from the lungs to the vocal tract and producing quasi-periodic pulses of air as the excitation. The rate of the opening and closing gives the pitch of the sound. This can be adjusted by varying the shape of, and the tension in, the vocal cords, and the pressure of the air behind them. Voiced sounds show a high degree of periodicity at the pitch period, which is typically between 2 and 20 ms.
- Unvoiced sounds result when the excitation is a noise-like turbulence produced by forcing air at high velocities through a constriction in the vocal tract while the glottis is held open. Such sounds show little long-term periodicity, although short-term correlations due to the vocal tract are still present.
- Plosive sounds result when a complete closure is made in the vocal tract, and air pressure is built up behind this closure and released suddenly.

Some sounds cannot be considered to fall into any one of the three classes above, but are a mixture. For example voiced fricatives result when both vocal cord vibration and a constriction in the vocal tract are present.

Although there are many possible speech sounds which can be produced, the shape of the vocal tract and its mode of excitation change relatively slowly, and so speech can be considered to be quasi-stationary over short periods of time (of the order of 20

ms). Speech signals show a high degree of predictability, due sometimes to the quasi-periodic vibrations of the vocal cords and also due to the resonances of the vocal tract. Speech coders attempt to exploit this predictability in order to reduce the data rate necessary for good quality voice transmission.

Commonly Used Speech Codecs

Coding algorithms seek to minimize the bit rate in the digital representation of a signal without an objectionable loss of signal quality in the process. High quality is attained at low bit rates by exploiting signal redundancy as well as the knowledge that certain types of coding distortion are imperceptible because they are masked by the signal. The models of signal redundancy and distortion masking are becoming increasingly more sophisticated, leading to continuing improvements in the quality of low bit rate signals. Here its briefly discussed the main speech coding techniques which are used today, and those which may be used in the future. In order to simplify the description of speech codecs they are often broadly divided into three classes - waveform codecs, source codecs and hybrid codecs. Typically waveform codecs are used at high bit rates, and give very good quality speech. Source codecs operate at very low bit rates, but tend to produce speech which sounds synthetic. Hybrid codecs use techniques from both source and waveform coding, and give good quality speech a intermediate bit rates.

Waveform Codecs:

Waveform codecs attempt, without using any knowledge of how the signal to be coded was generated, to produce a reconstructed signal whose waveform is as close as possible to the original. This means that in theory they should be signal independent and work well with non-speech signals. Generally they are low complexity codecs which produce high quality speech at rates above about 16 kbits/s. When the data rate is lowered below this level the reconstructed speech quality that can be obtained degrades rapidly.

The simplest form of waveform coding is Pulse Code Modulation (PCM), which merely involves sampling and quantizing the input waveform. Narrow-band speech is typically band-limited to 4 kHz and sampled at 8 kHz. If linear quantization is used then to give good quality speech around twelve bits per sample are needed, giving a bit rate of 96 kbits/s. This bit rate can be reduced by using non-uniform quantization of the samples. In speech coding an approximation to a logarithmic quantizer is often used. Such quantizers give a signal to noise ratio which is almost constant over a wide range of input levels, and at a rate of eight bits/sample (or 64 kbits/s) give a reconstructed signal which is almost indistinguishable from the original. Such logarithmic quantizers were standardised in the 1960's, and are still widely used today. In America u-law companding is the standard, while in Europe the slightly different A-law compression is used. They have the advantages of low complexity and delay with high quality reproduced speech, but require a relatively high bit rate and have a high susceptibility to channel errors.

A commonly used technique in speech coding is to attempt to predict the value of the next sample from the previous samples. It is possible to do this because of the correlations present in speech samples due to the effects of the vocal tract and the vibrations of the vocal cords, as discussed earlier. If the predictions are effective then the error signal between the predicted samples and the actual speech samples will have a lower variance than the original speech samples. Therefore we should be able to quantize this error signal with fewer bits than the original speech signal. This is the basis of Differential Pulse Code Modulation (DPCM) schemes - they quantize the difference between the original and predicted signals.

The results from such codecs can be improved if the predictor and quantizer are made adaptive so that they change to match the characteristics of the speech being coded. This leads to Adaptive Differential PCM (ADPCM) codecs. In the mid 1980's the CCITT standardised a ADPCM codec operating at 32 kbits/s, which gave speech quality that was very similar to the 64 kbits/s PCM codecs. Later ADPCM codecs operating at 16,24 and 40 kbits/s were also standardised. The waveform codecs described above all code speech with an entirely time domain approach. Frequency domain approaches are also possible, and have certain advantages. For example in Sub-Band Coding (SBC) the input speech is split into a number of frequency bands, or sub-bands, and each is coded independently using for example an ADPCM like coder. At the receiver the sub-band signals are decoded and recombined to give the reconstructed speech signal. The advantages of doing this come from the fact that the noise in each sub-band is dependent only on the coding used in that sub-band. Therefore we can allocate more bits to perceptually important sub-bands so that the noise in these frequency regions is low, while in other sub-bands we may be content to allow a high coding noise because noise at these frequencies is less perceptually important. Adaptive bit allocation schemes may be used to further exploit these ideas. Sub-band codecs tend to produce communications to toll quality speech in the range 16-32 kbits/s. Due to the filtering necessary to split the speech into sub-bands they are more complex than simple DPCM coders, and introduce more coding delay. However the complexity and delay are still relatively low when compared to most hybrid codecs.

Another frequency domain waveform coding technique is Adaptive Transform Coding (ATC), which uses a fast transformation (such as the discrete cosine transformation) to split blocks of the speech signal into a large numbers of frequency bands. The number of bits used to code each transformation coefficient is adapted depending on the spectral properties of the speech, and toll quality reproduced speech can be achieved at bit rates as low as 16 kbits/s.

Source Codecs:

Source coders operate using a model of how the source was generated, and attempt to extract, from the signal being coded, the parameters of the model. It is these model parameters which are transmitted to the decoder. Source coders for speech are called vocoders, and work as follows. The vocal tract is represented as a time-varying filter and is excited with either a white noise source, for unvoiced speech segments, or a train of pulses separated by the pitch period for voiced speech. Therefore the information which must be sent to the decoder is the filter specification, a

voiced/unvoiced flag, the necessary variance of the excitation signal, and the pitch period for voiced speech. This is updated every 10-20 ms to follow the non-stationary nature of speech.

The model parameters can be determined by the encoder in a number of different ways, using either time or frequency domain techniques. Also the information can be coded for transmission in various different ways. Vocoder tend to operate at around 2.4 kbits/s or below, and produce speech which although intelligible is far from natural sounding. Increasing the bit rate much beyond 2.4 kbits/s is not worthwhile because of the inbuilt limitation in the coder's performance due to the simplified model of speech production used. The main use of vocoders has been in military applications where natural sounding speech is not as important as a very low bit rate to allow heavy protection and encryption.

Hybrid Codecs:

Hybrid codecs attempt to fill the gap between waveform and source codecs. As described above waveform coders are capable of providing good quality speech at bit rates down to about 16 kbits/s, but are of limited use at rates below this. Vocoder on the other hand can provide intelligible speech at 2.4 kbits/s and below, but cannot provide natural sounding speech at any bit rate. Although other forms of hybrid codecs exist, the most successful and commonly used are time domain Analysis-by-Synthesis (AbS) codecs. Such coders use the same linear prediction filter model of the vocal tract as found in LPC vocoder. However instead of applying a simple two-state, voiced/unvoiced, model to find the necessary input to this filter, the excitation signal is chosen by attempting to match the reconstructed speech waveform as closely as possible to the original speech waveform. AbS codecs were first introduced in 1982 by *Atal* and *Remde* with what was to become known as the Multi-Pulse Excited (MPE) codec. Later the Regular-Pulse Excited (RPE), and the Code-Excited Linear Predictive (CELP) codecs were introduced. These coders will be discussed briefly here.

AbS codecs work by splitting the input speech to be coded into frames, typically about 20 ms long. For each frame parameters are determined for a synthesis filter, and then the excitation to this filter is determined. This is done by finding the excitation signal which when passed into the given synthesis filter minimises the error between the input speech and the reconstructed speech. Thus the name Analysis-by-Synthesis - the encoder analyses the input speech by synthesising many different approximations to it. Finally for each frame the encoder transmits information representing the synthesis filter parameters and the excitation to the decoder, and at the decoder the given excitation is passed through the synthesis filter to give the reconstructed speech. The synthesis filter is usually an all pole, short-term, linear filter. This filter is intended to model the correlations introduced into the speech by the action of the vocal tract.

The synthesis filter may also include a pitch filter to model the long-term periodicities present in voiced speech. Alternatively these long-term periodicities may be exploited by including an adaptive codebook in the excitation generator so that the excitation signal includes a component, where the estimated pitch period is

mentioned. Generally MPE and RPE codecs will work without a pitch filter, although their performance will be improved if one is included. For CELP codecs however a pitch filter is extremely important, for reasons discussed below.

The error weighting block is used to shape the spectrum of the error signal in order to reduce the subjective loudness of this error. This is possible because the error signal in frequency regions where the speech has high energy will be at least partially masked by the speech. The weighting filter emphasises the noise in the frequency regions where the speech content is low. Thus minimising the weighted error concentrates the energy of the error signal in frequency regions where the speech has high energy. Therefore the error signal will be at least partially masked by the speech, and so its subjective importance will be reduced. Such weighting is found to produce a significant improvement in the subjective quality of the reconstructed speech for AbS codecs.

The distinguishing feature of AbS codecs is how the excitation waveform $u(n)$ for the synthesis filter is chosen. Conceptually every possible waveform is passed through the filter to see what reconstructed speech signal this excitation would produce. The excitation which gives the minimum weighted error between the original and the reconstructed speech is then chosen by the encoder and used to drive the synthesis filter at the decoder. It is this 'closed-loop' determination of the excitation which allows AbS codecs to produce good quality speech at low bit rates. However the numerical complexity involved in passing every possible excitation signal through the synthesis filter is huge. Usually some means of reducing this complexity, without compromising the performance of the codec too badly, must be found.

The differences between MPE, RPE and CELP codecs arise from the representation of the excitation signal used. In multi-pulse codecs the excitation signal is given by a fixed number of non-zero pulses for every frame of speech. The positions of these non-zero pulses within the frame, and their amplitudes, must be determined by the encoder and transmitted to the decoder. In theory it would be possible to find the very best values for all the pulse positions and amplitudes, but this is not practical due to the excessive complexity it would entail. In practice some sub-optimal method of finding the pulse positions and amplitudes must be used. Typically about 4 pulses per 5 ms are used, and this leads to good quality reconstructed speech at a bit-rate of around 10 kbits/s.

Like the MPE codec the Regular Pulse Excited (RPE) codec uses a number of non-zero pulses to give the excitation signal. However in RPE codecs the pulses are regularly spaced at some fixed interval, and the encoder needs only to determine the position of the first pulse and the amplitude of all the pulses. Therefore less information needs to be transmitted about pulse positions, and so for a given bit rate the RPE codec can use many more non-zero pulses than MPE codecs. For example at a bit rate of about 10 kbits/s around 10 pulses per 5 ms can be used in RPE codecs, compared to 4 pulses for MPE codecs. This allows RPE codecs to give slightly better quality reconstructed speech quality than MPE codecs. However they also tend to be more complex. The pan-European GSM mobile telephone system uses a simplified RPE codec, with long-term prediction, operating at 13 kbits/s to provide toll quality speech.

Although MPE and RPE codecs can provide good quality speech at rates of around 10 kbits/s and higher, they are not suitable for rates much below this. This is due to the large amount of information that must be transmitted about the excitation pulses' positions and amplitudes. If we attempt to reduce the bit rate by using fewer pulses, or coarsely quantizing their amplitudes, the reconstructed speech quality deteriorates rapidly. Currently the most commonly used algorithm for producing good quality speech at rates below 10 kbits/s is Code Excited Linear Prediction (CELP). This approach was proposed by *Schroeder* and *Atal* in 1985, and differs from MPE and RPE in that the excitation signal is effectively vector quantized. The excitation is given by an entry from a large vector quantizer codebook, and a gain term to control its power. Typically the codebook index is represented with about 10 bits (to give a codebook size of 1024 entries) and the gain is coded with about 5 bits. Thus the bit rate necessary to transmit the excitation information is greatly reduced - around 15 bits compared to the 47 bits used for example in the GSM RPE codec.

Originally the codebook used in CELP codecs contained white Gaussian sequences. This was because it was assumed that long and short-term predictors would be able to remove nearly all the redundancy from the speech signal to produce a random noise-like residual. Also it was shown that the short-term probability density function (pdf) of this residual was nearly Gaussian. *Schroeder* and *Atal* found that using such a codebook to produce the excitation for long and short-term synthesis filters could produce high quality speech. However to choose which codebook entry to use in an analysis-by-synthesis procedure meant that every excitation sequence had to be passed through the synthesis filters to see how close the reconstructed speech it produced would be to the original. This meant the complexity of the original CELP codec was much too high for it to be implemented in real-time - it took 125 seconds of Cray-1 CPU time to process 1 second of the speech signal. Since 1985 much work on reducing the complexity of CELP codecs, mainly through altering the structure of the codebook, has been done. Also large advances have been made with the speed possible from DSP chips, so that now it is relatively easy to implement a real-time CELP codec on a single, low cost, DSP chip. Several important speech coding standards have been defined based on the CELP principle, for example the American Department of Defence (DoD) 4.8 kbits/s codec, and the CCITT low-delay 16 kbits/s codec.

The CELP coding principle has been very successful in producing communications quality speech at bit rates between 4.8 and 16 kbits/s. The CCITT standard 16 kbits/s codec produces speech which is almost indistinguishable from 64 kbits/s log-PCM coded speech, while the DoD 4.8 kbits/s codec gives good communications quality speech. Recently much research has been done on codecs operation below 4.8 kbits/s, with the aim being to produce a codec at 2.4 or 3.6 kbits/s with speech quality equivalent to the 4.8 kbits/s DoD CELP.

The CELP codec structure can be improved and used at rates below 4.8 kbits/s by classifying speech segments into one of a number of types (for example voiced, unvoiced and transition frames). The different speech segment types are then coded differently with a specially designed encoder for each type. For example for unvoiced frames the encoder will not use any long-term prediction, whereas for

voiced frames such prediction is vital but the fixed codebook may be less important. Such class-dependent codecs have been shown to be capable of producing reasonable quality speech at rates down to 2.4 kbits/s. Multi-Band Excitation (MBE) codecs work by declaring some regions in the frequency domain as voiced and others as unvoiced. They transmit for each frame a pitch period, spectral magnitude and phase information, and voiced/unvoiced decisions for the harmonics of the fundamental frequency. Originally it was shown that such a structure was capable of producing good quality speech at 8 kbits/s, and since then this rate has been significantly reduced. Finally *Kleijn* has suggested an approach for coding voiced segments of speech called Prototype Waveform Interpolation (PWI). This works by sending information about a single pitch cycle every 20-30 ms, and using interpolation to reproduce a smoothly varying quasi-periodic waveform for voiced speech segments. Excellent quality reproduced speech can be obtained for voiced speech at rates as low as 3 kbits/s. Such a codec can be combined with a CELP type codec for the unvoiced segments to give good quality speech at rates below 4 kbits/s.

In recent years other types of speech coders are also investigated. One of the successful outcome of these investigations is the Mixed Excitation Linear Predictive (MELP) coder, which is the new 2400 bps Federal Standard speech coder in the US. It was selected by the United States Department of Defense Digital Voice Processing Consortium (DDVPC) after a multi-year extensive test program. The selection test concentrated on four areas: intelligibility, voice quality, talker recognizability, and communicability. The selection criteria also included hardware parameters such as processing power, memory usage and delay. MELP is robust in difficult background noise environments such as those frequently encountered in commercial and military communication systems. It is very efficient in its computational requirements. This translates into relative low power consumption, an important consideration for portable systems.

The Continuing Need for Speech Compression

The capability of speech compression has been central to the technologies of robust long-distance communication, high-quality speech storage, and message encryption. Compression continues to be a key technology in communications in spite of the promise of optical transmission media of relatively unlimited bandwidth. This is because of our continued and, in fact, increasing need to use band-limited media such as radio and satellite links, and bit-rate-limited storage media such as CD-ROMs and silicon memories. Storage and archival of large volumes of spoken information makes speech compression essential even in the context of significant increases in the capacity of optical and solid-state memories.

Low bit-rate speech technology is a key factor in meeting the increasing demand for new digital wireless communication services. Impressive progress has been made during recent years in coding speech with high quality at low bit rates and at low cost. Only ten years ago, high quality speech could not be produced at bit rates below 24 kbps. Today, we can offer high quality at 8 kbps, making this the standard rate for the new digital cellular service in North America. Using new techniques for channel coding and equalization, it is possible to transmit the 8 kbps speech in a robust fashion over the mobile radio channel, in spite of channel noise, signal fading and

intersymbol interference. The present research is focussed on meeting the critical need for high quality speech transmission over digital cellular channels at 4 kbps. Research on properly coordinated source and channel coding is needed to realize a good solution to this problem.

Wireless communication channels suffer from multipath interference producing error rates in excess of 10%. The challenge for speech research is to produce digital speech that can be transmitted with high quality over communication networks in the presence of up to 10% channel errors. A speech coder operating at 2 kbps will provide enough bits for correcting such channel errors, assuming a total transmission rate on the order of 4 to 8 kbps.

The bit rate of 2 kbps has an attractive implication for voice storage as well. At this bit rate, more than 2 hours of continuous speech can be stored on a single 16 Mbit memory chip, allowing sophisticated voice messaging services on personal communication terminals, and extending significantly the capabilities of digital answering machines. Fundamental advances in our understanding of speech production and perception are needed to achieve high quality speech at 2 kbps.

Applications of wideband speech coding include high quality audioconferencing with 7 kHz-bandwidth speech at bit rates on the order of 16 to 32 kbps, and high-quality stereoconferencing and dual-language programming over a basic ISDN link. Finally, the compression of a 20 kHz-bandwidth to rates on the order of 64 kbps will create new opportunities in audio transmission and networking, electronic publishing, travel and guidance, teleteaching, multilocation games, multimedia memos, and database storage.

The Dimensions of Performance in Speech Compression

Speech coders attempt to minimize the bit rate for transmission or storage of the signal while maintaining required levels of speech quality, communication delay, and complexity of implementation (power consumption). We will now provide brief descriptions of the above parameters of performance, with particular reference to speech.

Speech Quality: Speech quality is usually evaluated on a five-point scale, known as the mean-opinion score (MOS) scale, in speech quality testing an average over a large number of speech data, speakers, and listeners. The five points of quality are: bad, poor, fair, good, and excellent. Quality scores of 3.5 or higher generally imply high levels of intelligibility, speaker recognition and naturalness.

Communication Delay: Speech coders often process speech in blocks and such processing introduces communication delay. Depending on the application, the permissible total delay could be as low as 1 msec, as in network telephony, or as high as 500 msec, as in video telephony. Communication delay is irrelevant for one-way communication, such as in voice mail.

Complexity: The complexity of a coding algorithm is the processing effort required to implement the algorithm, and it is typically measured in terms of arithmetic capability and memory requirement, or equivalently in terms of cost. A large complexity can result in high power consumption in the hardware.

Current Capabilities in Speech Coding

PCM (pulse-code modulation) is the simplest coding system, a memoryless quantizer, and provides essentially transparent coding of telephone speech at 64 kbps. With a simple adaptive predictor, adaptive differential PCM (ADPCM) provides high-quality speech at 32 kbps. The speech quality is slightly inferior to that of 64 kbps PCM, although the telephone handset receiver tends to minimize the difference. ADPCM at 32 kbps is widely used for expanding the number of speech channels by a factor of two, particularly in private networks and international circuits. It is also the basis of low-complexity speech coding in several proposals for personal communication networks, including CT2 (Europe), UDPCS (USA) and Personal Handyphone (Japan)

For rates of 16 kbps and lower, high speech quality is achieved by using more complex adaptive prediction, such as linear predictive coding (LPC) and pitch prediction, and by exploiting auditory masking and the underlying perceptual limitations of the ear. Important examples of such coders are multi-pulse excitation, regular-pulse excitation, and code-excited linear prediction (CELP) coders. The CELP algorithm combines the high quality potential of waveform coding with the compression efficiency of model-based vocoders. At present, the CELP technique is the technology of choice for coding speech at bit rates of 16 kbps and lower. At 16 kbps, a low-delay CELP (LD-CELP) algorithm provides both high quality, close to PCM, and low communication delay and has been accepted as an international standard for transmission of speech over telephone networks.

At 8 kbps, which is the bit rate chosen for first-generation digital cellular telephony in North America, speech quality is good, although significantly lower than that of the 64 kbps PCM speech. Both North American and Japanese first generation digital standards are based on the CELP technique. The first European digital cellular standard is based on regular-pulse excitation algorithm at 13.2 kbps.

The rate of 4.8 kbps is an important data rate because it can be transmitted over most local telephone lines in the United States. A version of CELP operating at 4.8 kbps has been chosen as a United States standard for secure voice communication. The other such standard uses an LPC vocoder operating at 2.4 kbps. The LPC vocoder produces intelligible speech but the speech quality is not natural.

The present research is focussed on meeting the critical need for high quality speech transmission over digital cellular channels at 4 and 8 kbps. Low bit rate speech coders are fairly complex, but the advances in VLSI and the availability of digital signal processors have made possible the implementation of both encoder and decoder on a single chip.

Technology Targets

Given that there is no rigorous mathematical formula for speech entropy, a natural target in speech coding is the achievement of high quality at bit rates that are at least a factor of two lower than the numbers that currently provide high quality: 4 kbps for telephone speech, 8 kbps for wideband speech and 24 kbps for CD-quality speech. These numbers represent a bit rate of about 0.5 bit per sample in each case.

Another challenge is the realization of robust algorithms in the context of real-life imperfections such as input noise, transmission errors and packet losses.

Finally, an overarching set of challenges has to do with realizing the above objectives with usefully low levels of implementation complexity.

In all of these pursuits, we are limited by our knowledge in several individual disciplines, and in the way these disciplines interact. Advances are needed in our understanding of coding, communication and networking, speech production and hearing, and digital signal processing.

In discussing directions of research, it is impossible to be exhaustive, and in predicting what the successful directions may be, we do not necessarily expect to be accurate. Nevertheless, it may be useful to set down some broad research directions, with a range that covers the obvious as well as the speculative. The last part of this section is addressed to this task.

Future Directions

Coding, Communication, and Networking: In recent years, there has been significant progress in the fundamental building blocks of source coding: flexible methods of time-frequency analysis, adaptive vector quantization, and noiseless coding. Compelling applications of these techniques to speech coding are relatively less mature. Complementary advances in channel coding and networking include coded modulation for wireless channels and embedded transmission protocols for networking. Joint designs of source coding, channel coding, and networking will be especially critical in wireless communication of speech, especially in the context of multimedia applications.

Speech Production and Perception: Simple models of periodicity, and simple source models of the vocal tract need to be supplemented (or replaced) by models of articulation and excitation that provide a more direct and compact representation of the speech-generating process. Likewise, stylized models of distortion masking need to be replaced by models that maximize masking in the spectral and temporal domains. These models need to be based on better overall models of hearing, and also on experiments with real speech signals (rather than simplified stimuli such as tones and noise).

Digital Signal Processing: In current technology, a single general-purpose signal processor is capable of nearly 100 million arithmetic operations per second, and one square centimeter of silicon memory can store about 25 megabits of information. The

memory and processing power available on a single chip are both expected to continue to increase significantly over the next several years. Processor efficiency as measured by mips-per-milliwatt of power consumption is also expected to improve by at least one order of magnitude. However, to accommodate coding algorithms of much higher complexity on these devices, we will need continued advances in the way we match processor architectures to complex algorithms, especially in configurations that permit graceful control of speech quality as a function of processor cost and power dissipation. The issues of power consumption and battery life are particularly critical for personal communication services and portable information terminals.

Conclusion

In this thesis low bit rate speech coding techniques and the implementation aspects by using digital signal processors are investigated. Some applications around 4.8 kbps are developed on DSP basis. The actual aim was not to reduce the bit rate of the coded signal, mostly the target was to use the resources (the DSP) efficiently and to improve the quality. In 1985 as Schroeder and Atal were first introduced the CELP coding approach, the needed CPU power was around 333 MIPS. This means at that time for processing 1 second of speech 125 sn CPU time was needed on the Cray-1 computer. For this purpose some simplification methods are proposed and applied for the closed loop LTP and code book search. On the other hand it is shown that for the analysis by synthesis loop of the coder after the excitation parameters are defined, the short term synthesis parameters should be updated to improve the coding quality like bandwidth expansion and interpolation.

In Chapter 2 to form a basis for the later usage, the basic properties and the modelling aspects of the speech signal are investigated. In Chapter 3 speech coding principals and applications are discussed. On top of this, the important particularities for the speech coding algorithm selection for various application areas are examined. Chapter 4 surveys the coding systems, besides this the quantization types and coding schemes are described. Chapter 5 is the consideration of low bit rate speech coding tools. The Chapter 6 is a detailed description of the Analysis by Synthesis (AbS) technique, which is the base approach for CELP and CELP like coding algorithms. Finally in Chapter 7, real time implementation issues are discussed, covering the topics of fixed versus floating point implementation, currently available DSP's, and the processing requirement of a CELP coder and the proposed codebook search simplifications. Also ways of improving the Analysis by Synthesis (AbS) structure used in CELP speech codecs are studied. Methods for updating the short-term synthesis filter once the excitation parameters have been determined are examined. It is shown that, significant improvements can be achieved by updating the synthesis filter, similar to those obtained using the well known methods of interpolation and bandwidth expansion. Furthermore all the proposed methods are applied, to implement a CELP codec by using a fixed point DSP and the high level language C. This result will be used for the voice mail application on a PBX system.

BÖLÜM 1

GİRİŞ

Yüzyıllardan beri, konuşma insanlar arasındaki en önemli iletişim yolu olmuştur. Bundan başka insan yüzyüze iletişimde iletişim imkanlarını da kullanmaktadır. Konuşmanın yanısıra, mimikler, el hareketleri gibi birçok ek iletişim imkanları da kullanılarak insanlar arasında etkin yüzyüze bir iletişim sağlanabilmektedir. Görüşülen kişinin duyma veya görme uzaklığında olmayıp bilgi alışverişi için bir telekomünikasyon sisteminin kullanılması durumunda gerçekleşen iletişim, yüzyüze kurulan iletişimin yerinim olarak tutamadığı halde, günümüzde insanlararası bilgi alışverişinde çok önemli bir rol oynamaktadır. Telekomünikasyon sistemleri üzerinden yapılan konuşma, veri, metin ve resim haberleşmeleri karşılaştırıldığında, insanlararası iletişimin % 90'ının konuşma ile gerçekleştirildiği görülür.

Konuşmanın iletilmesi için kullanılan en popüler haberleşme çeşidi olan telefon haberleşmesi için başlangıçta analog iletim yöntemleri kullanılmıştır. Uzak mesafelere analog iletim için konuşma işareti bir taşıyıcıyı modüle etmek için kullanılmış ve sonra elde edilen bu işaret, kablo veya radyo ile aktarılmıştır. Geleneksel analog iletim yöntemleri basit alıcı ve verici devre yapılarıyla telefon haberleşmesine uzun yıllar hizmet etmiş, fakat uzak mesafe hatlarında kaliteli iletim zorlukları ve gürültüye karşı duyarlılıkları nedeni ile yerini büyük ölçüde giderek ucuzlayan sayısal yöntemlere bırakmıştır. Sayısal iletim yöntemlerinde analog konuşma işareti örneklenmekte, nicemlenmekte, elde edilen örnekler ikili olarak kodlanmakta ve buna karşı düşürülen bir darbe katarı ile iletilmektedir.

Konuşmanın sayısal olarak kodlanmasının birçok üstünlüğü vardır [1]. Öncelikle ikili kodlanmış örneklerin, gürültülü ortamlar sözkonusu olduğunda bile belli bir seviyenin altına düşmediği takdirde yeniden elde edilmeleri çok kolaydır. Böylelikle uzak mesafeli iletimde gürültünün ve kazanç değişimlerinin etkisi en aza indirilmiş

olur. İkinci olarak çeşitli servislere ait işaretler (ses, telgraf, görüntü, metin, faks v.b.) sayısal olarak kodlandıklarında aynı karakterde örneklere sahip olur ve kolaylıkla birlikte aktarılabilir. Üçüncü bir üstünlük ise, bir kere işaret sayısal olarak kodlandıktan sonra güvenli bir iletim sağlamak üzere şifreleme yapılabilmesidir. Bu özellik askeri iletim sistemleri için çok önemlidir.

Konuşmanın sayısal olarak kodlanmasının üstünlükleri yanında kaçınılmaz olarak bir takım sakıncaları da bulunmaktadır. Bunlardan ilki Analog/Sayısal ve Sayısal/Analog dönüşümlerin gerekmesidir. Fakat sayısal tümdevre teknolojisindeki gelişmeler bunu günümüzde bir sakınca olmaktan çıkarmış, hem kalite hem de fiyat yönünden etkin çözümlerin kullanıcıya sunulmasını sağlamıştır. İkinci ve belki de en önemli sorunlardan biri ise, sayısal olarak kodlanan ses işaretinin (ayrıca bir ek kodlama kullanılmadan) analog iletimdekinden çok daha fazla bir bantgenişliği gerektirmesidir. Ancak iletim teknolojisinde son yıllarda meydana gelen çarpıcı gelişmeler, bantgenişliği gereksiniminin giderek daha ucuza ve daha etkin karşılanabilmesi sonucunu doğurmuştur. Böylelikle genişbantlı kodlama yöntemleri ucuz uç cihazı maliyetleri ile bantgenişliğinin etkin şekilde kullanılabilmesini sağlayan, fakat karmaşık kaynak kodlaması ve dolayısı ile pahalı uç cihazları gerektiren yöntemlere karşı tercih edilir olmuştur. Ancak her ne kadar büyük bandgenişlikleri sağlansa da, iletilecek işaretlerin bandgenişlikleri de üstel olarak artmakta, insanoğlu iletim ortamından gittikçe daha çok şey beklemektedir. Sayısal iletişimin kanal kapasitesinin artması gereksinimi, haberleşme kanalının etkin olarak kullanılması problemini, en önemli araştırma konularından biri haline getirmiştir.

1.1 DÜŞÜK BİT HIZLARINDA KONUŞMA KODLAMA

Düşük bit hızlarında konuşma kodlama günümüzde bir çok alanda kullanılmakta ve her geçen gün yeni bir kullanım alanı ortaya çıkmaktadır. İletim ortamlarının etkin band genişliği gün geçtikçe artmakta olmasına rağmen, iletim ortamında aktarılacak istenilen verinin niteliği de değişmekte ve daha fazla band genişliği gerekmektedir. Ayrıca sayısal verinin saklanması uygulamalarında da kaliteyi bozmadan saklanacak

verinin kodlanması ve dolayısı ile bellekte daha az yer işgal etmesi sistem verimliliğini arttırmaktadır.

Sayısal telefon haberleşmesinin bit hızı uzun yıllar Nyquist örnekleme teoremi ile belirlenmiştir [2]. İlk nesil sayısal konuşma iletim sistemlerinde 8 kHz örnekleme hızlı ve 8 bit/örnek kodlamalı 64 kbit/sn'lik standart PCM kullanılmıştır. İkinci nesil şebekelerde kullanılmak üzere geliştirilen 32 kb/sn hızındaki ADPCM bile, sabit ve hareketli uydu haberleşmesi, hareketli sayısal radyo, portatif haberleşme sistemleri ve özel şebekeler gibi uygulamalarda bandgenişliği etkinliği açısından yetersiz kalmıştır. Bu uygulamalarda hem bantgenişliği, hem de güç oldukça sınırlıdır ve etkin kullanılması gerekmektedir. Bundan dolayı bu sistemlerde konuşmanın daha düşük bit hızlarında kodlanması gereği ortadadır. Günümüzde konuşma kodlama teknikleri konuşma kalitesinden fazlaca ödün vermeden 1.2 kbit/san hızın altına bile inebilmektedir. Bu alandaki en popüler kodlama tekniklerinden olan "Kod ile Uyarılan Doğrusal Öngörülü Kodlama" (*Code Excited Lineer Predictive Coding-CELP*) ile konuşmayı 16-4.8 kbit/san'lik hızlarda çok iyi kalitede kodlamak mümkündür [3]. 4.8 kbit/san'nin altında ise kaliteli kodlama için yeni kodlama teknikleri geliştirilmesi gerekmiştir.

Bölüm 2'de konuşma işareti fizyolojik olarak incelenmiş, konuşmanın yapısına ve konuşma üretiminin modellenmesine değinilmiştir. Bölüm 3'de konuşma kodlamanın temelleri, prensipleri ve uygulama alanları araştırılmıştır. Bunun yanı sıra çeşitli uygulama alanları için algoritma seçimini etkileyen önemli noktalar açıklanmıştır. Bölüm 4'de kodlama sistemlerinin genel bir sınıflandırması yapılmış, nicemleme ve kodlama tipleri incelenmiştir. Bölüm 5'de düşük bit hızlı konuşma kodlama uygulamalarında kullanılan konuşma kodlama araçları verilmiştir. Bölüm 6'da önemli kodlama yaklaşımlarından sentezleyerek-analiz yaklaşımı ayrıntılı olarak açıklanmış, başta CELP olmak üzere bu yaklaşımı uygulayan belli başlı kodlama tipleri ele alınmıştır. Bölüm 7'de ise DSP tabanlı bir CELP kodlayıcı gerçeklemesi açıklanmış, kaynakları verimli kullanmak için konvolüsyon, çaprazilişki ve özilişki hesaplamalarında bazı sadeleştirme teknikleri önerilmiş, ayrıca kodlayıcının performansını arttırmak için sentezleyerek-analiz çevriminde çeşitli uyarma

parametrelerini bandgeniřlięi geniřletme (*bandwith expansion*) ve aradeęerleme (*interpolation*) yaklařımlarında olduęu gibi gncellemek performansta nemli geliřmeler saęlamaktadır.

Bundan bařka btn nerilen geliřtirmelerin kullanıldıęı bir CELP kodlayıcısı, yksek seviyeli yazılım dili C kullanılarak gereklenmiř ve sabit noktalı bir sayısal iřaret iřlemciye uygulanmıřtır. Elde edilen sonuların sayısal bir dahili santralin sz bellek (voice mail) modlnde kullanılması hedeflenmektedir.

1.2 TEZ İLE GETİRİLEN YENİLİKLER

Bu tez kapsamında dřk bit hızlarında konuřma kodlama teknikleri ve sayısal iřaret iřlemciler (DSP) ile gerekleme yntemleri arařtırılmıř; bu tekniklerden hareketle, 4.8 kbit/san civarındaki bit hızları iin DSP tabanlı gereklemeler yapılmıřtır. alıřmada, daha dřk hızlara inmek deęil eřdeęer hızlarda kaynakları (DSP imkanlarını) daha verimli kullanıp, daha kaliteli kodlayıcılar tasarlama amacı gdlmřtr.

- Tipik bir CELP konuřma kodlama algoritmasında iřlem sresinin oęu kapalı evrim LTP ve kod kitabı arama iin harcanmaktadır. Bu iřlemlerde oluřan karmařıklık konvolsyon, apraziliřki ve ziliřki hesaplamaları sonucunda oluřmaktadır. Tezde bu karmařıklıęı azaltacak sadeleřtirme teknikleri nerilmiřtir.
- Ayrıca kodlayıcının peformansını arttırmak iin sentezleyerek-analiz (Analysis by Synthesis-AbS) evriminde eřitli yenilikler uygulanmıřtır.
- Bylelikle bu nerilen geliřtirmelerin kullanıldıęı bir CELP kodlayıcısı, yksek seviyeli yazılım dili C kullanılarak gereklenmiř ve sabit noktalı bir sayısal iřaret iřlemciye uygulanabilmiřtir.

Yukarıda sayılan maddelerden dolayı tez zgn nitelik tařımakta ve varolan bir ynteme yenilik getirerek uygulama alanını geliřtirmek iin destek oluřurmaktadır.

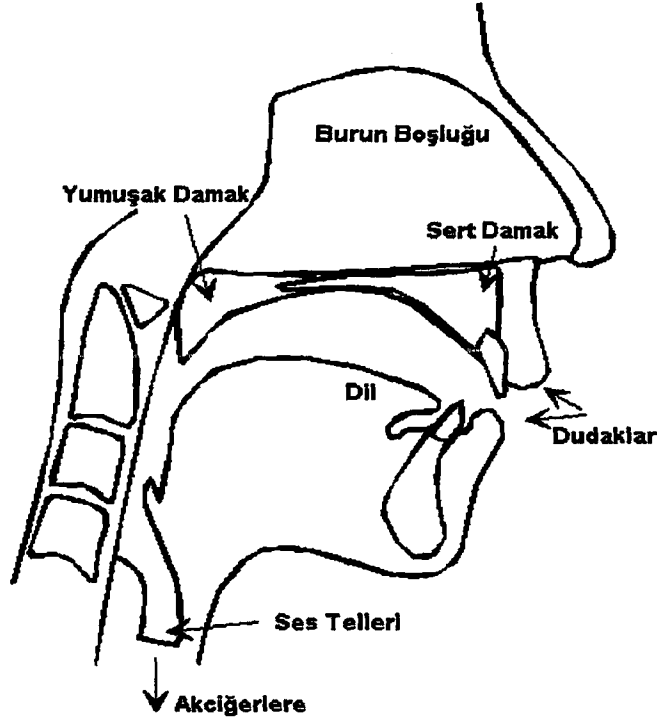
BÖLÜM 2

KONUŞMANIN FİZYOLOJİK OLARAK İNCELENMESİ

2.1 KONUŞMANIN YAPISI

İnsan-İnsan iletişiminin en önemli araçlarından biri olan konuşma, ağızdan çıkartılan seslerin ve sessizliklerin arka arkaya gelmesi ile oluşur. Tabii ki iletişimde bulunacak olan kişilerin önceden kullanacakları ses dizilimleri (sözcükler) ,sözcük dizilimleri (cümleler) ve sözcük dizilimlerinin kuralları (gramer) üzerinde anlaşmış olmaları gerekir. Bütün bunlar, insanların iletişimde kullandıkları dilin temellerini oluşturur.

İnsan, sesi nasıl üretir? Normal insan sesi belli bir basınç altında akciğerlerden gelen havanın ses tellerini titreştirmesine uyarak art arda, kesik kesik çıkması ile oluşur. Ses telleri gırtlak dili denilen bir aralığın içinde yatay olarak dizilmiş kas kıvrıntılarıdır. Ses telleri bir müzik aletinin telleri gibi ilk hareket verildikten sonra kendi kendine titremez; bunların titreştirilmesi için en önemli rolü gırtlak hareket siniri oynar. Örneğin bir kişi do sesi çıkarttığında gırtlak harekeket siniri ses tellerine saniyede 259 uyarı göndererek bunların kasılmasını sağlar, böylelikle gırtlak dili arasından 259 defa hava geçer ve ağız yutak boşluğunda basınç yükselmiş olur. Soluk verici kasların kasılması ile beslenen (özellikle karın çeperi kasları) bu gırtlak dili altı basınç, çeşitli ses siddetlerinde (normal konuşma, bağırma, şarkı söyleme) değişen değerler alır. Ses tellerinin titreşimi saf bir kas-sinir eylemidir ve beyinden gelen darbeli bir akım ile idare edilir [4]. Bu sonuca ilk olarak *Raoul Husson* 1950 yılında varmuştır.



Şekil 2.1 İnsanın Ses Üretme Aracı

Sesin oluşumu için; 1. *Enerji Kaynağı*, 2. *Titreşim Yapan Organ*, 3. *Rezonans Boşluğu*, 4. Ses oluşumunu ve bununla ilgili organların eşzamanlamasını yapabilen bir *Sinir Sistemi* gereklidir[2]. *Enerji Kaynağı*, soluk alıp verme sırasında, basınç altında dışarı atılan havadır. Akciğerlerin esnekliği ve solunum kaslarının kasılması ile basınç sağlanmaktadır. *Titreşim Yapan Organ*, ses tellerinden ibarettir. Ses telleri gırtlakta bulunan bir çift ince kastır; bunların bir ucu sabittir, diğer uçları ise ibriksi kıkırdakların (aritenoitler) içine girer. Sessizlik halinde, ibriksi kıkırdaklar dışarıya doğru çekilerek solunum için gırtlak dilini açık bırakır. Soluk verirken meydana gelen ses oluşumu sırasında, ibriksi kıkırdaklar ses tellerini birbirine yaklaştırarak gırtlak dilini tıkar; bu durumda, hava basınç altında, bu tellerin arasından geçerek titreşim yapar (gırtlak sesi); sonra da bu ses üstteki hava tabakasına geçer. Gırtlak sesi, sesli harflerin üretilmesi için tek titreşim kaynağıdır. Dil ve dudaklar ise yardımcı titreşim organlarıdır; dil damağa veya dişlere dayanarak, dudaklar birbirine değerek sessiz harflerin oluşmasında rol oynar, fısıldama ve ısıklık seslerini de bunlar sağlar. *Rezonans Boşluğu*; burun boşluğu, sinüsler gibi sabit ve ağız, yutak gibi hareketli boşluklardan ibarettir. Bu boşluklar, harmonikleri hafifleterek veya kuvvetlendirerek sesleri değiştirir; sesin başlangıcında veya sonunda az veya çok sert

çıkmasında etkili olarak, insan sesinin boğumlanmasına özgü inişli çıkışlı ara sesleri oluşturur. Ses oluşumunu sağlayan *Sinir Sistemi*'nin merkezi omurilik soğanındadır; bu merkez, solunum, gırtlak ve ağız-yutak sinirleri aracılığı ile solunum, gırtlak ve ağız-yutak kaslarının çalışmasını düzenler. İnsanın konuşması Sylvius yarığının iki yanında, beyin kabuğunda bulunan konuşma merkezinin omurilik soğanındaki merkeze emir vermesi ile olur [5].

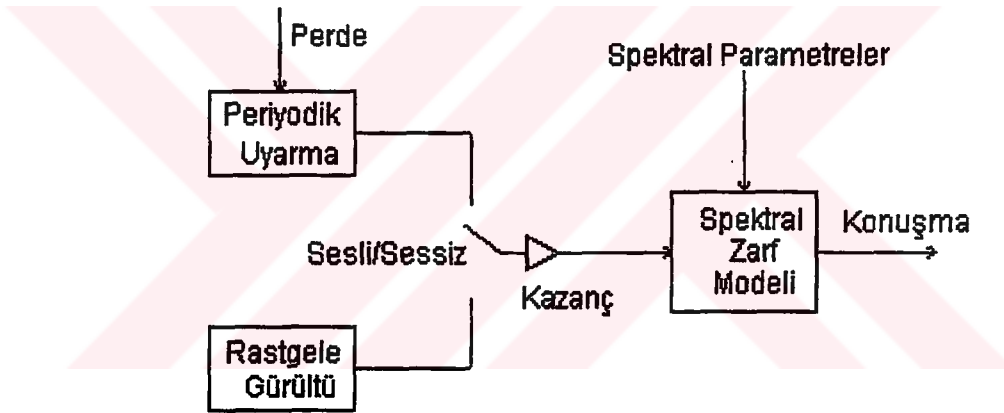
Bütün bu unsurlar ortaya çıkan seslerin fiziksel özelliklerinde, değişik oranda rol oynamaktadır: *Sesin Genliği* göğüs kafesinin yarattığı basınca ve ses tellerinin bu basınca karşı gösterebileceği dirence bağlıdır. *Sesin Frekansı* ses tellerinin boyutları, gerilimi ve rezonans boşluklarının büyüklüğü ile ilişkilidir. Çocuk ve kadınlarda ses telleri ve rezonans boşlukları küçük olduğu için sesler ince çıkar; ses telleri ve rezonans boşlukları gelişmiş olan yetişkin bir erkekte ise ses kalın olur. İnsanda rezonans boşlukları çıkarılmak istenen sese göre ayarlanır; kalın ses çıkarılacaksa , gırtlak ve yutak aşağıya iner, boşluklar büyür; ince ses çıkarılacaksa aksine gırtlak-yutak sistemi yukarıya kalkar. *Ses Perdesi*, herhangi bir insanda iki oktav, sesini geliştirmiş kimselerde ve şarkıcılarda iki oktavdan fazla olabilir. En kalından en inceye doğru ses perdeleri şöyle sıralanır: *bas, bariton, tenor, kontralto, mezzo-soprano* ve *soprano*. *Sesin tınısı*, yani çeşitli harmoniklerin miktarı öncelikle gırtlak sesine (ses tellerine); sonra da ağız, gırtlak ve burun boşluğu gibi rezonans boşluklarına bağlıdır [6].

2.2 KONUŞMA ÜRETİMİNİN MODELLENMESİ

Yukarıda insanın konuşurken fizyolojik olarak uyarılan ve uyaran iki tür yapı kullandığı anlatılmıştır. Uyaran yapının önemli bir parçası olan ses telleri, sesli harflerin oluşumu sırasında önemli bir rol oynar ve ciğerlerden gelen hava içinden geçerken beyinden gelen emirle uygun frekansta titreşir. Burada oluşan titreşimler özellikle sesli harflerin kalıcı (stasyonier) bölgelerinde periyodiktir. Bu periyodik işaretin frekansına, *konuşmanın temel frekansı* adı verilir ve periyodu da ses işaretinin perdesi (*pitch*) olarak tanımlanır. Sessiz harflerin üretilmesi sırasında ise ses telleri açılır; hava akımı hemen hemen bir engelle karşılaşmadan ses tellerinin

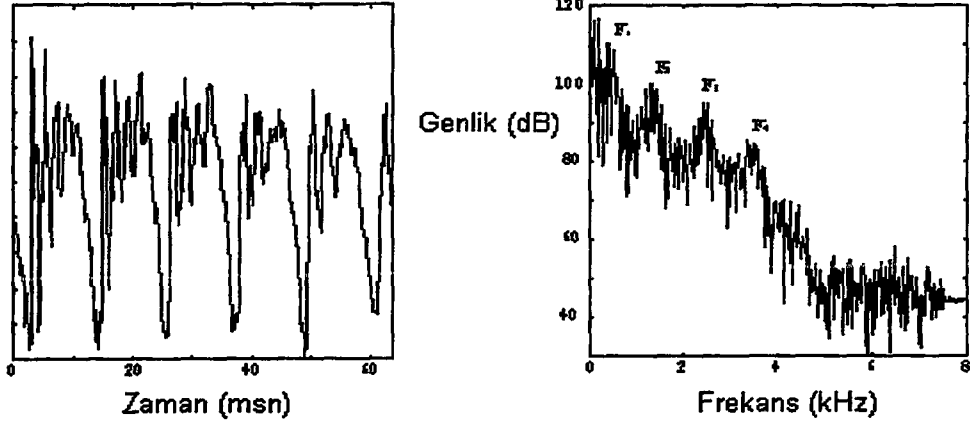
içinden geçer, dişler, geniz ve ağız kullanılarak hava akımı küçük bir aralıktan geçmeye zorlanır. Bu şekilde oluşan türbülans rastgele değişim gösteren gürültü şeklinde bir işaret yaratır. Ses tellerinin bitiminden başlayan ve diğer kısma açılan bölge, akustik bir tüp gibi düşünülebilir. Hem periyodik işaret (sesli harfler), hem de gürültü şeklindeki işaret (sessiz harfler) ile ağız ve geniz boşluğundan oluşan bu tüpteki hava tetiklenir ve titreşime sokulur. Burada oluşan titreşim veya rezonans frekanslarına *formant* adı verilir. Formant frekansları akustik tübün frekans cevabını oluşturur. Titreşim halindeki hava sonuçta ağız ve/veya burundan boşluğa yayılır ve insan kulağı veya herhangi bir ses alıcısı tarafından konuşma olarak algılanır [5].

Bu bilgilerden hareketle insan sesi üretim mekanizması Şekil 2.2'deki gibi modellenebilir.

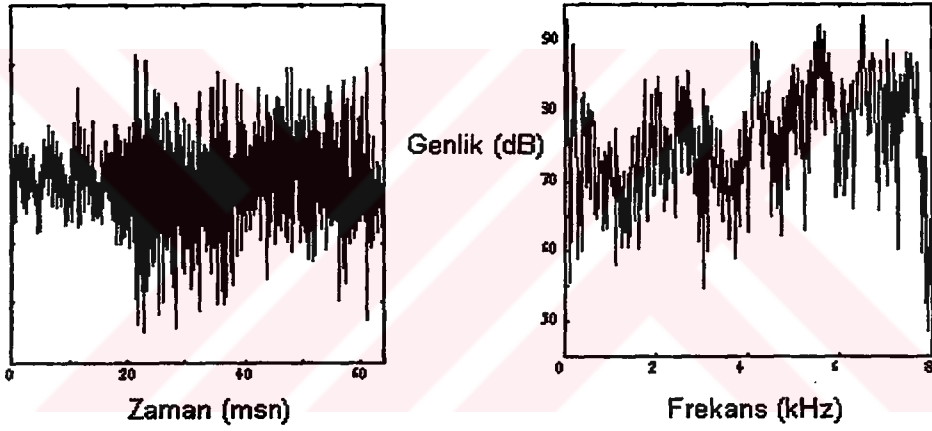


Şekil 2.2 İnsan Sesi Üretim Mekanizması Modeli

Şekil 2.3 ve 2.4'de konuşma sırasında kullanılan sesli ve sessiz harflere ait örnek frekans spektrumu ve rezonans (*formant*) frekanslarının birleşiminden oluşan ses işareti spektrumu verilmiştir.

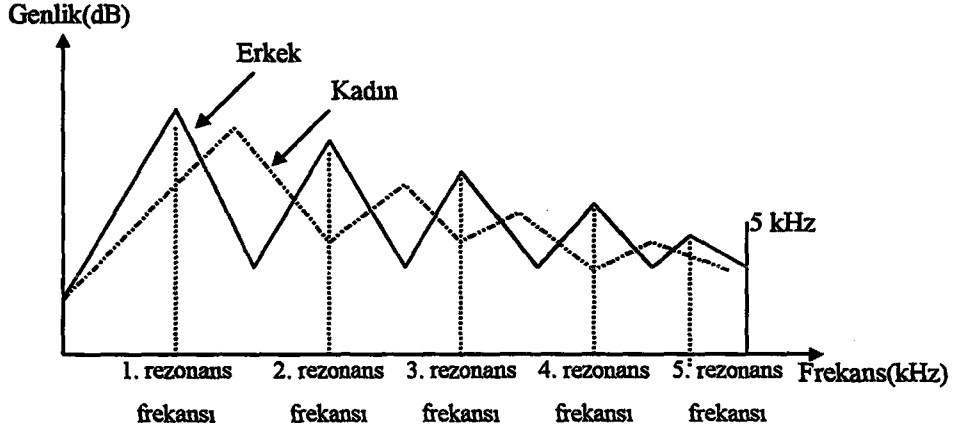


Şekil 2.3 Sesli Harf Zaman İşareti ve Spektrumu



Şekil 2.4 Sessiz Harf Zaman İşareti ve Spektrumu

Konuşma sırasında ağız ve geniz boşluğunun oluşturduğu akustik tübün frekans cevabını oluşturan rezonans frekansları, konuşmacının kadın veya erkek olmasına göre farklılık göstermektedir. Titreşim halindeki hava sonuçta ağız ve/veya burundan boşluğa yayılırken Şekil 2.5'te de görüldüğü gibi erkek sesleri için beş, kadın sesleri için dört rezonans frekansı oluşmaktadır [6].



Şekil 2.5 Erkek ve Kadın Ses Spektrumlarının Karşılaştırılması

Konuşma kodlamada istenilen başarımın elde edilebilmesi için ilgilenilen işaretin özelliklerinin iyi bilinmesi çok önemlidir. Ancak bu şekilde konuşma işaretinin içerisindeki artık bilgi atılarak daha etkin kodlama ile düşük bit hızlarına inilebilir.

BÖLÜM 3

KONUŞMA KODLAMA; GEREKSİNİMLER VE UYGULAMALAR

3.1 KONUŞMA KODLAMA PRENSİPLERİ

Konuşma sırasında iletilen bilgi miktarını değerlendirmek için çeşitli yollar bulunmaktadır. Bunlardan en sık kullanılanı *Shannon*'un enformasyon teorisi yaklaşımıdır. Enformasyon teorisine göre konuşma içerdiği mesaj yada taşıdığı enformasyon miktarına göre değerlendirilebilir. Buna alternatif bir yaklaşım ise mesajı taşıyan işaret, yani akustik dalga şekline göre karakterize etmektir. Enformasyon teorisi yaklaşımları iletişim sistemlerinin şekillenmesinde önemli bir rol oynamış olmasına rağmen, konuşmanın dalga şekli veya parametrik modellerle temsili de pratik uygulamalarda oldukça sık kullanılmaktadır.

1930'lu yılların sonlarında PCM teknolojisinin bulunarak, 1960'lı yıllarda kullanılmaya başlanması konuşma iletişimde bir devrim yaratmıştır. Sayısal teknoloji o zamandan beri birçok avantajı nedeniyle neredeyse her tür iletişim türüne uygulanmış ve başarıyla kullanılmıştır. *Nyquist* teoremi uyarınca telefon haberleşmesi (300-3400 Hz) için konuşma 8 kHz ile örneklenmekte ve Avrupa standardında 64 kbit/san (8 bit/örnek) hızla iletilmektedir. Adaptif diferansiyel PCM (ADPCM) iletim hızını 32 kbit/san'ye düşürmüş olmasına rağmen, teknolojik kısıtlar nedeni ile o zamanlarda daha aşağılara inilememiştir.

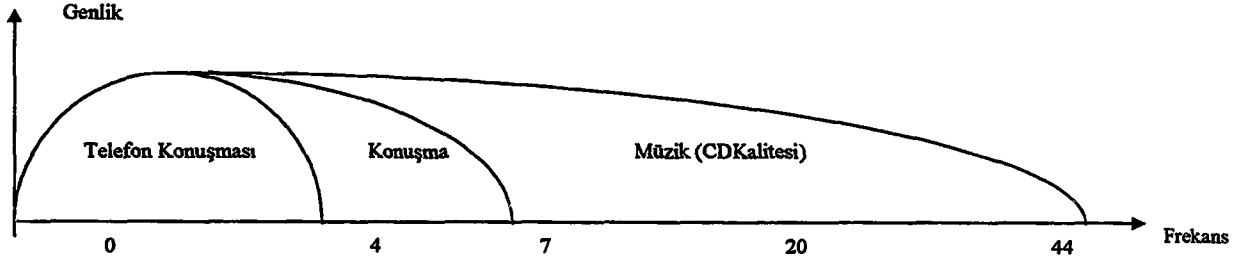
Konuşma kodlama yöntemleri, bit hızı, iletme ve saklamanın maliyeti, karmaşıklık ve konuşma kalitesi ölçütlerine göre değerlendirilir. Değişik alanlarda uygulanan kodlama teknikleri kullanım yerine göre çeşitlilik gösterir. Buna rağmen uygulama spektrumu iki ekstrem arasında değerlendirilebilir. Bir kamu telefon şebekesinde kodlamanın kalitesi vazgeçilmez ölçüttür. 64 kbit/san'lik PCM uygulaması bu

gereksinimi fazlasıyla karşılar. Bir askeri haberleşme sisteminde ise durum farklıdır; burada istenen tek bir kanal üzerinden yüksek güvende haberleşme ve her zaman haberleşebilme garantisidir ve konuşma kalitesi ve maliyet ikinci planda kalmaktadır. Bu iki ekstrem arasındaki farklı uygulamalar için en iyi maliyet/performans oranını sağlayacak bir çok yöntem geliştirilmiştir.

Sayısal haberleşme sistemlerinde eğilim yüksek bit hızlarından (>16 kbit/san), ilk aşamada orta hızlı (8 ila 16 kbit/san arası) ve daha sonra da düşük hızlı (<8 kbit/san) konuşma kodlayıcıların kullanılması yönünde olmuştur. Yüksek ve orta hızlı kodlayıcılar şimdiki sayısal telefon haberleşmesinde, orta hızlı kodlayıcılar hücresel sayısal mobil radyo sistemleri ve mobil haberleşmede ve düşük hızlı kodlayıcılar ise askeri haberleşmede ağırlıklı olarak kullanılmaktadır.

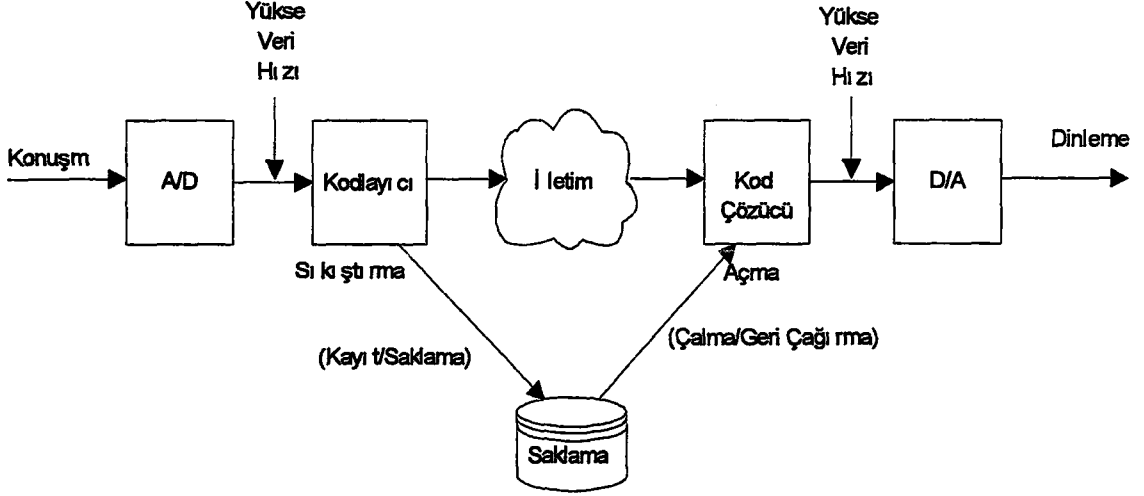
3.2 KONUŞMA KODLAMANNIN TEMELLERİ

İnsan kulağı 5Hz ila 20kHz arasındaki frekanslardaki sesleri duyabilmektedir. Bunun doğal sonucu olarak bu işaretlerin sayısal olarak kodlanması gerektiğinde 20kHz bandgenişliğine sahip olan kodlayıcılar kullanılması gerekmektedir. Bu şekilde uygulanan kodlama sonucunda elde edilen kaliteye kompakt disk (*Compact Disc-CD*) kalitesi denilmekte ve örnekleme frekansı olarak genellikle 44kHz kullanılmaktadır. Bu oldukça büyük band genişliği yüksek doğrulukta (*High Fidelity-Hi-Fi*) müzik, özellikle klasik müzik dinlemek için kullanılmaktadır. Bununla birlikte ses bilgisinin büyük çoğunluğu bu 20kHz'lik bandın alt yarısında bulunmaktadır. İnsan sesi bileşenleri ise 7 ila 10 kHz arasında iyice zayıflamaktadır. Buna rağmen yalnızca 3.4kHz bandgenişliğine sahip olan telefon haberleşmesinde insan sesi ve özellikle müzik nitelik değiştirmektedir. Bundan dolayı bazı çokluortam (*multimedia*) ve telekonferans (*teleconferencing*) uygulamalarında band genişliği 7 kHz olan kodlayıcılar kullanılmaktadır.



Şekil 3.1 Çeşitli Karakterdeki Ses İşaretlerinin Band Genişlik leri

Konuşma kodlama sistemlerinin temelinde hemen hemen bütün algoritmalarından bağımsız olarak işaretin analizi, önceden tahmin edilen fazlalık bilginin atılması ve esas bilgiyi içeren kısmın verimli bir şekilde kodlanması bulunmaktadır. Kodlama işlevi, “sıkıştırma” ve kod çözme işlevi de, “açma” olarak tanımlanabilir. Kodlama işleminden sonra sıkıştırılmış bilgi ya alıcıya iletilir, ya da bir bellekte saklanır. İletilen bilgi alıcıda yeniden işlenerek açılır ve orijinal işaret elde edilir. Bilgiyi saklama durumunda ise bellekten çağrılan bilgi açılarak “çalınır” (*playback*). Bir sıkıştırma algoritması yukarıda anlatılan süreç içinde orijinal bilginin alıcı tarafından anlaşılabilir veya anlaşılamayacak bir kısmını kaybedebilir, bu tip algoritmalara “kayıplı sıkıştırma algoritmaları” denilir. Eğer bir sıkıştırma algoritması herhangi bir bilgi kaybına yol açmıyorsa ve kodlanan işaret alıcıda tam olarak orijinal haline dönüştürülebiliyorsa bu tip algoritmalara “kayıpsız sıkıştırma algoritmaları” adı verilir.



Şekil 3.2 Tipik Konuşma Kodlama (Sıkıştırma) Uygulamaları

3.3 KONUŞMA KODLAMA UYGULAMALARI

Konuşma kodlama günümüzde bir çok alanda kullanılmakta ve her geçen gün yeni bir kullanım alanı ortaya çıkmaktadır. İletim ortamlarının etkin band genişliği gün geçtikçe artmakta, öte yandan iletim ortamında aktarılacak istenilen verinin niteliği de değişmekte ve daha fazla band genişliği gerekmektedir. Ayrıca sayısal verinin saklanması uygulamalarında da kaliteyi bozmadan saklanacak verinin kodlanması ve dolayısı ile bellekte daha az yer işgal etmesi sistem verimliliğini arttırmaktadır. Aşağıda konuşma kodlamanın değişik alanlardaki uygulamalarına kısaca değinilecektir.

3.3.1 Kamu Telefon Şebekesi

Kamu telefon şebekelerindeki uygulamalar ITU-T tarafından düzenlenmekte ve uygulamalara ait standartlar yayınlanmaktadır. Telefon şebekesindeki konuşma işaretine ait gereksinimler aşağıda belirtilmiştir:

Konuşma kalitesinin yüksek olması: Kamu telefon şebekesinde kullanılan kodlayıcıların kalitesi genellikle trunk (*toll*) kalitesi ile yayın (*broadcast*) kalitesi arasında tanımlanmaktadır.

Gecikmenin düşük olması: Kamu telefon şebekelerinde kullanılan teçhizatların gecikmesi ekoyu engellemek için düşük tutulmaktadır.

Şebeke maliyetlerinin düşük olması: Şebekede kullanılan elemanların karmaşıklığı ve gerçekleştirme maliyetleri düşük olmalıdır.

Bu gereksinimlerden dolayı kamu telefon şebekelerinde kullanılan kodlayıcılar genelde yüksek bit hızlı dalga şekli kodlayıcılardır. Kamu haberleşme şebekelerinde, sayısal teknolojinin ilk uygulaması 1972 yılına 64/56 kbit/san'lık PCM sisteminin uluslararası standart olarak kabulü ile olmuştur [2][7]. Sayısal sistemlerin şebeke uygulamasındaki üstünlükleri ilk uygulamalar sonucunda görüldüncel, 1985 yılında 32 kbit/san'lık ADPCM yöntemi takip eden standart olarak yayınlanmıştır. Her iki kodlayıcı da son derece kaliteli çıktı vermekte ve gerçeklenmeleri de oldukça basittir [8]. Bu yöntemler iletim ortamının tam olarak verimli kullanılmasına izin vermediklerinden, kamu haberleşme şebekelerindeki kodlayıcılar için 16 kbit/san'lık LD-CELP (*Low Delay-CELP*) standardı üçüncü nesil standart olarak yayınlanmıştır. Günümüzde kamu haberleşme şebekeleri geniş coğrafi yayılma alanları ile gittikçe bir Tümüleşik Hizmetler Sayısal Şebekesi'ne (*ISDN-Integrated Services Digital Network*) dönüşmektedir. Bunun sonucunda aynı şebeke konuşma işaretlerinin yanında faks, veri, görüntü gibi işaretleri de aynı fiziksel ortam üzerinde taşımakta ve iletim ortamının daha etkin kullanılması gereksinimi her geçen gün daha da artmaktadır.

3.3.2 Sayısal Hücreşel Mobil Radyo Sistemleri

Hücreşel telefon sistemleri gibi spektral yeterlilikleri ile bilinen, modern taşınabilir radyo sistemleri yüksek kapasite elde etmek için kanalın yeniden kullanımını ilkesini uygular. Bundan dolayı sayısal hücreşel mobil radyo sistemleri işaretin sönmesine (*fading*) ve kanallar-arası girişime karşı yeterli önlemler içermelidir. Son olarak, sistem gerçeklemesi de, maliyet ve güç kısıtlamalarını sağlamak için, uygun

karmaşıklıkta olmalıdır. Büyük boyutlu bir sayısal hücrese mobil radyo sisteminin sağlanması gereken temel şartlar aşağıdaki gibidir:

- Büyük abone kapasitesi
- Spektrumun etkin olarak kullanılması
- Ulusal ve uluslararası uyumluluk
- Geniş kapsama alanı
- Değişik trafik yoğunluklarına uyabilme
- Kamu haberleşme sistemleri kalitesinde iletişim

Şu anda Avrupa'da kullanılmakta olan sayısal hücrese mobil radyo sistemi, *GSM (Group Speciale Mobile)* adı verilen çalışma grubunun uzun yıllar süren çalışmaları sonucunda ortaya çıkmıştır. Bu çalışma çerçevesinde önce ileriye doğru hata düzeltme bilgisi ile birlikte 22,8 kbit/san hızındaki RPE-LTP (*Regular Pulse Excitation-Long Term Prediction*) kodlama algoritması kullanılması öngörülmüştür [9][10]. Daha sonra kapasiteyi daha da arttırmak için algoritma 13 kbit/san hızında çalışacak şekilde güncellenmiş, böylelikle aynı band genişliği kullanılarak iki konuşma iletilebilmiştir.

3.3.3 Mobil Uydu Haberleşmesi

Günümüzde dünyanın herhangi bir yerindeki bir kişi, mobil uydu hizmetleri aracılığıyla dünyanın başka bir yerindeki bir kişi ile haberleşebilmektedir. Mobil uydu sistemleri jeo-senkron uydu, yada uydularla haberleşebilen terminaler ile ya yayın modu, ya da başka çoklu erişim yöntemleri kullanılarak çalışır. Bu tip sistemlerde kullanılan konuşma kodlayıcıların en önemli özelliği düşük bit hızlarıdır. Böylelikle iletim bandı daha verimli olarak kullanılabilir ve uyduda zaten kısıtlı bulunan güç gereksinimini azaltılabilir. Mobil uydu servisleri, pratikte herbiri farklı kullanıcılara hitap eden farklı servislerdir. Bu servisler aşağıdaki gruplara ayrılabilir:

- i) **Deniz Mobil Haberleşmesi:** Uluslararası Denizcilik Uydu Organizasyonu (INMARSAT) yukarda bahsedilen hizmetleri sağlayan ilk sivil işletmecidir.

Bu organizasyon gemi işletmecilerine hem veri, hem de ses haberleşmesi sağlayabilmektedir. Denizcilik hizmetlerinin doğası gereği bu servis geniş kapsama alanlı ve uluslararası nitelikte olmalıdır. Ayrıca bu şebekede kullanılacak terminal cihazları ağırlık/karmaşıklık, transmisyon kapasitesi ve fiyat açısından da elverişli olmalıdır. Transmisyon kapasitesinin düşürülebilmesi için düşük bit hızlarında kodlama yöntemi (örn. 16 kbit/san APC-*Adaptive Predictive Coding*) kullanılmakta ve bandın en etkin şekilde kullanılması sağlanmaktadır. INMARSAT-A ve INMARSAT-B [11] uyduları, günümüzde dünyada denizcilik alanında oldukça geniş bir bölgede gemiler tarafından kullanılmakta ve hizmetin genişletilmesi için çalışmalar sürdürülmektedir.

- ii) **Kara Mobil Haberleşmesi:** Kara mobil haberleşme sistemleri de kara taşıtlarına ses ve veri bazında haberleşme imkanı sağlamaktadır. Bu sistemlerde de yukarıda anlatılan güç ve band kısıtlamaları aynen geçerlidir. Kara mobil haberleşme uydularına ayrılam iletim band genişliği 4 MHz civarındadır. Bu tip servislerin ekonomik olabilmesi için haberleşme kanalları için ayrılan bandın çok dar tutulması gerekmektedir. Bu nedenle kullanılan ses sıkıştırma algoritmalarına, INMARSAT'ın 6.4 kbit/san'lik "Çoklu-Band Uyarmalı" (*Multi-Band Excitation-MBE*) kodlayıcısı [12] ve NASA'nın MSAT-X'de kullandığı 4.8 kbit/san'lik "Vektör Adaptif Öngörülü" kodlayıcısı (*Vector Adaptive Predictive Coding-VAPC*) örnek olarak verilebilir.
- iii) **Hava Mobil Haberleşmesi:** Askeri havadan-havaya ve havadan-karaya uygulamaların bir uyarlaması kullanılarak, hava mobil haberleşme sektörü INMARSAT'ın bu alana girmesi ile canlanmıştır. Dünyada şu anda var olan 2500 adet yolcu uçağında günde milyonlaca kişi seyahat etmektedir. Bu yolcuların havada iken de haberleşme ihtiyacını karşılayabilmek ve mürettebatın da kıtalararası uçuşlar sırasında haberleşmelerini yürütebilmesini sağlayabilmek için ilk olarak 1989 yılında uçak içinde uydu üzerinden telefon servisi başlatılmıştır. Gök telefonu (sky-phone) adı verilen bu hizmet için, kısıtlı uydu band genişliğinden dolayı 9.6 kbit/san'lik MPLPC

(*Multi Pulse Linear Predictive Coding*) algoritması kullanılmıştır. Günümüzde bu servis yaygınlaşmış ve yolcu uçaklarına dünya çapında telefon hizmeti verilmeye başlanmıştır.

3.3.4 Kişisel Haberleşme Sistemleri(*Personal Communication Systems-PCS*)

Kişisel haberleşme sistemleri felsefesinin ardında yatan düşünce, insanların her türlü haberleşme için küçük el terminalleri kullanmasıdır. Bu tip şebekeler kullanıcıya bu el terminalini kullanarak, bulunulan coğrafi yerden ve kullanılacak haberleşme tipinden bağımsız olarak haberleşme imkanı sağlamayı amaçlar. Bu bir anlamda Tümleşik Hizmetler Sayısal Şebekesinin radyo dalgaları ile gerçekleşmiş olarak da düşünülebilir. Kişisel haberleşme sistemleri, yukarıda bahsedilen hizmetleri tek bir telsiz şebekede toplayan bir yapı olarak henüz bir idealdir. Fakat çok yakın bir gelecekte, el terminalleri yeterince elverişli ve etkin olarak sağlanıp, mobil şebeke altyapıları da kaplama alanlarını yeterli seviyeye ulaştırmaya yettiğinde; bir ideal olmaktan çıkıp gerçek olacaktır. Hele aynı el terminalinin hem kamu şebekesinde hem de dahili santral şebekelerinde etkin olarak çalışması sağlandığında ise, insanlar hem dışarıda hem de işyerlerinde aynı el terminalini kullanabilecektir. Konuşmanın, insan haberleşmesi içindeki önemli yeri düşünüldüğünde, bu sistemlerde de konuşmanın düşük bit hızlarında etkin olarak kodlanması önemli bir konu olacaktır.

3.3.5 Çok Küçük Açıklıklı Terminal (VSAT) Şebekeleri

1984 yılında Amerika Birleşik Devletleri'nde haberleşme alanındaki özelleştirmenin başlamasından sonra, yakın iş şebekeleri için sabit uydu haberleşmesinin kullanımında bir patlama yaşanmıştır. Burada önemli rolü; ucuz, düşük kapasiteli, hafif ve çanak anten çapları küçük olan, çok küçük açıklıklı terminal (*Very Small Aperture Terminal-VSAT*) olarak da bilinen cihazların piyasaya sürülmesi oynamıştır. Bu şebekeler genellikle, terminallerin merkez istasyonla haberleştiği yıldız yapısındadır. Merkezdeki istasyon daha büyük bir antene ve uydu alıcı vericisine erişim için gerekli protokolü yürütecek, bağlaşmayı gerçekleştirecek, aynı zamanda da diğer merkezi işlevleri yerine getirecek işlem kapasitesine sahiptir.

VSAT sistemleri için önemli bir uygulama alanı da ekonomik ve coğrafi nedenlerden dolayı kısıtlı yerüstü haberleşme altyapısı olan, kırsal haberleşme sistemleridir. Çok küçük açıklıklı terminaller aslında düşük hızlı veri haberleşmesi için düşünülmese de, özellikle kırsal bölgelerde ve gelişmekte olan ülkelerde artan bir oranda konuşma haberleşmesi için de kullanılmaktadır[13]. Haberleşme band genişliğinin etkin şekilde kullanılması ve güç harcamasının en aza indirilebilmesi için, bu gibi uygulamalarda kullanılacak kodlayıcılar da düşük bit hızlı, ucuz ve kolay şekilde gerçekleştirilebilir olmalıdır.

3.3.6 Askeri Haberleşme

Haberleşme endüstrisinde önemli bir paya sahip olan askeri haberleşme, aynı zamanda kendine özgü karakteristikleri taşımaktadır. Bunlar aşağıdaki gibi özetlenebilir:

- i) **Güvenlik:** Askeri haberleşmede genellikle istenmeyen kişilerin dinlemesini engellemek için şifreleme yöntemi kullanılır. Bir şifreleme yönteminin güvenlik derecesi şifrelenmiş işarette kalan esas işaretin anlaşılabilirliği ve şifreleme için kullanılan anahtar sayısına bağlıdır.
- ii) **Konuşma Haberleşmesinin Kalitesi:** Haberleşmede sıklıkla istenen alıcıya mümkün olduğu kadar aslına uygun işareti taşımaktır. Bunun tersine askeri haberleşmede ise konuşmanın aslına uygunluğu, konuşmacının kimliği veya duyguları önemli değildir. Esas önemli olan konuşmanın taşıdığı enformasyonun, hatasız şekilde alıcıya ulaştırılmasıdır. Bu nedenle askeri haberleşmede kullanılan konuşma kodlama algoritmaları çok düşük bit hızlarına inebilmekte ve sentetik kalitedeki en iyi sesi üretebilmeyi amaçlamaktadır.
- iii) **Kolay Kurulum ve Kullanılabilirlik:** Askeri haberleşme donanımı savaş alanındaki hareketlilik ve zor şartlar düşünüldüğünde fazla parçadan oluşmamalı, küçük olmalı ve kolayca kullanıma hazır hale getirilebilmelidir.

Askeri uygulamalarda kullanılan düşük bit hızındaki konuşma kodlama yöntemlerine örnek olarak; Amerikan ordusu için federal standart olarak belirlenmiş olan 2.4 kbit/san hızındaki LPC-10 algoritması [14], 4.8 kbit/san hızındaki CELP algoritması [15][16] ve çok yeni olarak bir çok aday arasından seçilen 2.4 kbit/san hızındaki MELP (*Mixed Excited Linear Predictive Coding*) algoritması [17] sayılabilir.

3.4 ALGORİTMA SEÇİMİNDE ÖNEMLİ UNSURLAR

Bir kodlama algoritmasının tasarımı ve kapasitesi genellikle uygulama alanı ile belirlenmektedir. Bundan dolayı algoritmanın tasarımı çatışan amaçların ve birbirini etkileyen faktörlerin dengelenmesi ile başlamaktadır. Algoritmanın seçilmesini etkileyen temel faktörler Tablo 3.1’de verilmektedir.



Tablo 3.1 Konuşma Kodlama Uygulamaları ve Şebeke Gereksinimleri

(DMR-Digital Mobile Radio: Sayısal Mobil Radyo, DCM-Digital Circuit Multiplication: Sayısal Devre Çarpımı)

Uygulama	Bit Hata Oranı	Kodlama Gecikmesi	PCM İletim ve Kodlama	Ses Bandı Veri İletimi	Diğer Ses olmayan Veri	Min. Konuşma Kalitesi
DMR ve portatifler	10^{-2}	70 msan	2 Asenkron	(-)	Tonlar	900 Mhz analog
Denizcilik	10^{-3}	60 ila	2	2.4	Tonlar	6 bit
Uydu Sistemleri		80 msan	Asenkron	kb/san kadar		PCM
DCM cihazları	10^{-3}	40 ila 80 msan	2 Asenkron	Evet	Tonlar	6 ila 7 bit PCM
ISDN	10^{-4}	(-)	4 Senkron	Hayır	Hayır	6 ila 7 bit PCM
Sayısal Kiralanmış Hatlar	10^{-4}	70 msan	(-)	(-)	Tonlar	7 bit PCM
Ses Saklama-İlet Sistemleri	10^{-4}	(-)	(-)	Hayır	Hayır	6 ila 7 bit PCM
Sayısal Anons Sistemleri	10^{-4}	(-)	(-)	Hayır	Hayır	7 bit PCM

3.4.1 Kalite ve Kapasite

Konuşma kalitesi ile bit hızı birbiri ile doğrudan çatışan iki faktördür. Konuşma kodlayıcının bit hızı ne kadar düşük ise, yani işaret sıkıştırılması ne kadar büyük ise, kodlanmış konuşmanın kalitesi o kadar düşük olur. Kamu telefon ağına ve bununla

ilintili ortamlara bağlanacak bir sistemin uyması gerekli kalite gereksimleri çok sıkıdır ve ITU-T tarafından belirlenmiştir. Bu sistemde genellikle yüksek kalitede bir kodlama, örneğin dalga şekli kodlama yöntemlerinden biri kullanılır. Buna rağmen özel telefon şebekeleri ve askeri sistemler gibi kapalı sistemlerde, kapasite gereksinimlerini azaltmak için kalite faktörü düşürülebilir. Fakat mutlak kalite genellikle belirlenmiştir ve genellikle diğer faktörlerin standardının yüksek tutulması ile ortalama bir seviyeye getirilir.

3.4.2 Kodlama Gecikmesi

Bir konuşma kodlama sisteminin gecikmesi, kaliteyi etkileyen önemli unsurlardan biridir. Kodlama gecikmesi algoritmik (konuşmanın analiz için depolanması), işlemsel (saklanmış konuşma örneklerinin işlenmesi için gereken süre) gecikmeler ve iletim gecikmesinden oluşur. Algoritmik ve işlemsel gecikmeler, konuşma kodlama sisteminin gecikmesini oluşturan temel faktörler gibi gözüксе de, konuşma bilgisinin tamamının işlenmesinden önce spektral parametrelerin hazır olur olmaz gönderilmesi söz konusu olduğunda (Avrupa Mobil Haberleşme Sistemi'nde olduğu gibi) iletim gecikmesi de hesaba katılmalıdır. Özellikle kamu telefon ağında, yankı probleminin üstesinden gelebilmek için gecikmenin en aza indirilmesi şarttır. Mobil ve uydu haberleşme sistemlerinde propagasyon gecikmeleri zaten var olduğu için yankı bastırma önlemleri alınmalıdır. Bununla birlikte gecikmenin çok küçük olduğu kamu haberleşme sisteminde uzun gecikmeli kodlayıcılar kullanıldığı takdirde ilave yankı bastırıcılar kullanılması gerekir ki, bu da sistemin genel maliyetini yükseltmektedir. Bir çok düşük bit hızlı kodlama yöntemi, 64 kb/san'lik standart sistem ile karşılaştırıldığında önemli bir kodlama gecikmesi getirmektedir. Örneğin Avrupa Sayısal Mobil Radyo (DMR) sisteminde gecikmenin üst sınırı 65 msan iken, 16 kb/san'lik ITU-T standardında gecikme en fazla 5 msan olarak tanımlanmıştır.

3.4.3 Bağışıklık (*Robustness*)

Birçok uygulamada kodlanmış konuşma bilgisi tipik olarak kanal kapasitesinin bir bölümünü işgal eder; kalan kapasite ise ileriye doğru hata düzeltme (*FEC-Forward*

Error Correction) ve işaretleşme için kullanılır. Hem yüksek oranda rastlantısal, hem de yığınsal (*bursty*) hataların oluşabildiği mobil haberleşmede, kodlama yönteminin içinde bulunan (*build-in*) kanal hatalarına karşı bağımsızlık özelliği, kabul edilebilir ortalama bir performansın elde edilebilmesi için gereklidir. Kodlamanın içinde bağımsızlık bulunması durumunda ise, ileriye doğru hata kontrolü daha az kullanılabilmekte ve ancak böylelikle daha iyi kalite için daha yüksek kaynak kodlama kapasitesi kalmaktadır. Kalite ve bağımsızlık arasındaki bu dengenin bir çok durumda elde edilmesi çok güçtür ve konuşma kodlama algoritmasının saptanmasında dikkate alınmalıdır. Mobil sistemlerin aksine; sabit linklerin bulunduğu fiber optik sistemlerde ise, kanaldan kaynaklanan hatalar azalmakta ve böylelikle kalite için bağımsızlıktan ödün verilebilmektedir. Bu da mobil/uydu sistemleri ile sabit linkli sistemler arasındaki temel farklardan biridir.

3.4.4 Karmaşıklık ve Maliyet

Kodlama algorimaları geliştikçe, hesap karmaşıklığı da artmaktadır. Sayısal İşaret İşlemcilerin (*Digital Signal Processor-DSP*) ve Uygulamaya Özgü Tümdevrelerin (*Application Specific Integrated Circuit-ASIC*) kullanımı sonucu hesaplama gücünün maliyeti ucuzlamıştır. Buna rağmen, karmaşıklık/güç harcaması ve fiyat, özellikle donanımın taşınabilirliği söz konusu olduğunda (örn. GSM uç cihazları), hala temel sorunlar arasında yer almaktadır. Güç harcamasını en aza indirmek ve aynı zamanda kanal verimini arttırmak için uygulanan tekniklerden biri de Sayısal Konuşma Aradeğerlemesi'dir (*Digital Speech Interpolation-DSI*). DSI tekniğinde gerçekte konuşmanın sadece yarı süresinde aktif konuşma olduğu; aktif konuşma bulunmayan zamanlarda kanal kapasitesinin başka amaçlar için kullanılabileceği ve böylelikle bu süre zarfında vericinin aktivitesini kısıtlayarak güç tasarrufu sağlanabileceği düşüncesinden yola çıkılmıştır. DSI sisteminin önemli bir parçası olan ses sezici (*Voice Activity Detector-VAD*) alt sistemi, konuşmayı sessizlikten ayırmaya yarar ve sistemin verimli ve güvenilir olarak çalışabilmesinde önemli bir rol oynar.

3.4.5 Kaskat Bağlantı ve Yeniden Kodlama

Herhangi bir kodlama algoritması için haberleşme sisteminin uçtan uca kullanıcıya yansıyan kalitesi, o algoritmanın hem kendi performansına, hem de diğer kodlama sistemleri ile verimli olarak arka arkaya gelebilmesine bağlıdır. Kaskat bağlantıdan meydana gelen bozulmalar genellikle toplamsaldır ve eğer herhangi bir algoritma belirli parametrelere sıkı şekilde bağlı ise meydana gelecek bozulma üstesinden gelinemez durum alabilir. Bu sorun çıkış konuşma işareti için sonradan filtreleme (*post-filtering*) uygulanan algorimalarda bugüne kadar tam olarak çözülememiştir. Bu gibi durumlarda PCM gibi başka bir kodlama sistemine geçmek, işaret kalitesini bozmakta ve ek maliyet getirmektedir.

3.4.6 Ses Bandında Veri İletimi

Sayısal ses iletim sistemleri genellikle modem, faks, veri işaretleri gibi diğer haberleşme işaretlerinin de taşınması için kullanılmaktadır. Dolayısı ile bir kodlama algoritmasının değerlendirme kriterlerinden birisi de o algoritmanın sayısal veriyi iletme kabiliyeti olmaktadır. Ses bandındaki ses dışı işaretlerin dalga şekli istatistikleri ve frekans spektrumları konuşma işaretlerine göre farklılıklar göstermektedir. Genellikle veri işaretlerinin algoritma tasarımının son aşamasında gözönüne alındığı ve bu nedenle de, eğer kodlama telefon haberleşme sisteminde kullanılacaksa, kullanıcı açısından verimsiz sistemler olduğu görülmektedir.

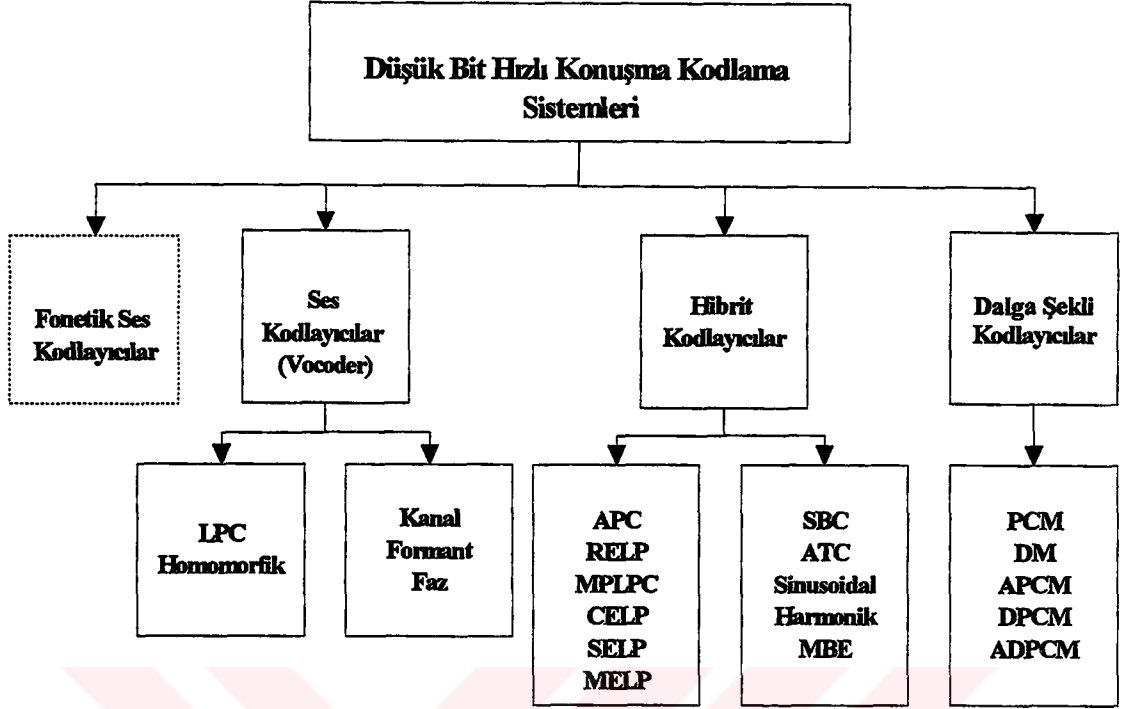
BÖLÜM 4

KODLAMA YÖNTEMLERİ VE STANDARTLARI

4.1 KODLAMA SİSTEMLERİNİN SINIFLANDIRILMASI

Son 30 yılda konuşmanın sayısal olarak kodlanması, araştırmacıların üzerinde çok çalıştığı konulardan biri olmuştur. Bu yoğun çalışmanın sonucu olarak, konuşmanın kodlanması için birçok yaklaşım ve strateji geliştirilmiştir. Yeni bulunan kodlama teknikleri olgunlaştıkça, belirli uygulamalara yönelik standartlaşma çalışmaları başlamıştır. Bütün bu yaklaşımlar ve stratejiler geliştirilirken, özel uygulama amaçlı standartlaşma çalışmaları da bunu takip etmiştir. Aşağıda konuşma kodlama teknikleri ve standartları ile ilgili temel bilgiler verilmekte; bunun yanı sıra yeni nesil kodlama standartlarının bugünkü gereksinimleri tartışılmaktadır. Böylelikle geçmişte ve bugün uygulanan çeşitli tekniklerin üstünlük ve sakıncaları ortaya çıkmaktadır.

Konuşma kodlama sistemleri temel olarak Şekil 4.1’de gösterildiği gibi sınıflandırılabilir. Bunlar arasında en çok Ses Kodlayıcılar, Hibrit Kodlayıcılar ve Dalga Şekli Kodlayıcılar üzerinde araştırma yapılmaktadır.



Şekil 4.1 Konuşma Kodlama Sistemlerinin Sınıflandırılması

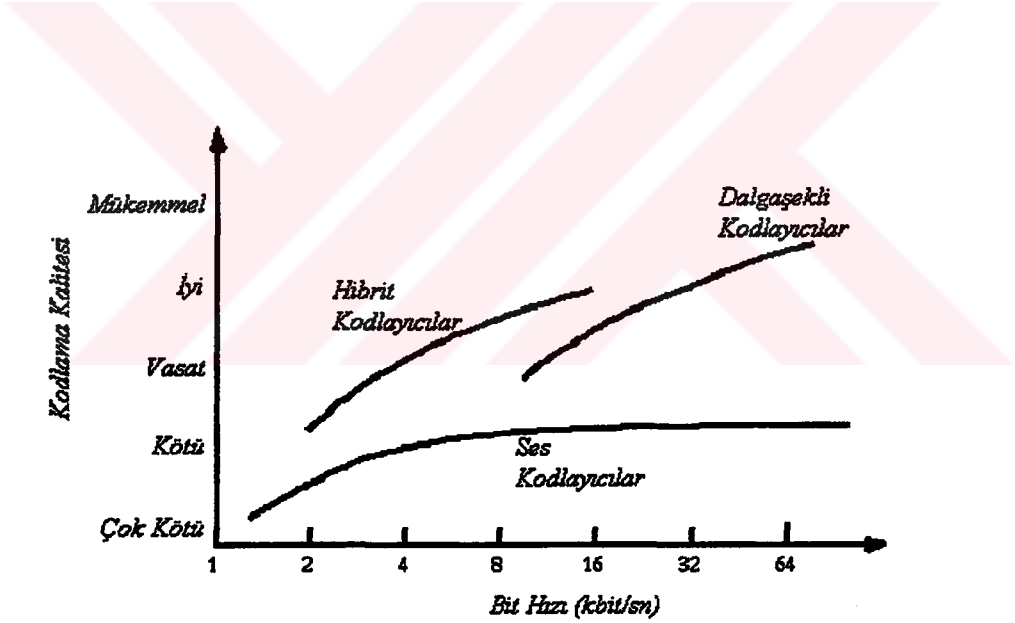
Hemen hemen bütün kodlama sistemlerinin temelinde işaretin analizi, fazlalıkların atılması ve esas bilgiyi içeren kısmın verimli bir şekilde kodlanmasının bulunduğu yukarıda belirtilmiştir. Kodlama kapasitesinin artırılması, fazlalıkların atılması ve bit atanması işlemlerinin yeterli verimlilikte gerçekleştirilmesi ile mümkün olmaktadır.

Tablo 4.1 Konuşma Kodlama Uygulamaları

Hız (kb/san)	Uygulama	Kodlayıcı Tipi	İşletme Başlangıcı
64	PSTN (1.Jenerasyon)	Darbe Kod Modülasyonu (PCM)	1972
32	PSTN (2.Jenerasyon)	Adaptif Farksal PCM (ADPCM)	1984
16	PSTN (3.Jenerasyon)	Düşük Gecikmeli Kod ile Uyarılan Doğrusal Öngörü (LD-CELP)	1992
16	INMARSAT Standart B (Deniz Mobil)	Adaptif Öngörülü Kodlama (APC)	1985
13	-Avrupa Sayısal Mobil Radyo(DMR) -Hücreli Sistem(GSM)	Düzenli Darbe Uyarma- Uzun Sürekli Öngörü (RPE-LTP)	1991
9.6	Gök Telefonu (Skyphone)	Çoklu Darbeli Doğrusal Öngörülü Kodlama (MPLPC)	1990
8	Kuzey Amerikan Sayısal Mobil Radyo (DMR)	Vektör Toplamı Uyarmalı Doğrusal Öngörülü Kodlama (VSELP)	1992
6.7	Japon Sayısal Mobil Radyo(DMR)	VSELP	1993
6.4	INMARSAT Standart M (Kara Mobil)	Çoklu Band Uyarma (MBE)	1993
5-6	Yarı-Hızlı GSM	Kod ile Uyarılan Doğrusal Öngörü (CELP)	1994
4.8	Amerikan Hükümeti Federal Standardı	CELP	1991
4.8	NASA MSAT-X (Kara Mobil)	Vektör Adaptif Öngörülü Kodlama (VAPC)	1991
2.4	Amerikan Hükümeti Federal Standardı	Doğrusal Öngörülü Kodlama (LPC-10)	1977
2.4	Amerikan Hükümeti Federal Standardı	Karışık Uyarmalı Doğrusal Öngörülü Kodlama (MELP)	1997

Tablo 4.1’de konuşma kodlamasında halen kullanılmakta veya geliştirilmekte olan yöntemler verilmiştir. Burada dikkat edilmesi gereken, PCM teknolojisinin yaygın olarak kullanılmaya başlanmasından itibaren kanal kapasitesinden daha verimli bir şekilde faydalanmak için uygulamaya özgü yeni kodlama tekniklerinin hızla gerçekleştirilmiş olmasıdır.

Üzerinde en çok çalışıldığı belirtilen üç kodlama grubu; yani Dalga Şekli Kodlama, Ses Kodlama ve Hibrit kodlama için kalite ile bit hızının ne şekilde değiştiği Şekil 4.3’deki diyagramda görülmektedir. Diyagramda kodlamanın kalitesi için subjektif bir kriter olan, ortalama değerlendirme puanı (MOS-Mean Opinion Score) kullanılmıştır. MOS değeri 1 ila 5 arasında değişebilmekte ve 1=çok kötü, 2=kötü, 3=vasat, 4=iyi, 5=mükemmel değerlerini alabilmektedir.



Şekil 4.2 Kodlama Yöntemlerinin Kalite Açısından Karşılaştırılması

4.2 KODLAMA SÜRECİ

Aşağıda örnekleme, nicemleme ve kodlamadan oluşan kodlama süreci kısaca incelenerek, Şekil 4.1’de verilmiş olan konuşma kodlama yöntemleri tanıtılacaktır. Burada gözden kaçırılmaması gereken nokta, vericide uygulanan aşamaların (örnekleme, nicemleme ve kodlama) tersinin de alıcıda uygulanacağıdır. Bu nedenle aşağıda açıklanan kodlama yöntemlerinde hem verici (kodlayıcı), hem de alıcı (kod çözücü) bloklar ele alınmıştır.

4.2.1 Nicemleme

Zamanda sürekli bir işaret olan konuşma işaretinin sayısal sistemler tarafından işlenebilir hale getirilmesi için zamanda ayrık bir işarete dönüştürülmesi gereklidir. Bilindiği gibi bu işleme örnekleme (*sampling*) adı verilir. *Nyquist* örnekleme teoremi [1][2], herhangi bir işaretin içerdiği maksimum frekans bileşeninin iki katı frekans ile örneklenmesi durumunda alıcıda kayıpsız olarak yeniden elde edilebileceğini göstermiştir.

Örnekleme işleminden sonra, elde edilen örnek değerlerinin, kapasiteyi etkin kullanmak üzere belirli bir ölçüte göre yuvarlatılması gerekir. Bu işleme nicemleme (*quantization*) adı verilir. Nicemleme ayrık zamanlı sürekli değerli bir işaretten, ayrık zamanlı ayrık değerli işaret elde etme işlemidir. İşaretin her örneğine ait değer, sonlu elemanlı bir kümenin elemanlarından bu değere en yakın olanı ile temsil edilir. Nicemlenmemiş işaret ile nicemlenmiş işaret arasındaki farka nicemleme hatası adı verilir ve bu hatanın genliği en aza indirilmeye çalışılır. Bütün nicemleme yöntemleri giriş işaretine göre nicemleyici çıkışının optimum olmasını sağlamaya çalışır ve en yüksek kaliteyi en düşük bit hızı ile sağlamayı hedefler. Fakat pratikte nicemleyicinin yeterliliği arttıkça karmaşıklığı, dolayısı ile de maliyeti artar.

4.2.1.1 Doğrusal Nicemleme

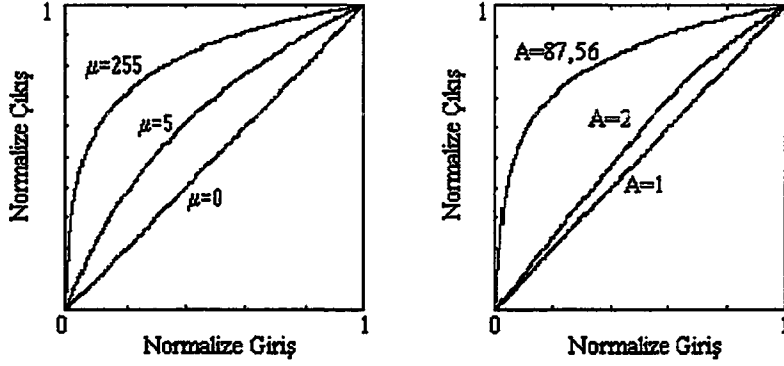
Doğrusal veya düzgün nicemleyicilerde nicemleme seviyelerinin arasındaki mesafe eşittir. Bu türden nicemleyiciler işlenen işaretin doğası ile ilgili hiç bir varsayımda bulunmaz. Bu nedenle performansları düşük olmasına karşın, gerçekleştirilmeleri son derece kolay ve ucuzdur. Telefon kanalındaki konuşmanın bu türden bir nicemleyici ile işlenmesi durumunda, yeterli kalitede bir konuşma elde etmek için 13 bitlik (8192 seviye) bir lineer nicemleyici kullanmak gereklidir.

4.2.1.2 Logaritmik Nicemleme

Konuşma işaretlerinin dinamiğinin 60 dB'den daha büyük olduğu düşünüldüğünde, düzgün nicemleyici kullanarak yeterli kalitede bir işaret elde etmek için çok fazla nicemleme seviyesi kullanılması gerekeceği açıktır. Aslında nicemleyici rezolüsyonu konuşmanın düşük genlikli kısımları için daha önemlidir. Bundan dolayı düzgün nicemleyicinin nicemleme seviyesinin, dolayısı ile de bandgenişliğinin verimsiz kullanılmasına yol açtığı ortadadır. Bu durum işaret genliği arttıkça nicemleme seviyelerinin arasındaki mesafenin arttırılması ile çözümlenir.

Yukarıda sözü edilen işlemi gerçekleştirmek için işaret öncelikle logaritmik karakteristikli bir sıkıştırıcıdan (*compressor*) geçirilir. Bu sıkıştırılmış işarete ise düzgün nicemleme uygulanır. Sistemin çıkışında elde edilen işaret sıkıştırıcı karakteristiğinin tersi uygulanarak açılır. Bu tekniğe “sıkıştırma-açma” (*companding*) adı verilir. Buradaki temel özellik, açıklanan tekniğin oldukça basit ve ucuz olarak gerçekleştirilebilmesi, ayrıca ses olmayan işaretlere de uygulanabilmesidir.

En popüler iki sıkıştırma-açma yöntemi olarak A yasası ve μ yasası gösterilebilir. Bunlar birbirine çok benzemektedir ve transfer karakteristikleri Şekil 4.3'de gösterilmiştir.



Şekil 4.3 μ ve A yasasına ait transfer karakteristikleri

Yukarıda gösterilen transfer karakteristikleri aşağıdaki gibi ifade edilebilir:

A Yasası:

$$c(x) = \begin{cases} \frac{A|x|}{1 + \log A} \operatorname{sgn}(x) & 0 \leq |x|/x_{\max} \leq 1/A \\ x_{\max} \frac{1 + \log(A|x|/x_{\max})}{1 + \log A} \operatorname{sgn}(x) & 1/A < |x|/x_{\max} \leq 1 \end{cases} \quad (4.1)$$

μ Yasası:

$$c(x) = x_{\max} \frac{\log(1 + \mu|x|/x_{\max})}{\log(1 + \mu)} \operatorname{sgn}(x) \quad (4.2)$$

Bir çok telefon sisteminde $A=87.56$ ve $\mu=255$ alınır. Bu sabitler kullanılarak 8 bitlik logaritmik bir nicemleyici ile telefon haberleşmesinde yeterli konuşma kalitesi elde edilebilir.

4.2.1.3 Düzgün Olmayan Nicemleme

Düzgün nicemlemenin temel problemi olan, işaret genliği azaldıkça işaret gürültü oranının azalması problemi logaritmik nicemleyici kullanımı ile kısmen çözülür. Bununla birlikte eğer giriş işaretinin olasılık yoğunluk fonksiyonu biliniyorsa, nicemleme seviyeleri olasılık yoğunluk fonksiyonuna uydurularak ortalama karesel nicemleme hatası en aza indirilebilir. Böylelikle nicemleme seviyeleri, giriş işaretinin olasılığı daha fazla olan değerleri civarında daha sık, diğer yerlerde daha seyrek olacaktır. Pratikte nicemleyici tasarımında olasılık yoğunluk fonksiyonunun öngörüsü kullanılabilir. Bu öngörü, nicemlenecek işarete ait geniş bir veri bankasının incelenmesi sonucunda oluşturulur. Daha sonra ise iterasyon metodu kullanılarak eldeki veriden nicemleme seviyeleri çıkartılır.

4.2.1.4 Vektör Nicemleme

Yukarıda incelenen nicemleme yöntemlerinde işareten alınan her örnek, komşu örneklerden bağımsız olarak değerlendirilmiştir. Aslında komşu örneklerin de dikkate alınması sistem performansını artırarak bozulmayı (*distortion*) azaltmaktadır. Her koşulda işlenecek verinin N 'lik bloklar halinde ele alınması daha elverişlidir. Bu uygulama daha önce ele alınan skalar nicemleme yöntemlerinin bir uzantısı olarak düşünülebilir. Skalar nicemlemede giriş örneği reel sayı ekseninde bir değer olarak ele alınır ve önceden belirlenmiş bir değere yuvarlatılır. Diğer yandan vektör nicemlemede ise N örneklik bir blok N boyutlu vektör olarak ele alınır ve N boyutlu uzayda önceden belirlenmiş bir değere yuvarlatılır.

Konuşma kodlamadaki ana hedef, aslına sadakatten (*fidelity*) en az ödün vererek veriyi sıkıştırmaktır. Klasik konuşma kodlama tekniklerinde, basitlikleri ve transmisyon hızı yeterince yüksek olduğunda iyi performans vermeleri sebebiyle ölçekli nicemleme (*scaled quantization-SQ*) kullanılır. Ancak, düşük bit hızlarında, genelde örnek başına işlenecek bit sayısı 1'den az olduğundan, SQ pratik değildir. Bu nedenle, hem kapasite sıkıştırmasını sağlamak, hem de yüksek bir sadakat

derecesini korumak için daha etkin ve üstün nicemleme yöntemleri gereklidir. Konuşma kodlamada popüler bir çözüm vektör nicemleme (*vector quatization-VQ*) uygulamasıdır [19]. VQ kullanımını teşvik eden ana unsurlardan biri, skalar kodlama yerine vektör kodlamanın daha iyi bir performans sağlayacağını gösteren *Shannon*'un hız distorsiyon teorisinin sonuçlarıdır.

VQ daha iyi bilinen SQ'nun genelleştirilmiş bir şeklidir ve şu şekilde ifade edilebilir: $\vec{x} = [x_1, x_2, \dots, x_N]$ bileşenleri $\{x_k\}_{k=1}^N$ olan gerçek değerli, sürekli genlikli rastgele değişkenler olan N boyutlu bir vektör olsun. VQ'da, $\vec{x}$ başka bir gerçek değerli, ayrık genlikli, N boyutlu $\vec{y} = [y_1, y_2, \dots, y_N]$ vektörü üzerine eşlenir. Böylece nicemleme gerçekleşmiş olur :

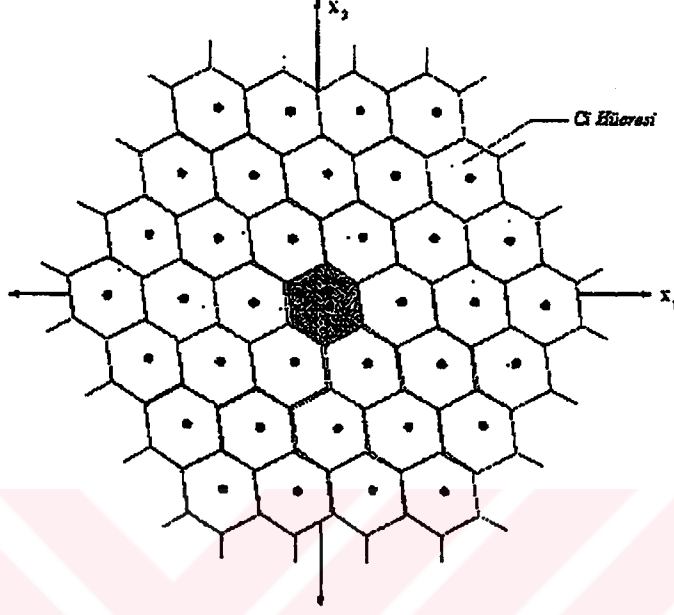
$$\vec{y} = Q(\vec{x}) \quad (4.3)$$

Burada, $Q(.)$ nicemleme operatörüdür. Tipik olarak, $\vec{y}$, bir $Y = [\vec{y}_1, \vec{y}_2, \dots, \vec{y}_L]$ kod kitabından (L , kod kitabındaki girdi sayısıdır) bir sonlu değerler kümesini alır. Kod kitabının büyüklüğü genelde ikinci dereceden üsteldir ve bu sebeple de vektör nicemleyicinin oranı $R = \log_2 L$ bit/vektör ve $r = R/N$ bit/örnektir. r 'den, kesirli bit/örnek oranının VQ'da mümkün olduğu görülebilir. Yukarıdaki VQ formülünde, Y kod kitabının nasıl oluşturulduğu ve N ve L 'nin gerekli performansı elde etmek için ne büyüklükte olması gerektiği oldukça önemlidir.

- Kod Kitabı Oluşturma

Shannon'un teorisi VQ kullanımını teorik olarak desteklemesine rağmen, VQ kodlayıcılar için maalesef yapıcı tasarım teknikleri sağlamaz. *Lloyd*'un optimum SQ üzerindeki çalışmalarının belleksiz VQ'lara genişletilmesi, bazı iyi bilinen ve yararlı algoritmalar (yani, *Linde-Buzo-Gray* (LBG) algoritması gibi) ortaya çıkartmıştır [20], fakat bellekli VQ (yani, VPQ vektör öngörü nicemlemesi ve sonlu VQ gibi geri beslemeli VQ) için kanıtlanmış algoritmalar henüz geliştirilme aşamasındadır.

L seviyeli kod kitabı tasarlanırken N boyutlu alan L adet $\{C_i\}_{i=1}^L$ hücreye bölünür, ve $\vec{x}$ C_i 'de olduğu takdirde her bir C_i hücresine bir $\vec{y}$ vektörü tayin edilir. Bu, Şekil 4.4'de açık olarak gösterilmektedir.



Şekil 4.4 İki boyutlu alanda bir vektör nicemleme örneği

Uzun zamanlı örnek ortalaması veya pratikteki beklenti her bir L seviyesi için minimuma indirilirse nicemleyicinin optimum olduğu söylemek mümkündür :

$$E = \frac{1}{M} \sum_{i=0}^{M-1} d(\vec{x}, \vec{y}) \quad (4.4)$$

Burada, $d(.)$ henüz belirlenmemiş distorsiyon ölçütü ve M de veri tabanındaki vektör sayısıdır. Nicemleyici için optimum bulmanın iki temel koşulu bulunmaktadır:

- i) Nicemleyicinin, $\vec{x}_i$ 'ye göre minimum distorsiyonu oluşturan $\vec{y}_i$ kod vektörünü seçmesi.

- ii) C_i hücresindeki ortalama distorsiyonu en aza indirmek için her bir $\vec{y}_i$ kod vektörünün seçilmesi ve böylece $\vec{y}_i$ 'nin C_i hücresinin ortağı (*centroid*) olması.

Birinci koşul için 4.5 eşitliğinde verilen ortalama hataların karesi (*Mean Square Error-MSE*) distorsiyon ölçütü çok sıklıkla kullanılmaktadır :

$$d(\vec{x}, \vec{y}) = \sum_{k=1}^N (x_k - y_k)^2 \quad (4.5)$$

MSE ölçütü pek çok durumda nesnel olarak anlamlı olmamasına rağmen hesaplaması çok basittir. $\vec{x}$ 'in elemanlarına eşit olmayan ağırlık veren değiştirilmiş bir MSE modeli tanımlanabilir. DÖK yapıları için ortak olarak kullanılan distorsiyon ölçütü *Itakura-Saito* distorsiyon ölçütüdür [24]. Ortağın hesaplanması yine kullanılan distorsiyon ölçütüne bağlıdır. Bu sebeple, MSE veya ağırlıklı MSE için $\vec{y}_i$ 'nin ortağı, C_i hücresindeki bütün M_i eğitim vektörlerinin ortalamasıdır :

$$\vec{y}_i = \frac{1}{M_i} \sum_{\vec{x} \in C_i} \vec{x} \quad (4.6)$$

Bir distorsiyon ölçütüne karar verildiğinde, kod kitabı, yukarıda bahsedilen algoritmalar kullanılarak oluşturulabilir veya düzenlenebilir [19]. Düzenleme sırası, tercihen, nihai kod kitabı büyüklüğünden çok daha büyük olmalı ve geniş bir işaret özellikleri sınıfına sahip olmalıdır; yani, konuşma için cinsiyet, yaş, konuşmanın içeriği, vs. gibi değişimler içermelidir.

Vektör nicemleme skalar nicemlemeye göre her zaman dah iyi sonuçlar vermektedir. Buna rağmen vektör nicemleme iletim hatalarına karşı çok duyarlıdır ve hesap yoğunluğu açısından skalar nicemlemeye göre çok daha karmaşıktır.

4.2.2 Zaman Bölgesinde Dalga Şekli Kodlama

Dalga şekli kodlayıcıları, işaretin dalga şeklinin alıcıda yeniden üretilmesini sağlar. Bunlar genellikle her türden giriş işareti için kullanılabilmek amacıyla, kodlanacak işaretten bağımsız olarak tasarlanır. Dalga şekli kodlamada bir darbe dizisi gelen işarete göre modüle edilir. Darbe dizisi ile çarpılarak elde edilen giriş işaretine ait örnekler nicemlenerek kodlanır. Dalga şekli kodlama hem zaman bölgesinde, hem de frekans bölgesinde uygulanabilir. Aşağıda zaman bölgesi dalga şekli kodlama yöntemlerine kısaca değinilmiştir.

4.2.2.1 Darbe Kod Modülasyonu (*Pulse Code Modulation-PCM*)

Dalga şekli kodlama tekniklerinin en sık kullanılanlarından birisi darbe kod modülasyonu PCM'dir. PCM aslında bir analog işaretin sayısallaştırılması sonucunda ortaya çıkan seri veri dizisinin genel adıdır. PCM'in haberleşmede özel uygulamalarda kullanılan birçok şekli bulunmaktadır. ITU-T'nin G.711 standardı ile belirlediği μ - yada A-tipi lineer olmayan nicemleme uygulanan PCM genellikle telefon haberleşmesinde kullanılmaktadır[1][2][7].

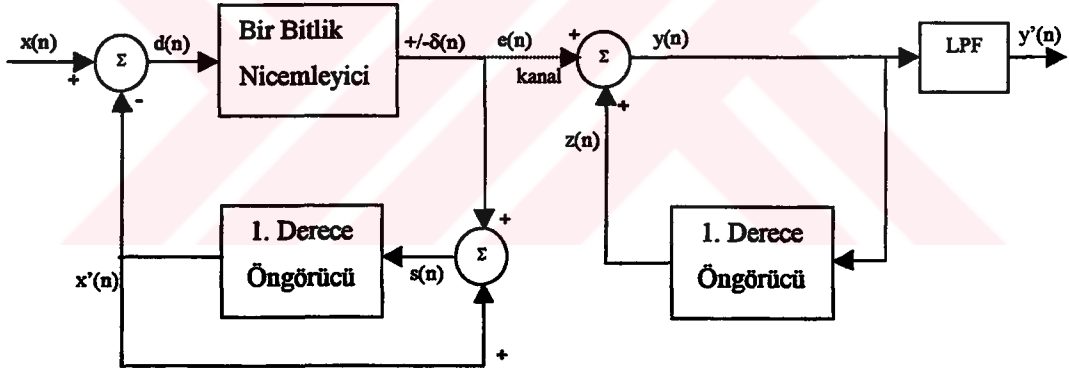
PCM modülasyonu uygulandıktan sonra elde edilen kodlanmış işaret hala temel bantta bulunmakta ve önemli miktarda DC bileşen içermektedir. Bu da elde edilen işaretin uzak iletim için uygun olmadığını gösterir. Bu nedenle bir PCM çıktısı genellikle daha yüksek frekanslı bir taşıyıcı dalga kullanılarak ve sayısal sürekli dalga modülasyonu tekniklerinden birisi (ASK-Genlik Kaydırmalı Anahtarlama-*Amplitude Shift Keying*, PSK- Faz Kaydırmalı Anahtarlama-*Phase Shift Keying* veya QAM- Karesel Genlik Modülasyonu- *Quadrature Amplitude Modulation*) uygulanarak uzak iletime uygun hale getirilir.

Genelde herhangi bir darbe kodlama sisteminde kanal kapasitesinin mümkün olduğu kadar yeterli kullanılmasına ve iletilen işaretin hatalara karşı bağımsızlığının artırılmasına çalışılır. PCM tekniği de karmaşıklığının azlığı, düşük gecikmesi ve

gürültüye karşı bağışıklığı nedeniyle haberleşme sistemlerinde sıklıkla kullanılmaktadır.

4.2.2.2 Delta Modülasyonu (Delta Modulation-DM)

Delta Modülasyonu kullanılarak PCM kodlamasında varolan gereksiz bitler ortadan kaldırılabilir. Delta modülasyonu kendi doğası nedeniyle iletimde gürültü nedeniyle oluşabilecek hatalara karşı daha az duyarlıdır. Delta modülasyonunda, giriş işaretinden alınan örnek önceki örneklerden toplanarak elde edilen sayısal değerden daha büyük ise sayısal değer bir artırılır. Aynı şekilde giriş örneği, toplanarak elde edilen sayısal değerden küçük ise bir azaltılır. Böylece her örnekleme anındaki çıkış değeri toplamsal değerın artırılmasına veya azaltılmasına göre bir yada sıfır olur. Delta modülatörü ve demodülatörüne ait blok diyagramlar aşağıdaki şekilde verilmiştir.



Şekil 4.5 Delta Modülatörü ve Demodülatörü

Kodlanacak analog işaret delta modülatörü girişine uygulanmadan önce, bir örnekle ve tut devresinden geçirilerek ayrık bir giriş işareti elde edilir. Başlangıç anında $x(n)$ 1 yada 0 olarak nicemlenir. Devredeki geribesleme elemanı, beklenen değer yani giriş değeri için, bir kestirimde bulunur. Delta modülatörünün temel avantajı, yalnızca $\pm\delta$ değerlerinin iletilmesi ve kodlanmış işaretin hızının darbe kod modülasyonuna göre $1/n$ oranında (n , PCM kodlamada kullanılan bit sayısı) daha

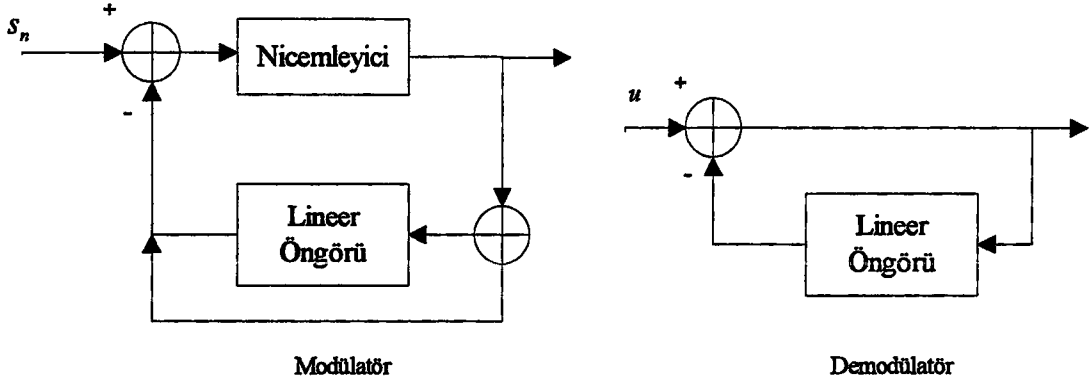
yavaş olmasıdır. Bu nedenle delta modülasyonu ile kodlanmış işaretlerin iletim sırasında karşılaşılabilecekleri gürültülere karşı bağışıklıkları daha fazladır.

Sabit adım boyu kullanılan delta modülasyonu genellikle *lineer delta modülasyonu* olarak adlandırılır. Lineer delta modülasyonu uygulamalarında karşılaşılan başlıca sorunlar, adım büyüklüğünün belirlenmesi ile ilgilidir. Adım boyu küçük olarak belirlenirse modüle edilmiş işaretin orijinal işareti takip etmesi zorlaşır, bunu engellemek için adım boyu büyük seçilirse o zaman da küçük genlikli giriş işaretleri için modülatör çıkışında osilasyon oluşur. Bu nedenle delta modülasyonu uygulamalarında örnekleme frekansı daha büyük seçilmek zorunda kalınır ve PCM'e karşı elde edilen avantajlar bir ölçüde kaybolur.

Delta modülasyonunda δ adım boyunun belirlenmesi çok önemlidir. Eğer giriş işaretinin karakteristikleri biliniyorsa bu problem değildir. Ancak insan konuşma işareti gibi önceden kestirimi daha güç olan işaretler için lineer delta modülasyonu uygun değildir. İnsan konuşma işaretinde sesli ve sessiz heceler arasında oldukça büyük genlik farkları meydana gelebilmektedir. Herhangi bir kelime içerisinde sesli ve sessiz hecelerin yanyana geldiği durumlarda bunu lineer delta modülasyonu ile kodlamak mümkün olmamaktadır. Buna çözüm olarak adım boyunun adaptif olarak belirlendiği *Adaptif Delta Modülasyonu* uygulamaları mevcuttur, ancak bu sefer de modülatör ve demodülatör yapıları oldukça pahalı hale gelmektedir.

4.2.2.3 Ayırık Darbe Kod Modülasyonu (*Differerantial PCM-DPCM*)

Ayrık darbe kod modülasyonu, delta modülasyonunun özellikle konuşma ve görüntü işaretlerinin kodlanması için geliştirilmiş, bu tip analog işaretlerdeki artık bilgiyi ortadan kaldıran, bir uzantısıdır. Ayrık kodlamada, ayrık giriş işareti ile önceden tahmin edilen değer arasındaki fark, p değerden birine nicemlenir ve sistemin karmaşıklığı öngörücünün (*predictor*) karmaşıklığı ile doğrudan orantılıdır. Genel DPCM yapısı Şekil 4.6'da gösterilmektedir.



Şekil 4.6 DPCM Modülör ve Demodülör Yapısı

DPCM Öngörücü genel olarak aşağıdaki ifade ile temsil edilebilir:

$$x'(n) = \sum_{R=1}^N h_R S(n-R) \quad (4.7)$$

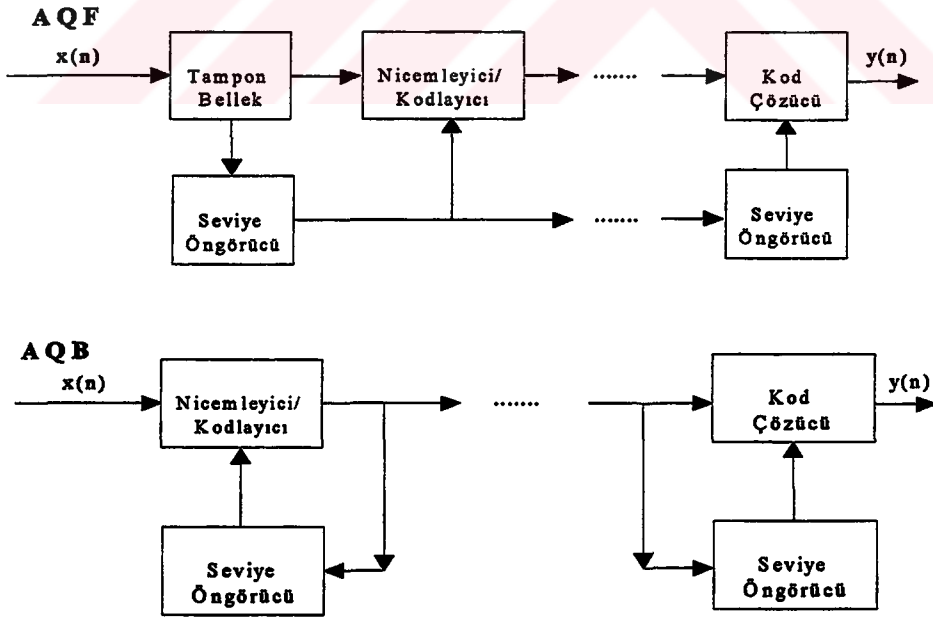
Burada h_R öngörücü katsayılar kümesidir. Eğer $h_1=1$ ve diğer katsayılar sıfır ise standart delta modülörü elde edilmektedir. Burada enteresan olan nokta öngörücü eşitliğinin FIR filtrelerin eşitliği ile aynı olmasıdır. Bu nedenle literatürde genellikle DPCM sistemi tarif edilirken, geribesleme çevrimindeki öngörücü yerine sayısal bir filtre konulmaktadır. Öngörücünün derecesi ile işaret gürültü oranı arasındaki ilişki kodlanan işarete göre değişmekte; genellikle konuşma işareti için $N=2$ 'de ve görüntü işareti için de $N=3$ 'de doymaya gidilmektedir [2].

Standart PCM yerine DPCM kullanmanın üstünlüğü, işaret gürültü oranının daha iyi olması, yani çıkış dalga şeklinin kalitesinin daha iyi olmasıdır. DPCM mutlak örnek değerini değil de, ardışık iki örnek arasındaki farkı kodladığından dolayı, kodlama için PCM ile aynı sayıda bit kullanmasına rağmen nicemleme hatası daha azdır. Bundan dolayı DPCM aynı sistem kaynakları ile daha duyarlı bir yapı oluşturmaktadır.

4.2.2.4 Adaptif Ayırık Darbe Kod Modülasyonu (*Adaptive DPCM-ADPCM*)

Kodlama algoritmasının başarımı, kodlanacak işaretin istatistiklerinin bilinmesine bağlıdır. Eğer elde açıkça tanımlanabilen bir giriş işareti varsa, zamandan bağımsız, sabit öngörücü yapıları kullanılabilir. Bir çok durumda, giriş işaretinin uzun dönemli istatistikleri bilinebilir; ancak işaret, kısa dönemde bu ortalamadan önemli ölçüde sapabilir. Bu gibi durumlarda adaptif kodlayıcı yapılarının önemli üstünlükleri vardır.

Adaptif DPCM terimi veya ADPCM iki farklı yapı için, yani nicemleyicinin uyumlaştırılması ya da öngörücünün uyumlaştırılması için kullanılır. Nicemleyicinin uyumlaştırılmasında giriş işaretinin dinamiğine bağlı olarak değişen üniform adım büyüklükleri kullanılır. Adım büyüklüğünün tahmini için iki farklı yöntem vardır: bunlar Şekil 4.7’de gösterilen *İleri Beslemeli Adaptif Nicemleme* ve *Geri Beslemeli Adaptif Nicemleme* yapılarıdır. (DPCM-AQF ve DPCM-AQB)



Şekil 4.7 DPCM-AQF ve DPCM-AQB Modülasyon ve Demodülasyon

İleri beslemeli adaptif nicemlemede (AQF) girişteki işaretin örnekleri tampon belleğe alınır ve girişin seviyesini tahmin etmek için kullanılır. Böylelikle nicemleme gürültüsünün etkisi azaltılmış olur. AQF'nin sakıncası, kodlanmış işaret ile birlikte seviye bilgisinin de iletilmesi gereksinimidir. Ayrıca bazı uygulamalarda AQF kodlamanın yarattığı ek gecikmeden dolayı kullanılamaz. Örneğin 4 kHz bandgenişliğine sahip bir konuşma işareti Nyquist hızında (8 kHz) kodlanırsa, seviye öngörmek için 256 örnek kullanan bir AQF algoritması çıkışın oluşmasında 16 msan'lik bir gecikmeye neden olur. Bu nedenle geri beslemeli adaptif nicemlemenin (AQB) kullanımı daha yaygındır.

Adaptif sistemler kullanılarak, adaptif olmayan sistemlere göre işaret-gürültü oranı önemli ölçüde arttırılabilir. Adaptif nicemleyiciler işaret-gürültü oranında adaptif olmayan düzenlere göre, 3 bitlik konuşma kodlama algoritmalarında bile 3 ila 7 dB arasında iyileşme sağlar. Aslında konuşma kodlama sistemlerindeki en yaygın yaklaşım, adaptif nicemleyici ile adaptif öngörücüyü tümleştirmektir. Uyumlaştırmamanın kalitesi her zaman olduğu gibi giriş işaretinin ortalamasını izleyen ve değişiklik olasılığını tahmin eden öngörücünün performansına bağlıdır. DPCM-AQF ve DPCM-AQB sistemleri 32 kbit/san hızı civarındaki uygulamalarda elverişli olarak kullanılabilir [2].

Yukarıda anlatıldığı gibi ADPCM uygulamasında hem nicemleyiciyi, hem de öngörücüyü ileri beslemeli veya geri beslemeli olarak uyumlaştırmak mümkündür. Dolayısı ile ortaya birbirinden farklı dört durum çıkmaktadır. ITU-T'nin G.721 standardı her iki yapı için de geri beslemeli uyumlaştırmayı öngörmektedir.

Kodlayıcının girişine standart 64 kbit/san'lik G.711 PCM işareti uygulanır. Kodlayıcı A veya μ yasasına göre kodlanmış olan giriş işaretini 32 kbit/san hızındaki ADPCM işaretine dönüştürür. Kodlayıcı bloğundaki format dönüştürücü A veya μ yasasına göre sıkıştırılmış PCM işaretini lineer yapıya dönüştürür; aynı blok, kod çözücüde de tersi işlemi yapar.

Kodlayıcıda elde edilen fark işareti, 4 bitlik adaptif nicemleyici kullanılarak nicemlenir. Öngörücüde geri beslemeli uyumlaştırma olduğundan dolayı, kodlayıcı

bloğunda ayrık işaret değerlerini yeniden elde etmek için ters adaptif nicemleyici bulunmaktadır. Daha sonra öngörücü önceden tahmin edilmiş $s_e(n)$ giriş değerlerini kullanarak, bir sonraki değeri bulur ve $d(n)$ fark işaretini bulmak için bu değeri giriş işaretinden çıkartır.

Geri beslemeli öngörücü kullanmanın avantajlarından biri de aynı öngörücü bloğunun hem kodlayıcıda, hem de kod çözücüde kullanılabilmesi ve böylelikle gerçeklemenin kolaylığıdır. Aynı avantaj adaptif nicemleyici için de geçerlidir. Bunun yanı sıra hem kodlayıcı, hem de kod çözücü bloğunda nicemleyicinin hızlı genlik değişimlerine (örn. konuşma işareti) hızlı; yavaş genlik değişikliklerine (örn. veri işareti) yavaş adapte olabilmesini sağlayan uyumlaştırma hız kontrol bloğu bulunmaktadır.

4.2.3 Frekans Bölgesinde Dalga Şekli Kodlama

Frekans bölgesinde çalışan dalga şekli kodlayıcılar, giriş işaretini bir dizi frekans bileşenine ayırır ve bunları birbirinden bağımsız olarak kodlar. Her frekans bileşenini kodlamak için kullanılacak bit sayısı dinamik olarak değişebilir. Aşağıda frekans bölgesi dalga şekli kodlama yöntemlerine kısaca değinilmiştir.

4.2.3.1 Adaptif Öngörülü Kodlama (*Adaptive Predictive Coding-APC*)

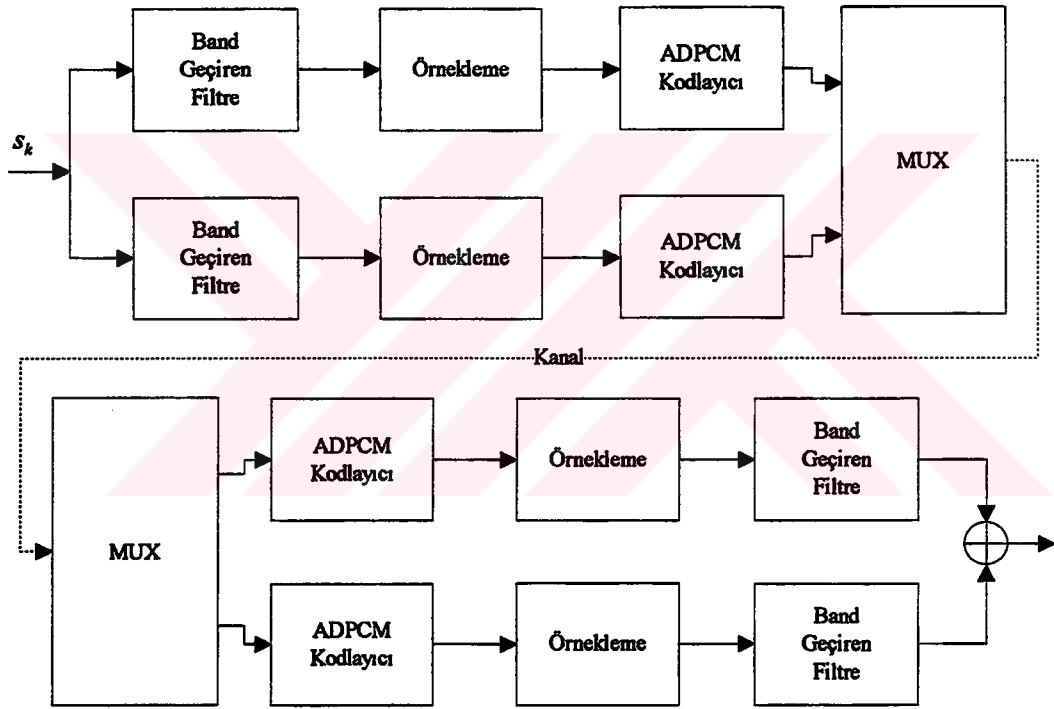
Adaptif öngörülü kodlama biçiminde konuşma işaretinin, içerdiği ortalama frekansa göre daha büyük bir periyod ile periyodik olduğu öngörülmektedir. Kodlanacak işaret yüksek ve alçak frekans bileşenlerine ayrılır ve iki öngörü algoritması uygulanır. Yüksek frekanslar “spektral” öngörücü ile alçak frekanslar ise “perde (*pitch*)” öngörücü ile tahmin edilir. Spektral öngörücü genellikle 4. dereceden seçilir ve katsayılar ADPCM’de olduğu gibi giriş işareti analiz edilerek bulunur. Perde öngörücünün girişine spektral öngörücünün çıkışı uygulanır.

APC konuşmanın 9 ila 16 kbit/san hızlarında kodlanması için kullanılmaktadır. En önemli dezavantajı ise kodlayıcının karmaşıklığıdır. Kodlayıcının kodlanacak

işaretin perde periyodunu tesbit etmesi, uygun spektral ve perde öngörücü katsayılarını bulması gerekmektedir. Buna rağmen, özellikle gürültülü ortamlarda iyi performans gösteren bir çok APC uygulaması bulunmaktadır.

4.2.3.2 Alt Band Kodlama (*Subband Coding*)

Alt band kodlama yukarıda anlatılan PCM bazlı tekniklerle birlikte kullanılan bir tekniktir. Bu yöntemde konuşmaya ait dalga şekli band geçiren filtreler kullanılarak alt bandlara bölünür ve her alt band Şekil 4.8’de görüldüğü gibi ADM, ADPCM, APC veya herhangi başka bir teknik kullanılarak kodlanır.



Şekil 4.8 Alt Band Kodlayıcı ve Kod Çözücü (iki band için)

Konuşma alt bandlara bölündüğü için her band içindeki nicemleme gürültüsü kendi içinde kalır ve diğer bandlar ile girişim yaratmaz. Kodlayıcı bölümde her alt bandta bulunan kodlama biti sayısını belirleyen örnek seyreltme (*decimation*) devreleri bulunmaktadır. Burada örnek seyreltmeden kasıt istenmeyen ve gereksiz olduğuna karar verilen örneklerin atılması işlemidir. Örneğin; giriş işaretini karakterize etmek

için sabit sayıda bit olduğu varsayılırsa ve sistemde aynı önemde iki alt band bulunuyorsa bitlerin yarısı bir banda, öteki yarısı diğer banda ayrılacaktır. Daha sonra her iki band da orijinal örnekleme frekansı ile örneklenirse izin verilenin iki misli bit sayısı elde edilir. Bu nedenle iki alt banddaki her iki örnekten birinin atılması gereklidir. Kodlamadan sonra iki alt band çoğullanarak işaret iletme hazır hale getirilir. Alıcıda (kod çözücüde) ise yeniden elde edilmiş işareti oluşturmak üzere alt bandlar filtrelenip toplanmadan önce, eksik örnekleri oluşturmak için aradeğerleme (*interpolation*) işlemi kullanılır.

9.6 kbit/san ile 32 kbit/san hızları arasında APC ile gerçekleştirilen alt band kodlama, ADPCM veya ADM kodlama yöntemleri ile boy ölçüşebilir. Buna rağmen alt band sayısına bağlı olarak sistemin karmaşıklığı artmaktadır. Bunun yanısıra, bandgeçiren filtreler de girişimi engelleyecek şekilde tasarlanmalıdır. Gerçeklemede genellikle QMF (*Quadrature Mirror Filter*) tipi filtreler kullanılarak hassas sistemler oluşturulabilmektedir.

Alt band kodlama geniş bantlı, yüksek kaliteli telekonferans uygulamalarında geniş bir kullanım alanı bulmuştur. Bu tip sistemlerde kullanılan kodlayıcı yapısı ITU-T G.722 standardında tanımlanmıştır.

Hareketli Resimler Uzman Grubu (*Motion Pictures Expert Group-MPEG*) tarafından önerilen ses sıkıştırma yöntemi aslında, adaptif nicemleyici içeren bir alt band kodlayıcıyı tariflemektedir.

Kodlayıcı gelen ses işaretini önce frekanslara göre 32 alt banda böler, daha sonra da her band 8 ila 24 milisaniyelik zaman çerçevelerine ayrılır. Psikoakustik (*psychoacoustic*) bir modelleme ile giriş işaretine göre her çerçeve ve alt band için kulağın nicemleme gürültüsüne hassasiyeti tesbit edilir. Bu bilgi ve elde bulunan iletim bandgenişliği gözönünde bulundurularak, en iyi subjektif ses kalitesini elde etmek üzere her bir kanal için nicemleme seviyeleri ayrı ayrı belirlenir. Her bir kanal için belirlenen nicemleme bilgisi kodlanmış örnekler ile birlikte kod çözücüye gönderilerek burada işaretin yeniden elde edilmesi sağlanır.

MPEG standardında, seviye 1,2 ve 3 olarak belirlenen üç karmaşıklık düzeyi bulunmaktadır. Ardışıl her seviye, kodlayıcı ve kod çözücüye ek karmaşıklık getirirken, aynı zamanda daha kaliteli işaret ve daha yüksek sıkıştırma oranı sağlar. İşaret kalitesi için elde edilen en yüksek standarda, “şeffaf kalite” adı verilir ve eğitilmiş bir dinleyicinin yeniden elde edilmiş işareti dinlerken kodlamayı farkedemeyeceği anlamına gelir. Bu algoritma ile 6:1 sıkıştırma oranında 2. seviye kodlama kullanılarak şeffaf kalite elde edilmektedir.

Birleşik-Stereo (*joint-stereo*) adı verilen kodlama yöntemi, stereo ses işaretlerine uygulanmaktadır. Bu yöntemde, sağ ve sol kanal arasındaki benzerlik kullanılarak işaret daha da sıkıştırılabilmektedir. Böylelikle daha yüksek sıkıştırma oranlarında bile şeffaf kodlama elde edilebilmektedir.

Her MPEG ses çerçevesi senkronlama için bir işaret ve konfigürasyon bilgisi içerir. Konfigürasyon bilgisi kodlayıcıda işareti sıkıştırmak için kullanılan parametrelerden oluşur. Bu parametreler iletim bit hızı ve ses örnekleme hızı, ses kanalı sayısı ve kodlama seviye numarası gibi bilgilerdir. Bu bilgi kullanılarak kod çözücü kodlayıcı konfigürasyonuna otomatik olarak uyum sağlar ve konfigürasyondaki değişiklikleri gerçek zamanda takip edebilir.

MPEG ses verisine seçilebilen bir veri dizisi eklenerek ses dizisine ait bir metin bilgisi veya kullanıcı sisteminin kontrol verisi gibi uygulamaya yönelik bilgiler iletilir. Bu veri için her ses çerçevesinde normalde kodlanmış ses verisinin gönderileceği yerden gereken kadar yer alınarak, gönderilecek veri için yer yaratılır.

4.2.4 Kaynak Kodlayıcılar (*Source Coders*)

Yukarıda anlatılan yöntemlerin hemen hepsi hem veri hem de konuşma işaretlerine uygulanabilir. Bu yöntemlerin yeterliliği kodlanacak işaretin uzun süreli istatistikleri hakkındaki bilgiye bağlıdır, fakat kodlanacak işaretin doğasını anlamaya yönelik herhangi bir yaklaşım içermez.

Kaynak kodlayıcılar, kodlanacak konuşma işaretinin insan kulağı tarafından algılanan spektral parametrelerini analiz eder. Bu parametreler daha sonra alıcıda ses işaretinin yeniden elde edilmesi için kullanılır. Alıcıda elde edilen işaret her zaman orijinal işaretin aynı olmaz, fakat aradaki fark insan kulağı tarafından algılanmaz veya fark önemli değildir. Bu da bir önceki bölümde incelenen dalga şekli kodlayıcıların felsefesine hiç uymamaktadır.

Kaynak kodlayıcılar 2.4 kbit/san ila 9.6 kbit/san hızları arasında konuşma işaretlerinin uygun şekilde kodlanmasını sağladıklarından oldukça popüler olmuştur. Özellikle iletim ortamında kısıtlı bandgenişliği olması veya saklama ortamında kısıtlı yer olması durumlarında (örneğin askeri uygulamalar) kaynak kodlayıcılar güvenle kullanılmaktadır. Yukarıda da söz konusu edildiği gibi bu kodlayıcılar konuşma işaretinin oluşum yapısını inceler ve kodlanacak işareten model parametrelerini bulmayı hedefler. Bulunan model parametreleri de karşı taraftaki alıcıya gönderilir. Konuşma içi kullanılan kaynak kodlayıcılara ses kodlayıcı (*vocoder*) adı verilir ve aşağıda açıklandığı gibi çalışır.

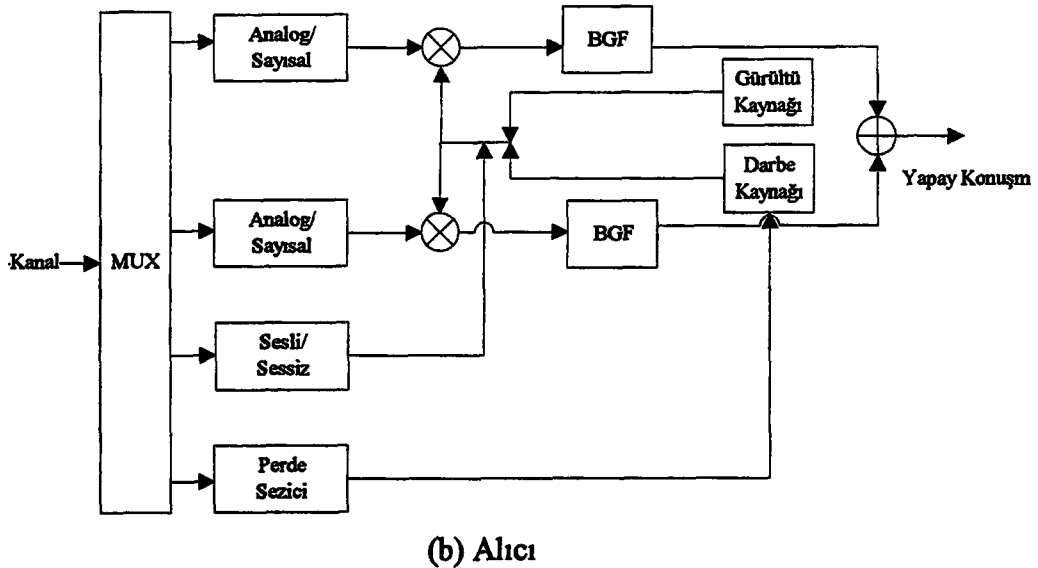
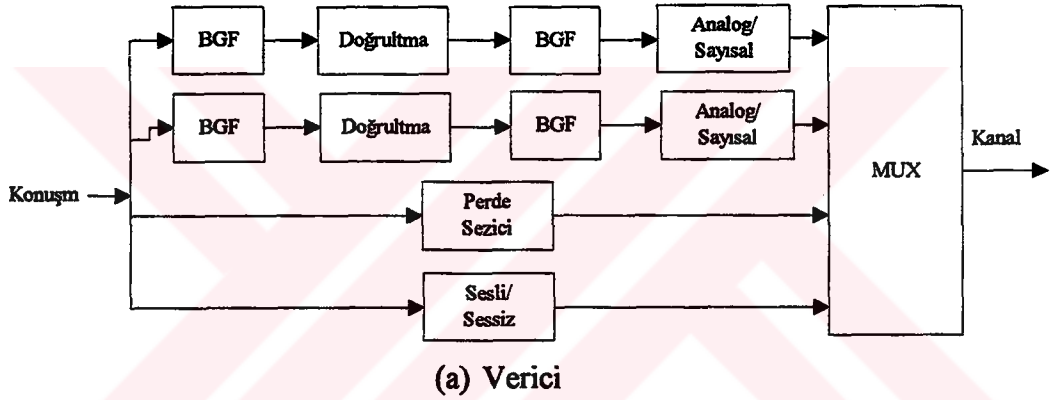
Ağızdaki ses oluşumunu sağlayan rezonans boşluğu (*vocal tract*) zamanla değişen bir filtre olarak ele alınır ve konuşmanın sessiz bölgeleri için beyaz gürültü, sesli bölgeleri içinse, frekansı konuşma işaretinin perde periyoduna uygun bir darbe katarı ile sürülür. Böylelikle alıcıya filtre bilgisi, sesli/sessiz kararı, uyarma işaretine ait değişim bilgisi ve sesli işaret için perde periyodu yollanması yeterli olur. Konuşmanın durağan olmayan karakterinden dolayı, gönderilen bilginin her 10-20 msan'de bir güncellenmesi gerekmektedir.

4.2.4.1 Kanal Ses Kodlayıcı (*Channel Vocoder*)

H.Dudley tarafından 1939 yılında önerilmiş olan bu kodlayıcı tipi, ses kodlayıcı yaklaşımlarının en eskilerindendir [18]. Bu kodlayıcı kulağın kısa süreli faz değişimlerine duyarsızlığından faydalanmaktadır. Yani yaklaşık 20 msan'lik konuşma işareti pencereleri için yalnızca spektruma ait genlik değişimi dikkate alınır. Bir filtre bankası yardımı ile işaretin spektrumu tahmin edilir. Burada, filtre

bankasında ne kadar çok filtre var ise öngörü o kadar iyi olmaktadır. Fakat filtre sayısı arttırıldığında ise iletilmesi gereken bit sayısı da artmaktadır. Daha sonra her filtre çıkışı doğrultularak, alçak geçiren filtreden geçirilir ve işaretin zarfı bulunur. Bulunan işaret örneklenerek alıcıya yollanır. Alıcı göndericinin tamamen ters işlemlerini yaparak, işareti yeniden elde eder. Kanal ses kodlayıcıya ait verici ve alıcı blok diyagramları Şekil 4.9'da verilmektedir.

Kanal ses kodlayıcıda kullanılan filtre bankalarının band genişlikleri frekans arttıkça artma eğilimi gösterir. Bu da insan kulağının frekansın logaritmik artışına lineer yanıt vermesi sayesinde kabul edilebilir. Kanal ses kodlayıcısı hem analog hem de sayısal donanım kullanılarak gerçekleştirilebilir ve 2.4 kbit/san civarında yüksek oranda anlaşılabilir konuşma üretebilir.



Şekil 4.9 Kanal Ses Kodlayıcı Blok Şeması (İki Kanal için)

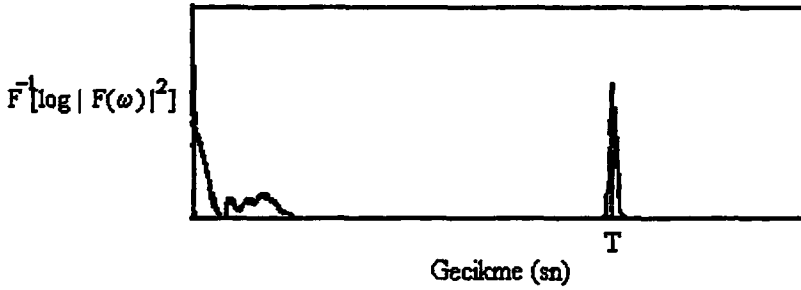
4.2.4.2 Homomorfik Ses Kodlayıcı (*Homomorphic Vocoder*)

Homomorfik işaret işleme, bileşik (*composite*) işaretler ile etkin olarak işlem yapabilen, genel amaçlı bir lineer olmayan işaret işleme tekniğidir. Homomorfik ses kodlayıcı; konuşma işaretini, ağızdaki ses oluşumunu sağlayan rezonans boşluğunun dürtü yanıtı ile uyarma fonksiyonunun zaman bölgesindeki bir konvolüsyonu olarak ele alır. Konvolüsyonun frekans bölgesinde çarpıma eşit olduğu ve spektrumun logaritması için, insan kulağının faz konusundaki zaafı da düşünüldüğünde (4.8) eşitliği yazılabilir:

$$\log(|S(e^{j\omega})|) = \log(|P(e^{j\omega})|) + \log(|V(e^{j\omega})|) \quad (4.8)$$

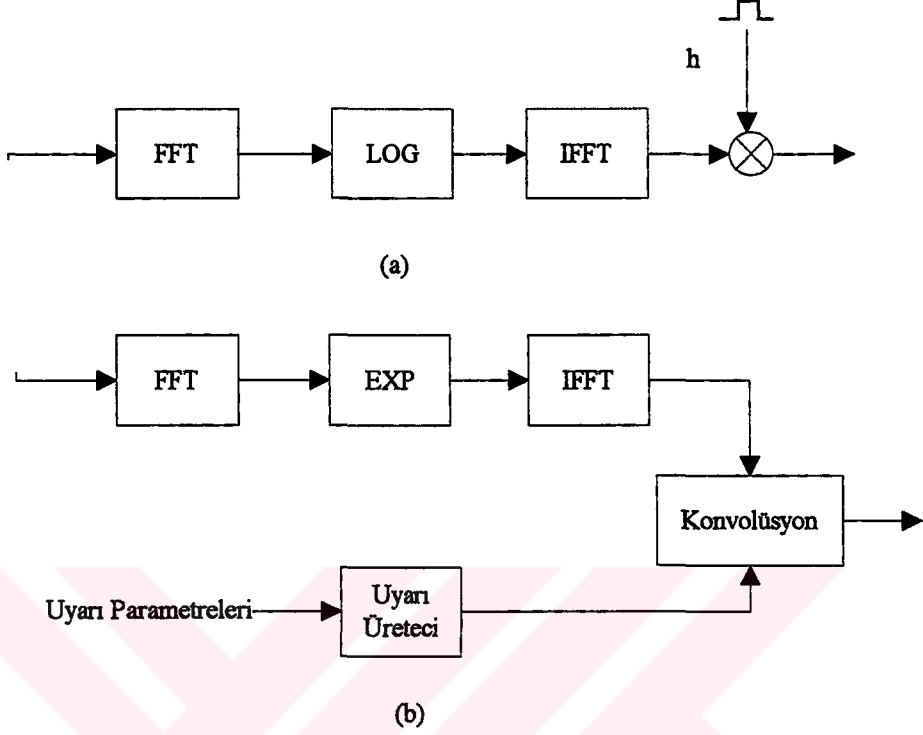
Burada $S(e^{j\omega})$ konuşma işaretinin, $P(e^{j\omega})$ uyarma işaretinin ve $V(e^{j\omega})$ 'da ağız rezonans boşluğunun spektrumudur. Bu işaret için kepsral (*cepstral*) katsayılar logaritmik spektrumun ters *Fourier* dönüşümünün alınması ile bulunabilir [8].

Ağızda konuşma üretilirken rezonans boşluğunun dürtü yanıtı, uyarma hızlı şekilde değişmesine rağmen, oldukça yavaş değişmektedir. Bu ayrım kepsral katsayılarda korunur ve Şekil 4.10'da görüldüğü gibi zaman kayması olarak kendini gösterir. Böylelikle konuşmaya ait iki önemli bileşenin (perde ve rezonans) katkıları alçak



Şekil 4.10 Perde bileşeninin gecikmesini gösteren tipik bir konuşma kepsstrumu

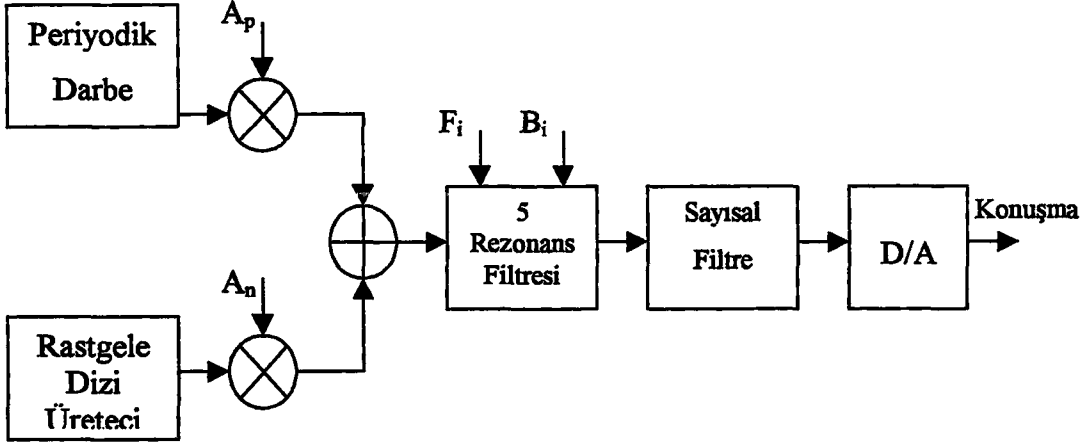
Homomorfik ses kodlayıcıda ağız rezonans boşluğunu ifade eden kepstral katsayılar Şekil 4.10'daki gibi, perde bilgisi ile birlikte gönderilir. Bu teknik 4 kbit/sn civarındaki kodlamalar için etkin olarak kullanılabilir.



Şekil 4.11 Homomorfik Ses Kodlayıcı; (a) Verici, (b) Alıcı

4.2.4.3 Rezonans Ses Kodlayıcı (*Formant Vocoder*)

Daha önceki bölümlerde söz konusu edildiği gibi, konuşma işaretindeki bilginin çok büyük bir bölümü, ağız rezonans boşluğunun rezonans frekanslarının pozisyonunda ve genişliğinde saklıdır. Bundan dolayı konuşma işaretine ait rezonans bilgisi hassas bir şekilde belirlenirse, kodlanmış işaret için çok düşük bit hızlarına erişmek mümkündür. Daha önce de değinildiği gibi erkek sesi için 5, kadın sesi için de 4 adet rezonans frekansı vardır. Bu yöntemde en önemli adım, analiz esnasında parametrelerin hesaplanmasıdır. Analiz sırasında konuşma işareti çeşitli filtreler kullanılarak frekans bandlarına ayrılır. Her banddaki sıfır geçişleri sayılarak karşı düşen rezonans frekansı bulunur. Doğru sonuçlara erişebilmek için, kullanılan filtrelerin performanslarının çok yüksek olması gerekmektedir [21].



Şekil 4.12 Rezonans Ses Kodlayıcı

Aslında bu teknik yardımı ile 1 kbit/san'nin altındaki hızlara kolayca inilebilir. Buna rağmen konuşma işaretine ait rezonans bilgisinin gereken hassasiyette bulunması son derece zor olduğundan, bu teknik hiç popüler olamamıştır.

4.2.4.4 Doğrusal Öngörülü Ses Kodlayıcı (*Linear Predictive Vocoder*)

Doğrusal öngörülü ses kodlayıcı yukarıda açıklanan tekniklerin kullandığı konuşma üretim modelinin aynısını kullanmaktadır [14]. Bu metoddaki tek değişik yan, ağız rezonans boşluğuna ait modelin aşağıdaki gibi tam kutuplu sonsuz dürtü yanıtı bir süzgeç ile ifade edilmesidir.

$$H(z) = \frac{G}{1 - a_1 z^{-1} - a_2 z^{-2} - \dots - a_p z^{-p}} \quad (4.9)$$

Diğer bir deyişle 4.9 ifadesinde verildiği gibi, her örnek kendinden öncekilerin lineer bir bileşimi olarak varsayılır. Bu filtreye ait katsayılar tahmin ile güncel örnek arasındaki farkı (hatayı) azaltacak şekilde hesaplanır.

Ses kodlayıcıda 20 msan uzunluğunda konuşma işareti saklanarak, öngörü katsayılarının belirlenmesi için analiz edilir. Bu katsayılar nicemlenerek, sesli/sessiz

bilgisi ve perde periyodu ile birlikte alıcıya gönderilir. Konuşma işareti aynı zamanda ağız rezonans boşluğu filtresinin tersinden geçirilerek tahmin hatası, ya da artık (*residue*) bulunur. Bu hata hem sesli/sessiz bilgisinin oluşturulması için kullanılır, hem de bir sonraki pencerede kullanılacak filtre katsayıları bu hatayı minimize edecek şekilde belirlenir [22].

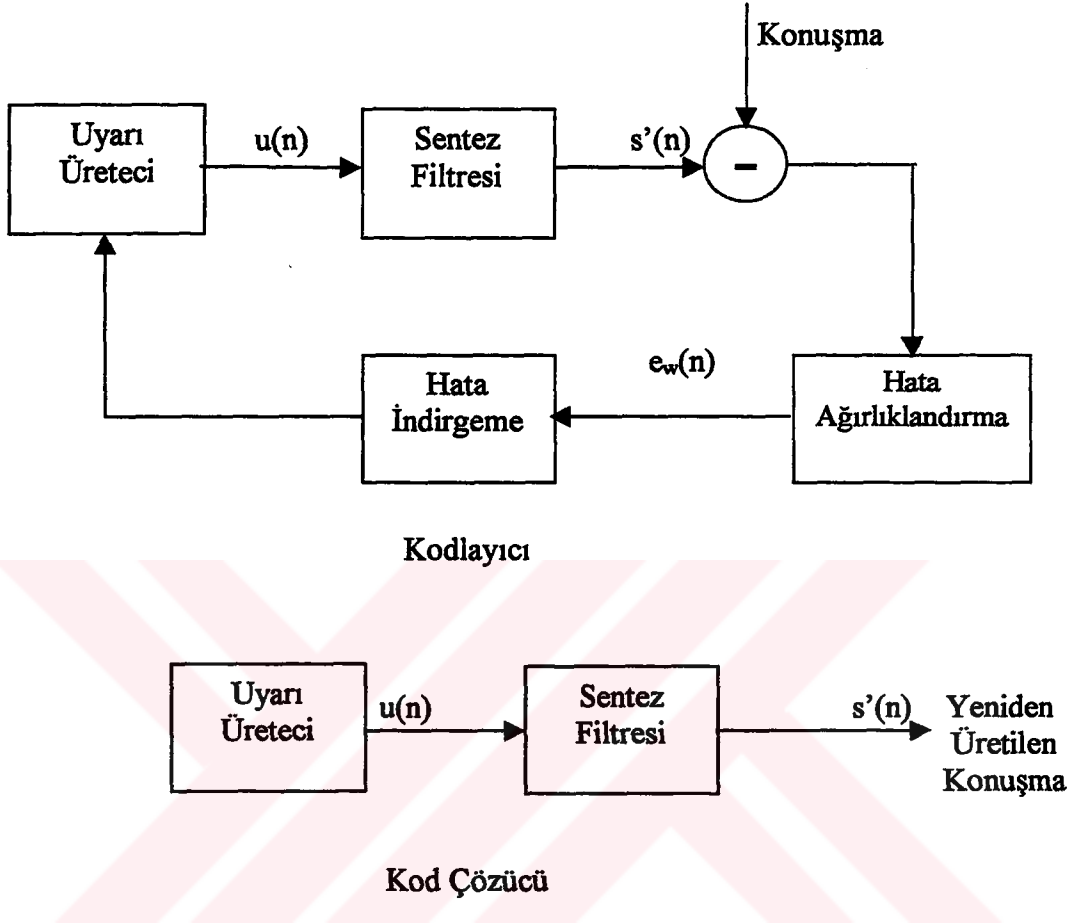
Doğrusal öngörülü ses kodlayıcılar, ağız rezonans boşluğunun tam kutuplu modeli oldukça iyi sonuç verdiğinden, ses kodlayıcılar arasında en popüler olanıdır. Bu tip kodlayıcılar ile 2.4 kbit/san civarına kadar inilebilmektedir. Ayrıca doğrusal öngörü modeli, ağız rezonans boşluğunun filtre katsayıları rezonans frekansları ile yakın ilişkili olduğundan, rezonans modelinin ilk katı olarak da kullanılabilir [18].

4.2.5 Hibrit Kodlayıcılar

Yukarıda görüldüğü gibi, konuşma kodlama için dalga şekli kodlama ve ses kodlama olarak iki temel yaklaşım bulunmaktadır. Dalga şekli kodlayıcılar kodlanacak işaretin dalga şeklini korumayı hedeflerken, ses kodlayıcılar giriş işaretine benzer nitelikte ses elde etmeye çalışır. Bundan dolayı dalga şekli kodlayıcılar orta hızlı (16 kbit/san civarı) kodlamalar için uygun olup düşük bit hızlarında kullanılamaz. Ses kodlayıcılar, düşük bit hızlarına inebilmekle birlikte, ürettikleri çıkış anlaşılabilir, fakat oldukça suni karakterdedir.

Hibrit kodlayıcılar, ses kodlayıcılar ile dalga şekli kodlayıcılar arasındaki boşluğu kapatmak üzere ortaya çıkmıştır. Yani hibrit kodlayıcılar düşük bit hızlarına erişmeyi hedeflerken, konuşmanın karakterine ilişkin bilgiyi de korumayı amaçlamaktadır. Başka yaklaşımlar da olmasına rağmen en başarılı ve sık kullanılan hibrit kodlayıcı tipi zaman bölgesinde sentezleyerek analiz (*Analysis-by-Synthesis Abs*) yöntemiyle çalışanlar olmuştur. Bu tip kodlayıcılar, doğrusal öngörülü ses kodlayıcıların kullandıkları ağız lineer öngörü filtre modelini kullanır. Fakat bu filtre için girişi oluştururken, basit iki durumlu sesli/sessiz yaklaşım modelini uygulamak yerine, yeniden oluşturulan konuşma işareti dalga şeklini mümkün olduğu kadar orijinal

işarete uydurmaya çalışır. Şekil 4.11’de blok şeması verilen *AbS* kodlayıcılar ilk olarak 1982 yılında *Atal* ve *Remde* tarafından ortaya atılmıştır.



Şekil 4.13 *AbS* Kodlayıcı ve Kodçözücü Modeli

Sentezleyerek analiz kodlayıcılar öncelikle giriş konuşma işaretini 20 msan’lik pencerelelere ayırır. Her pencere için sentez filtresinin parametreleri ve bu filtreyi uyaracak işaret belirlenir. Uyarma işareti belirlenirken, sentez filtresine verildiğinde orijinal konuşma işareti ve yeniden üretilen konuşma işareti arasındaki hatanın en aza indirilmesi kistas alınır. Böylelikle sentezleyerek analiz teriminin de kaynağı bulunmuş olur. Yani kodlayıcı, giriş konuşma işaretini birçok yaklaşımı sentezleyip, hatayı belli bir değerin altına indirmeye çalışarak, analiz eder. Sonuçta kodlayıcı her pencere için sentez filtresi parametrelerini ve uyarma işaretini karşı tarafa yollar. Kod çözücüde de alınan uyarı işareti sentez filtresinden geçirilerek, konuşma işareti

yeniden elde edilmiş olur. Sentez filtresi genellikle tam kutuplu, lineer bir filtredir ve transfer fonksiyonu aşağıda verilmiştir:

$$H(z) = \frac{1}{1 - \sum_{i=1}^p \alpha_i z^{-i}} \quad (4.10)$$

(4.10) ifadesinin paydası içinden orijinal konuşma işareti geçtiğinde artık işaretin enerjisini minimize ederek belirlenen hata filtresinin ifadesidir. Bu filtrenin derecesi p tipik olarak 10 seçilir.

Sentez filtresi, aynı zamanda sesli konuşma işaretlerine ait uzun dönemli periyodiklikleri ifade eden perde frekansını belirlemek için bir perde filtresi içerebilir. Alternatif olarak, işarettaki uzun dönemli periyodiklikler uyarı üreticine adaptif bir kod kümesi ekleyerek; yani “uyarı işareti $u(n)$, $Gu(n-\alpha)$ şeklinde bir bileşen içersin ve α ’da öngörülen perde periyodu olsun” denilerek çözülebilir.

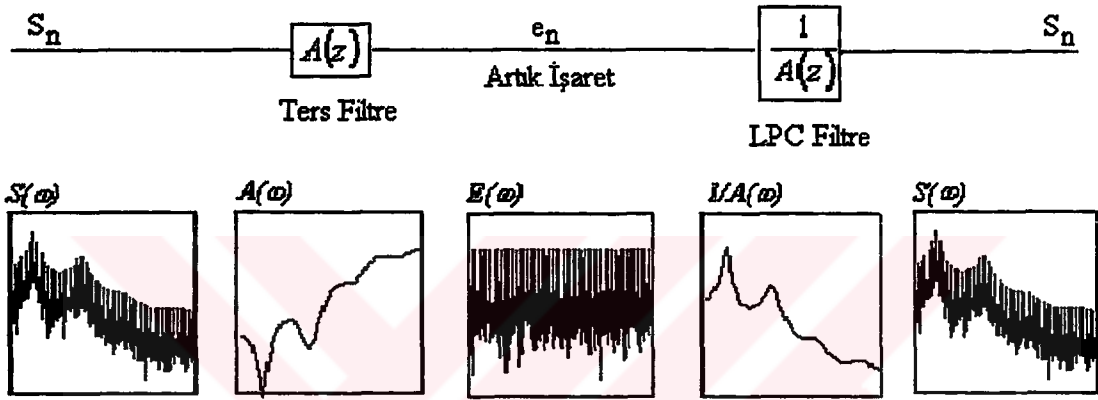
Hata ağırlıklandırma bloğu, hata işaretinin subjektif gürlüğünü azaltmak üzere spektrumunun şekillendirilmesi için kullanılır. Hata işareti, konuşmanın yüksek enerjili olduğu frekans bölgelerinde en azından kısmen konuşma tarafından maskelenecektir. Ağırlıklandırma filtresi hata işaretini konuşma enerjisinin düşük olduğu bölgelerde vurgulayacaktır. Böylelikle ağırlıklandırılmış hata işareti minimize edildiğinde hata işaretinin enerjisi konuşma enerjisinin yüksek olduğu bölgelerde yoğunlaşacaktır. Bundan dolayı hata en azından kısmen konuşma işareti tarafından maskelenecek ve bağıl önemi azalacaktır. Bu tip ağırlıklandırma yöntemleri AbS kodlayıcıları için yeniden üretilen konuşmanın bağıl kalitesini arttırmada önemli faydalar sağlamaktadır.

AbS kodlayıcıların ayırıcı özelliği sentez filtresi için uyarı işareti $u(n)$ ’in nasıl seçileceğidir. Kavramsal olarak yeniden elde edilen konuşma işaretinin nasıl olduğunu görmek için her türlü dalga şekli filtreden geçirilebilir. Orijinal ve yeniden üretilen konuşma işareti arasındaki minimum ağırlıklandırılmış hatayı veren uyarı, kodlayıcı tarafından seçilerek kodçözücüdeki sentez filtresini sürmek için kullanılır. AbS kodlayıcılara düşük bit hızlarında kaliteli kodlama imkanını veren, bu kapalı çevrim yaklaşımıdır. Fakat her mümkün uyarı işaretinin sentez filtresinden

geçirilmesinin yarattığı hesap yükü çok fazladır. Bu nedenle genellikle hesap yükünü azaltmak için uygulamada bazı basitleştirmelere gidilmektedir.

4.2.5.1 Artık İşaret Uyarmalı Doğrusal Öngörülü Kodlayıcı (RELP)

Konuşma işareti lineer bir öngörücüden geçirildiğinde örnekler arasındaki ilişki kaybolmaktadır. Eğer öngörü kalitesi yeterli kalitede ise, öngörücünün çıkışı beyaz gürültüye yakınsar ve Şekil 4.12'de görüldüğü gibi bir spektrum ortaya çıkar.



Şekil 4.14 Konuşma işaretinin ters filtrelenerek, LPC filtreden geçirilmesinin etkisi

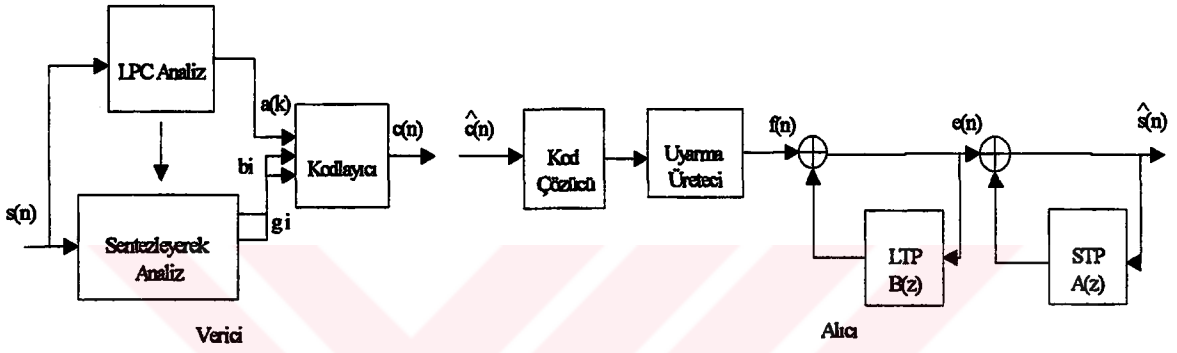
Yukarıdaki blok şemadaki artık işaretin spektrumu incelendiğinde, her bölüm için birbirine benzer bir yapı görülmektedir. Buna rağmen, bu artık işaret bütün uyarma bilgisini ve lineer öngörücünün kaçırdığı her türlü bilgiyi içerir. Bu bilgi ilgilenilen frekans bölgesinde aşağı yukarı uniformdur. Artık işaret ile uyarılan doğrusal öngörülü kodlamanın arkasındaki düşünce, yukarıda bahsedilen artık işaretin bir kısmını karşı tarafa ileterek, alıcıda tüm artık işaretin elde edilmesinin mümkün olmasıdır [21].

RELP kodlayıcısında artık işaret 1000 Hz kesim frekanslı bir alçak geçiren filtreden geçirilerek, elde edilen işaret daha önce anlatılan dalga şekli kodlama yöntemlerinden biri kullanılarak kodlanır. Alıcıda ise artık işaret, temel banddaki artık işaretin ilgilenilen bölgedeki diğer frekanslara kopyalanması ile oluşturulur. Bu

yapıdaki kodlayıcılar genellikle konuşmanın 9.6 kbit/san civarında iyi kalitede kodlanması için uygundur.

4.2.5.2 Çoklu Darbe ile Uyarılan Doğrusal Öngörülü Kodlayıcı (MPLPC)

MPLPC kodlamada uyarma işareti ağırlıklandırılmış ve geciktirilmiş darbeler dizisinden oluşmaktadır. Şekil 4.15’de MPLPC kodlamaya ait verici ve alıcının blok diyagramı verilmiştir.



Şekil 4.15 MPLPC Verici ve Alıcı

Verici tarafta konuşma işareti analiz edilerek LPC katsayıları, LTP(*Long Term Prediction*-Uzun Dönemli Öngörü) ve kaynak parametreleri elde edilir. Kaynak parametreleri giriş işaretinin içindeki darbelerin yerlerinden ve genliklerinden oluşur. Böylelikle uyarı işareti (1)’deki gibi ifade edilebilir:

$$f(n) = \sum_{k=1}^M \beta_k \delta(n - \gamma_k) \quad (4.11)$$

Burada γ_k darbelerin yerlerini, β_k ise buna ilişkin genlikleri göstermektedir. MPLPC için toplam uyarı dizisi ise (4.12)'deki gibidir:

$$e(n) = \beta_0 e(n - \gamma_0) + f(n) = \beta_0 e(n - \gamma_0) + \sum_{k=1}^M \beta_k \delta(n - \gamma_k) \quad (4.12)$$

Aynı şekilde β_0 ve γ_0 LTP parametreleri ve β_k ve γ_k ise $k=1, \dots, M$ olarak M adet darbenin yeri ve genliğidir. LTP parametreleri ve uyarı modelindeki her darbe ayrı bir dürtü olarak ele alınır ve bunlara ait yer ve genlikler sentezleyerek-analiz (analysis-by-synthesis) metodu ile bulunur. Konuşmanın sentezleyerek-analizinde öncelikle LTP parametreleri belirlenerek katkısı orijinal konuşma işaretinden çıkartılır. Daha sonra her darbenin parametreleri belirlenir ve katkısı konuşma işaretinden çıkartılarak, kalan işaret bir sonraki darbenin parametrelerini bulmak için kullanılır. Sabit bit hızlı bir kodlayıcı için genellikle çerçeve başına kullanılacak darbe sayısı önceden belirlenir. Değişken bit hızlı kodlayıcılarda ise, önceden belirlenen hata kriterini tutturmak için kullanılan darbe sayısı esnek olarak değiştirilebilir. MPLPC kodlayıcının parametreleri; LPC analiz katsayıları (pencere uzunluğu L , LPC çerçeve uzunluğu I , öngörücünün sınıfı P), hata ağırlıklandırma faktörü α , kaynak analiz çerçeve boyu N ve analiz çerçevesi başına darbe sayısı M' den oluşmaktadır [4].

Subjektif değerlendirme sonuçları MPLPC kodlayıcının 9.6 ila 16 kbit/san hızlarında oldukça kaliteli sonuçlar verdiğini göstermiştir. MPLPC kodlayıcıya ait sonuçlara göre, MPLPC modelinin orta bit hızlarındaki yeterliliği, LPC işaretinin hem periyodik hem de rastgele doğasını çok iyi modellemesinden kaynaklanmaktadır. Yapılan deneyler aynı zamanda LTP'nin, kodlayıcının tamamının başarısına etkisinin önemli olduğu sonucunu vermiştir. MPLPC kodlayıcıda kodlanması gerekli parametrelerin sayısının çokluğu (LPC katsayılar, LTP gecikme ve kazanç, ve darbelerin genlikleri ve yerleri) 9.6 kbit/san hızın altında performansın hızla düşmesine yol açmaktadır.

4.2.5.3 Düzenli Darbe ile Uyarılan Doğrusal Öngörülü Kodlayıcı (RPLPC)

MPLPC kodlayıcıda darbelerin pozisyonlarını sabitleyerek, yani parametre sayısını düşürerek RPLPC kodlayıcı elde edilmektedir. Bu nedenle RPLPC, MPLPC'nin dürtü darbelerinin zamanda eşit aralıklarla dağıldığı özel bir halidir. Analiz çerçevesi içerisinde ilk darbe herhangi bir yerde bulunabilir, ancak bir çerçeve içindeki darbe sayısı ve bunların aralıkları tamamiyle önceden belirlenmiştir. Böylelikle γ_1 belirlendikten sonra diğer darbe pozisyonları $\gamma_1 + kD$ olmaktadır. Burada D dürtü dizisi içindeki darbeler arasındaki boşluk olarak tanımlanmaktadır. Düzenli darbe ile uyarılan LPC'de kodlanan parametreler, ilk darbenin pozisyonu γ_1 , ve darbe genlikleri β_k 'lardır ($k=1, \dots, N/D$ N :çerçeve uzunluğu).

Analiz süreci, ağırlıklandırılmış hata enerjisinin en aza indirilmesi için ilk darbenin yerinin ve ilgili kazanç değerinin eşzamanlı olarak seçilmesi ile başlar. İlk darbe yeri sentezleyerek-analiz yöntemi ile bulunduktan sonra tüm darbe genlikleri N/D tane lineer denklemin çözülmesi ile elde edilir. Bilgisayardaki gerçeklemelerde analiz çerçeve genişliği 40 iken $D=4$ değeri için 9.6 kbit/san hızında oldukça kaliteli sonuçlar elde edilmiştir. Ayrıca darbelerin yerleri önceden belirlenip sabit tutulduğundan, hesap yükü olarak RPLPC MPLPC'ye göre daha az karmaşıktır.

Mobil telefon sistemi (*Global System for Mobile communication-GSM*) için kullanılan tam hızlı kodlayıcı da 13 kbit/san hızda çalışan bir RPLPC kodlayıcıdır [10]. GSM'de konuşma öncelikle 20 msan'lik çerçevelere ayrılır ve her çerçeve için 8 adet STP (*Short Term Prediction-Kısa Dönemli Öngörü*) katsayısı bulunur. Daha sonra her çerçeve 5 msan'lik alt çerçevelere ayrılır ve kodlayıcının uzun dönemli öngörüsü için her alt çerçevede bir gecikme ve kazanç bulunur. Sonuçta kısa dönem ve uzun dönem filtrelemelerinden sonra geriye kalan artık işaret her alt çerçeve için şu şekilde nicemlenir. 40 örneklilik artık işaret 13 örneklilik üç bölüme ayrılır. En yüksek enerjili dizi uyarma dizisini en iyi temsil eden örnek olarak alınır ve bu dizideki her darbenin genliği üç bitle kodlanır. Kodçözücüde ise, elde edilen uyarma işareti kısa dönemli ve uzun dönemli uyarma filtrelerinden geçirilerek, konuşma

işareti elde edilir. Yeniden elde edilen konuşma işaretinin anlaşılabilirliğini arttırmak için, en sonda bir filtreleme daha uygulanır.

GSM kodlayıcı hem gayet kaliteli sonuçlar vermekte, hem de diğer kodlayıcılardan çok daha basit bir şekilde gerçekleştirilebilmektedir.

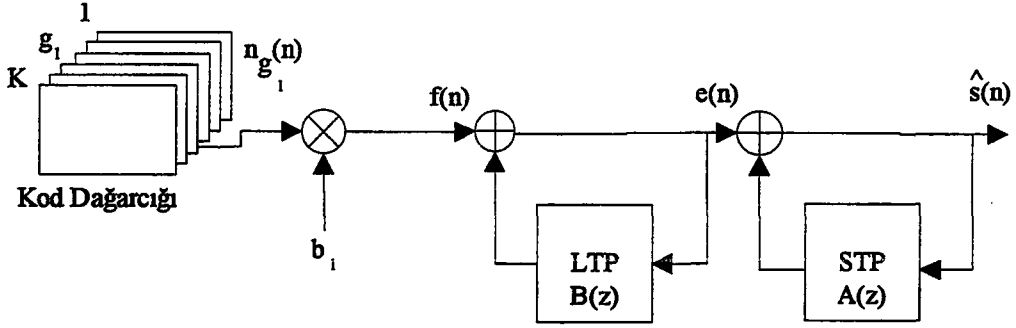
4.2.5.4 Kod ile Uyarılan Doğrusal Öngörülü Kodlayıcı (CELP)

Yukarıda verilen iki kodlama yöntemine göre CELP kodlayıcı 9.6 kbit/san hızının altında da oldukça yeterli olmakta ve kaliteli çıktılar üretmektedir. Kod ile uyarma dürtüyü modellemek için çok az sayıda parametre kullandığından, düşük hızlarda başarılı olmaktadır. Bu kodlama metodu da yine bir sentezleyerek-analiz kodlayıcıdır ve MPLPC’de olduğu gibi konuşma işaretinin spektral zarfını ve periyodik perdeli (pitch) yapısını modellemek için kısa süreli (STP) ve uzun süreli (LTP) öngörücüler kullanılmaktadır.

Kod ile uyarma modelinde, deterministik veya rastlantısal olarak belirlenen kod kelimelerinden oluşan göreceli olarak büyük bir kod dağarcığı kullanılmaktadır. CELP kodlayıcıda uyarma işareti aşağıdaki gibi ifade edilebilir:

$$e(n) = \beta_0 e(n - \gamma_0) + \beta_1 v_{\gamma_1}(n) \quad (4.13)$$

Burada $v_{\gamma}(n)$ ($\gamma=1, \dots, K$, $n=0, \dots, N-1$) K büyüklüğündeki kod dağarcığındaki γ indeksli N örnekli dizidir. N analiz çerçevesinin büyüklüğüdür ve genellikle 5msan olarak seçilir. Çerçeve büyüklüğünün artması kodlayıcı performansını azaltmaktadır. Kod dağarcığı boyu olarak 1024 tipik bir seçimdir, bu da kodlama için $\log_2 1024 = 10$ bit gerektirmektedir. Vericide en uygun kod kelimesini elde etmek için sentezleyerek-analiz metodu kullanılmaktadır. En düşük ağırlıklandırılmış hata enerjisini veren kod kelimesi seçilerek, bunun indeksi ve ilgili ölçekleme faktörü kodlanır. Sentezleyicide ise bu iki veri, dürtü dizisini elde etmek için kullanılır. Şekil 4.16’da CELP sentezleyici blok diyagramı verilmiştir.



Şekil 4.16 CELP Sentezleyici Blok Diyagramı

Bu yöntemin en büyük dezavantajı kod dağarcığındaki arama sürecinin hesap yükünü oldukça fazla olmasıdır. CELP algoritmasının hesap yükünün büyük bölümü bu arama işleminden gelmektedir. 1024 kod kelimesinden oluşan 40 pozisyonlu bir kod dağarcığı için, saniyede 500 milyon tane çarpma-toplama işlemi gerekmektedir. Kod dağarcığının en uygun şekilde aranması için çeşitli yöntemler geliştirilmiştir. Örneğin bir ön işleme uygulayarak hesap adedi indirgenebilmektedir. Ön işleme safhasında kod dağarcığının bir alt kümesi seçilerek, bu küme yoğun şekilde araştırılmaktadır.

Günümüzde düşük bit hızlı kodlama yöntemlerinin en popüler olan kod ile uyarılan doğrusal öngörülü kodlayıcı haberleşmede bir çok uygulama alanı bulmuştur. Hatta 1992 yılında düşük gecikmeli 16 kbit/san hızındaki CELP kodlayıcısı ITU-T tarafından telefon şebekesinde kullanılmak üzere G728 standardı ile tanımlanmıştır. Bunun yanı sıra Amerikan Savunma Dairesi (*Department of Defense-DoD*) 4.8 kbit/san hızındaki CELP kodlayıcısı Federal Standart 1016 adıyla askeri haberleşme standardı olarak kabul etmiştir [15]. CELP kodlayıcı 6. Bölümde ayrıntılı olarak incelenecektir.

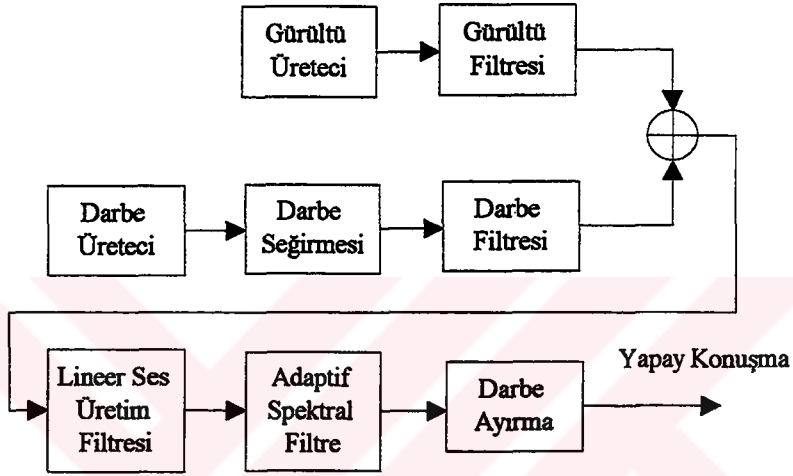
4.2.5.5 Karışık Uyarmalı Doğrusal Öngörülü Kodlama (MELP)

Oldukça yeni bir yaklaşım olan MELP (*Mixed –Excitation Linear Predictive*) kodlayıcı, Amerikan Savunma Dairesi (DoD) tarafından uzun çalışmalar sonucunda 1997 yılında 7 aday kodlayıcı arasından seçilen 2400 bit/san’lık federal standarttır [17]. Bu seçim yapılırken şu üç kriter göz önüne alınmıştır: anlaşılabilirlik, ses kalitesi ve konuşmacının tanınabilmesi. Ayrıca seçim kriteri olarak işlem yükü, hafıza gereksinimi ve gecikme gibi donanım parametreleri de değerlendirilmiştir. MELP kodlayıcı askeri ve ticari haberleşme sistemlerinde sık karşılaşılan gürültülü ortamlarda sağlam yapısı ile çok başarılı olmuştur. Ayrıca donanım olarak gereksiniminin diğer karşılaştırılabilir sistemlere göre oldukça az olması nedeniyle, özellikle gücün sınırlı olduğu taşınabilir sistemlerde kolaylıkla kullanılması mümkün olacaktır.

Geleneksel perde uyarmalı LPC kodlayıcılar, önceki bölümlerde de anlatıldığı gibi tam kutuplu sentez filtresini uyarmak için ya periyodik bir darbe katarı, ya da beyaz gürültü kullanır. MELP kodlayıcıda ise karışık bir uyarma modeli kullanarak daha doğal bir çıktı üretebilmektedir. MELP kodlayıcı geleneksel parametrik LPC yapısını kullanmakta; fakat karışık uyarma, periyodik olmayan darbeler kullanma, adaptif spektral filtreleme ve darbe ayırma gibi ek imkanlar sağlamaktadır. Karışık uyarma çok-bantlı bir karıştırma modeli ile gerçekleştirilmektedir. Bu model frekansa bağlı ses yapısını yeni adaptif bir sabit filtre bankası ile modellemektedir. Çok-bandlı bu karışık uyarmanın en önce farkedilen etkisi, özellikle genişbandlı akustik gürültü söz konusu olduğunda, LPC algoritmalarının temel problemi olan vızıltıyı (*buzz*) engellemesidir.

Konuşmanın sesli bileşenler içermesi durumunda MELP kodlayıcı periyodik veya periyodik olmayan uyarma darbe katarı kullanabilmektedir. Periyodik olmayan yapılar özellikle konuşmanın sesli sessiz (veya tersi) geçiş bölgelerinde gerekli olmaktadır. Böylelikle geçiş bölgelerinde oluşan tonlama gürültüsü ortadan kaldırılmış olmaktadır. Kodlayıcıdaki darbe ayırma işlevi, spektral olarak yassılatılmış

üçgen darbe temelli bir sabit darbe ayırma filtresi kullanılarak gerçekleştirilmektedir. Bu filtre perde periyodu içinde uyarma enejisini yaymak için kullanılmakta ve böylelikle yapay konuşmanın sertliği ortadan kaldırılabilir. Şekil 4.17'deki adaptif spektral filtre LPC ses üretme filtresinin kutuplarını temel almış ve yapay konuşmanın rezonans yapısını güçlendirmek için kullanılmaktadır. Bu filtre yapay ve doğal konuşma işareti arasındaki uyumu güçlendirmekte ve çıkışta daha doğal bir işaret elde edilmesini sağlamaktadır.



Şekil 4.17 MELP Kodlayıcı Blok Şeması

4.2.5.6 Sinüzoidal Modelleme (STC)

Sinüzoidal dönüşüm kodlayıcısı (*Sinusoidal Transform Coder-STC*) yukarıda anlatılmış olan diğer dönüşüm kodlayıcılarını andırmaktadır. Bu yapıdaki kodlayıcılar konuşma işaretini zamanla değişen genlik, faz ve frekanslı L adet sinüzoidal işaretin bileşimi olarak ifade eder:

$$\hat{s}_n = \sum_{k=1}^L A_k \cos(\Omega_k n + \phi_k) \quad (4.14)$$

Bunun yanısıra konuşma işaretini ifade etmek için kullanılan sinüzoidal işaretlerin sayısı da değişebilmektedir [18]. Sesli harfli konuşma yüksek oranda periyodiktir ve

daha az sayıda sinüzoidle ifade edilebilir. Sessiz harfli konuşma ise az sayıda uygun rastgele fazlı sinüzoid ile karakterize edilebilir. Böylelikle konuşma işareti az sayıda paarmetre ile karakterize edilebilir ve iyi bir sıkıştırma elde edilir. Daha düşük bit hızlarına inmek için sinüzoidlerin frekansları perde frekansının tam katı olarak seçilir.

Hızlı *Fourier* dönüşümünün tepe noktalarına denk düşen sinüzoidal işaretler seçilerek kodlamada kullanılır. Ortalama perde periyodunun 2.5 katı uzunluğunda bir *Hamming* penceresi ile konuşmadan örnek alınır. Frekans bölgesinde uygun bir çözünürlük elde etmek için bu gereklidir. Görüldüğü gibi kullanılan algoritma perde frekansına pek bağlı değildir. Perde frekansı yalnızca pencere uzunluğunu belirlemek için kullanılmaktadır. Bu nedenle sinüzoidal dönüşüm kodlaması konuşma olmayan işaretlerin kodlanması için de kullanılabilir.

4.2.5.7 Çoklu-Band Uyarma (MBE)

Çoklu-Band uyarma (*Multi-Band Excitation-MBE*) kodlamasının amacı düşük bit hızlarında ses kodlayıcıların ürettiğinden daha kaliteli konuşma elde ederken, hesap yoğunluğunu da CELP'in aşağısına indirmektir. MBE, geleneksel ses kodlayıcılarda kullanılan iki katlı konuşma üretim modelini (yani konuşma, ağızdaki ses oluşumunu sağlayan rezonans boşluğu modelini bir uyarı üretici modeli ile uyararak üretilmektedir) kullanmaktadır. Bu model ses kodlayıcılardakinin aynısı olmasına rağmen, uyarma modeli oldukça değişiktir [23].

Geleneksel ses kodlayıcılarda uyarma modeli, sesli/sessiz kararı ve eğer karar sesli ise temel frekanstan oluşmaktadır. Sessiz harfli konuşma için uyarma bir rastgele gürültü üretici, sesli harfli konuşma için ise periyodik bir darbe katarıdır. MBE kodlayıcı tek sesli/sessiz kararını bir dizi sesli/sessiz kararı ile değiştirmektedir. Uyarma bir çok alt banda bölünmekte ve konuşma spektrumu bu bandlardan her birinde sesli/sessiz kararı için incelenmektedir. Bu sürecin sonucu, bir dizi sesli/sessiz kararıdır. Daha sonra ses kodlayıcılarda olduğu gibi kodlama yapılır.

Böylelikle kodlanmış konuşma işaretinin gerçek konuşmada olduğu gibi anlık olarak sesli/sessiz karakterini ifade etmesi sağlanmaktadır.

4.2.6 Hece (Fonetik) Kodlayıcılar

Hece kodlayıcılar, kullanılan dilin içerdiği kuralların kodlama yordamı olarak kullanıldığı kodlayıcı tipidir [5]. Her dilde kelimeler hecelerin (*phonemes*) biraraya gelmesi ile oluşmaktadır. Eğer herhangi bir dilde kullanılabilecek tüm hece bileşimleri bulunur ve belli bir sıra ile depolanırsa istenilen sözcüğün kodlanması kolayca yapılabilir. Bu yöntem günümüzde genellikle metinden (*text*) konuşma elde edilmesi amacıyla kullanılmaktadır. Elde edilmek istenilen konuşma metni verilir, kelimeler hecelerine ayrılır ve ses parametreleri girilmiş heceler yardımıyla konuşma elde edilir.

Tahmin edilebileceği gibi bu yöntemde analiz işlemi (hecelerin bulunması, parametrelerin hesabı) uzun sürmektedir. Sonuçta elde edilen ses kalitesinde de düşmeler olması kaçınılmazdır.

BÖLÜM 5

DÜŞÜK BİT HIZLI KONUŞMA KODLAMA ARAÇLARI

5.1 KONUŞMA İŞARETİ

Konuşma işaretlerinin transmisyon hızını düşürmek için, kodlanacak konuşma işaretini detaylı olarak incelemek ve iletirken de bu işaretin sadece elektriksel olarak dönüştürülmüş bir işaret değil “konuşma” olduğunu unutmamak gereklidir. Bu tanımdan hareketle, amaç “konuşma” işaretini mümkün olduğunca az sayıda parametreyle ifade etmek ve bunun yanısıra anlam (mesaj içeriği), doğallık (örneğin, cinsiyet, yaş ve aksan gibi konuşmacı özellikleri) gibi konuşma özelliklerini korumaktır. Tipik konuşma işareti incelendiğinde, konuşmanın bazı çok tekrarlanan bileşenlere sahip olduğu ve konuşmayı daha düşük hızlarda iletmek için gerekli kapasitenin azaltılmasına izin veren olgunun bu fazlalıklar, ya da konuşma özellikleri olduğu ortaya çıkmaktadır.

Konuşmanın en önemli özellikleri rezonans ve perde yapısıdır. Bu özellikler; konuşmayı (özellikle ünlü konuşma) orijinalinden ayırd edilemeyecek şekilde hassas ifade etmek için konuşma kodlama algoritmalarında kullanılabilir ve bu da düşük bit hızlarında yüksek kaliteli konuşma kodlama sistemleri için bir potansiyel oluşturur. Gerçek dalga biçimini iletmek yerine sadece perde ve konuşma spektrumu zarfı bilgisi gibi zamana göre daha yavaş değişen parametreler iletilir ve daha sonra da konuşma alıcı uçta yeniden sentezlenirse önemli bir bant genişliği tasarruf edilir. Konuşmanın yüksek kaliteli bir şekilde yeniden elde edilebilmesi için bu parametrelerin öngörüsünde etkin yöntemler geliştirilmesi gerektiği açıktır.

Vektör nicemleme; kanal kapasitesinin veya sayısal bellek gereksiniminin verimli şekilde kullanılmasını sağlayan ve gerekli konuşma sadakatini (*fidelity*) koruyacak

şekilde bit hızını düşüren diğer bir diğer konuşma sıkıştırma yöntemidir. Bu nedenle, bu bölümde zamana bağlı *Fourier* dönüşümü, konuşma spektrumu zarfı, perde ve vektör nicemleme konularındaki araçlar ve bu kavramların düşük veri hızlarında konuşma kodlamasına uygulanmasını ele alınacaktır.

5.2 KISA SÜRELİ SPEKTRUM ANALİZİ

Kısa süreli (*short-time*) *Fourier* dönüşümü, frekans esaslı konuşma kodlama algoritmalarında önemli bir rol oynar. *Fourier* dönüşümünün önemi, konuşma işaretini tek (*unique*) bir karmaşık üstel ifade ile göstermesidir [25]. Konuşmanın zamana bağlı bir *Fourier* dönüşümü olarak gösterimi ise, konuşma dalgaşeklinin zamana göre değişen özelliklerini yansıtır. Genel olarak, zamana bağlı *Fourier* dönüşümü aşağıdaki formülle gösterilebilir :

$$S_k(\omega) = \sum_{n=-\infty}^{\infty} \omega(k-n)s(n)e^{-jn\omega} \quad (5.1)$$

Burada, $\omega(k-n)$, istenen herhangi bir konuşma parçasını seçmek için, herhangi bir belirli zaman indeksine (k) göre değiştirilebilen bir gerçek pencere fonksiyonudur. Zamana bağlı *Fourier* dönüşümü temel olarak iki değişkenden oluşan bir fonksiyondur: ayrık k zaman indeksi ve sürekli ω frekans değişkeni. Zamana bağlı *Fourier* dönüşümü başka şekilde aşağıdaki gibi ifade edilebilir :

$$\begin{aligned} S_k(\omega) &= \sum_{n=-\infty}^{\infty} \omega(n)s(k-n)e^{-j(k-n)\omega} \\ &= e^{jk\omega} \sum_{n=-\infty}^{\infty} \omega(n)s(k-n)e^{jn\omega} \end{aligned} \quad (5.2)$$

Bu eşitlikler iki şekilde yorumlanabilir: Birincisi, k sabit kabul edilirse, $S_k(\omega)$, $-\infty < n < \infty$ aralığındaki $\omega(k-n)s(n)$ dizisinin normal *Fourier* dönüşümüdür. Bu nedenle, sabit k için $S_k(\omega)$ normal *Fourier* dönüşümüyle aynı özellikleri taşır [26]. İkincisi

ise, $S_k(\omega)$, k zaman indeksinin bir fonksiyonu ve ω sabit ise, her iki eşitliğin (5.1 ve 3.2) konvolüsyon şeklinde olacağı açıkça görülebilir.

5.2.1 Pencereelerin Katkısı

$\omega(n)$ penceresi, istenen bölgenin dışında kalan işaretleri sıfırlamak suretiyle, konuşma işaretinin işlenecek kısmını belirler. Pencereelerin bir çok tipi (Barlett, Hanning, Hamming, Blackman, vs.) vardır, ancak, bu pencereelerin zamana bağlı *Fourier* dönüşümü gösterimindeki etkileri sadece dikdörtgen ve *Hamming* pencereelerinin özellikleriyle açıklanabilir:

Dikdörtgen pencere :

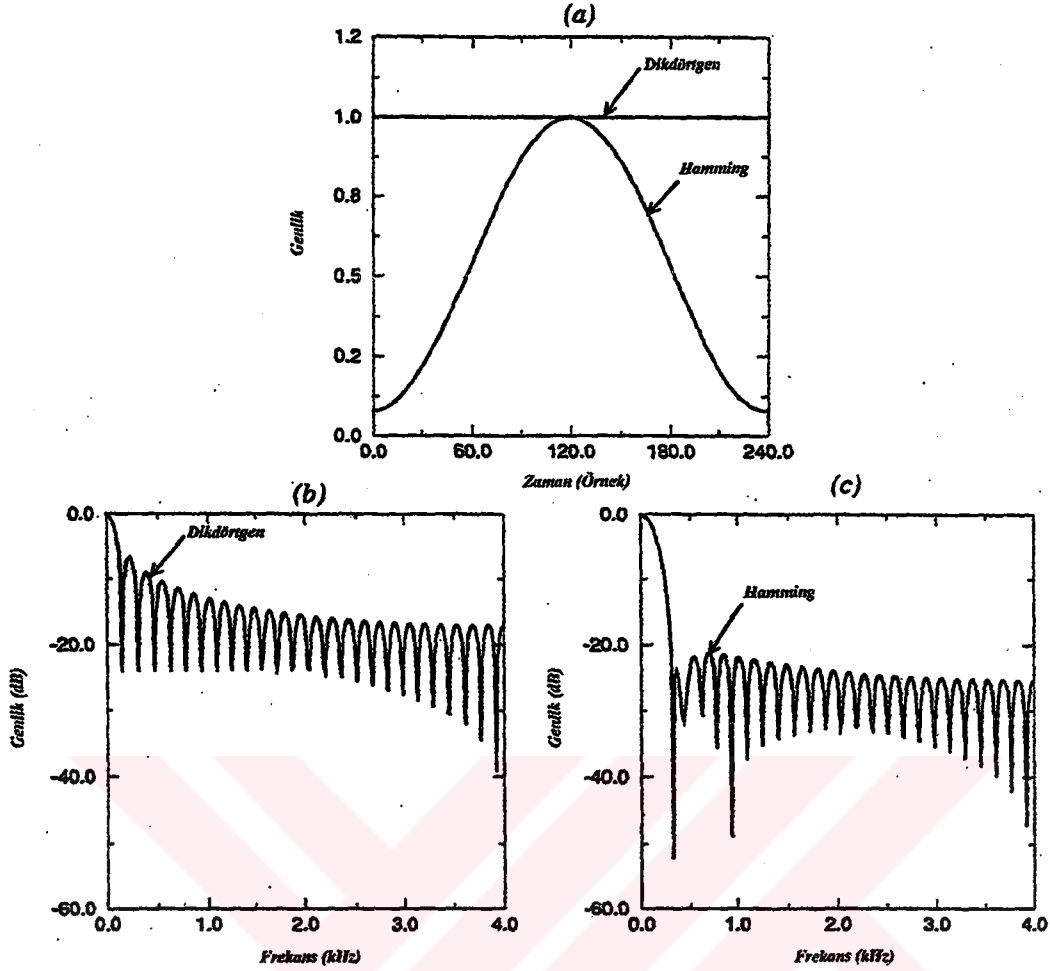
$$\omega(n) = \begin{cases} 1 & ; 0 \leq n \leq N-1 \\ 0 & ; \text{aksi halde} \end{cases} \quad (5.3)$$

Hamming penceresi :

$$\omega(n) = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) & ; 0 \leq n \leq N-1 \\ 0 & ; \text{aksi halde} \end{cases} \quad (5.4)$$

Dikdörtgen pencere, $(n-N+1)$ 'den n 'ye kadar olan aralıktaki bütün örneklerle eşit ağırlık uygulanmasına karşılık düşer. 30 msan'lik dikdörtgen ve Hamming pencereelerinin frekans cevapları sırasıyla Şekil 3.1b ve 3.1c'de gösterilmiştir.

Hamming penceresinin, geçirme bandının dışında dikdörtgen pencereye göre daha büyük bir işaret zayıflaması yarattığı açıktır. Bu iki pencerenin her ikisinde de işaret zayıflaması gerçekte pencere süresinden bağımsızdır. Bu suretle, pencere boyu N arttıkça, bant genişliği azalır. N çok küçük olursa (yani, perde periodu civarı veya daha kısa) kısa süreli enerji, dalgaşeklinin ayrıntılarına bağlı olarak düzensiz şekilde dalgalanır.



Şekil 5.1 Pencere özellikleri : (a) impuls cevabı, (b) dikdörtgen pencere cevabı ve (c) Hamming penceresi cevabı :

N çok büyük olursa (yani, birkaç perde periyodu mertebesinde) kısa süreli enerji çok yavaş değişir ve bu yüzden konuşma işaretinin değişen özelliklerini yeterli şekilde yansıtamaz. Bu durum, N 'nin hiçbir değerinin tam olarak tatmin edici sonuç vermemesi anlamına gelir, çünkü perde periyodunun süresi, yüksek perdede bir kadın veya çocuk için yaklaşık 16 örnekten (8 kHz örnekleme frekansında), düşük perdedeki bir erkek için 150 örneğe kadar değişir. Bu sorun dikkate alınarak, N için, 8 kHz örnekleme hızında (yani, 15-25 msan süre) 120-200 civarında uygun ve pratik bir değer seçilebilir.

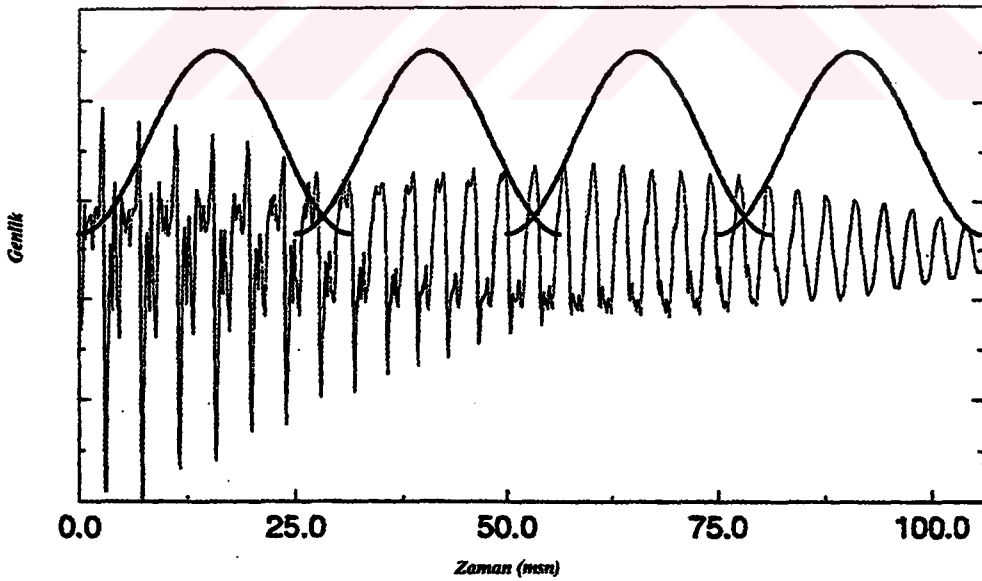
İdeal bir pencere fonksiyonu $w(n)$, spektrumu $W(\omega)$ olan aşağıdaki özelliklere sahip bir fonksiyon olmalıdır [21]:

- i) Frekans çözünürlüğü yüksek olmalı (yani ana lob dar ve keskin)
- ii) Ana lob dışındaki bir konvolüsyon integralinin sebep olduğu yapma (*spurious*) bozulma küçük olmalı ($W(\omega)$ 'nın yapma cevabı küçük).

Bu iki gereksinim birbirleriyle çelişkilidir ve aynı anda karşılanamaz. Bu sebeple, $w(n)$, bu iki gereksinime mümkün olduğunca uyumlu olacak şekilde seçilmelidir.

5.2.2 Fourier Dönüşümü Yaklaşımı

Zamana bağlı Fourier dönüşümü, k zaman indeksinin bir fonksiyonudur. k zaman indeksi, $w(k-n)$ penceresi $s(n)$ konuşma dizisi boyunca kaydırılabilecek şekilde bütün tam sayı değerlerini alır. Şekil 5.2'de, $s(n)$ ve $w(k-n)$ 'yi n 'nin fonksiyonu olarak çeşitli k değerleri için gösteren grafik verilmiştir.



Şekil 5.2 Konuşma işaretinin, pencereleme yöntemiyle heme n hemen durağan çerçevelere bölünmesi

Önceki bölümde açıklandığı gibi, sabit bir k zaman indeksi için, $S_k(\omega)$ zamana bağlı *Fourier* dönüşümü normal bir *Fourier* dönüşümü ile aynı özelliklere sahiptir. Bu da, $s(n)$ giriş dizisinin zamana göre değişen *Fourier* dönüşümünden tam olarak elde edilebilmesini sağlar. Bu durumda, $S_k(\omega)$ dönüşümü, $\omega(k-n)s(n)$ 'nin *Fourier* dönüşümüdür ve

$$\omega(k-n)s(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_k(\omega) e^{jn\omega} d\omega \quad (5.5)$$

eşitliği de ters *Fourier* dönüşümüdür. $\omega(0) \neq 0$ olursa, 3.5 nolu eşitlik $n=k$ için yeniden düzenlenirse aşağıdaki eşitlik elde edilir:

$$s(n) = \frac{1}{\omega(0)2\pi} \int_{-\pi}^{\pi} S_k(\omega) e^{jn\omega} d\omega \quad (5.6)$$

$\omega(0) \neq 0$ durumunda, bir tam periyotta bütün ω değerleri için $S_k(\omega)$ biliniyorsa konuşma işareti $S_k(\omega)$ 'dan tam olarak elde edilebilir [27].

Zamana bağlı *Fourier* dönüşümü $S_k(\omega)$, bir konuşma işaretinin kısa süreli özilişki fonksiyonuyla ilgili olan önemli bir özelliğe sahiptir. $S_k(\omega)$ zamana bağlı *Fourier* dönüşümü, k 'nın her değeri için $\omega(k-n)s(n)$ 'nin normal *Fourier* dönüşümüyse, güç spektrumunun da,

$$P_k(\omega) = |S_k(\omega)|^2 = S_k(\omega) S_k^*(\omega) \quad (5.7)$$

şeklinde yazılabileceği kolayca görülebilir. Bu da (5.8)'deki kısa süreli özilişki fonksiyonunun *Fourier* dönüşümüdür.

$$R_k(m) = \sum_{n=-\infty}^{\infty} \omega(k-n)s(n)\omega(k-m-n)s(n+m) \quad (5.8)$$

$S_k(\omega)$ kısa süreli *Fourier* dönüşümü çeşitli başka biçimlerde ifade edilebilir. Örneğin, konuşma spektrumunun gerçek ve sanal kısımlarıyla aşağıdaki gibi gösterilebilir:

$$S_k(\omega) = \text{Re}(\omega) + j \text{Im}(\omega) \quad (5.9)$$

Diğer bir $S_k(\omega)$ gösterimi de aşağıdaki gibi genlik ve faz cinsinden gösterimdir:

$$S_k(\omega) = |S_k(\omega)| e^{j\theta_k(\omega)} \quad (5.10)$$

$|S_k(\omega)|$ ve $\theta_k(\omega)$ büyüklükleri ile $\text{Re}(\omega)$ ve $\text{Im}(\omega)$ arasında bağıntı kurulabilir.

Spektrum analizinden önce konuşma işaretine uygulanan pencere fonksiyonunun şeklinin, zamana bağlı *Fourier* dönüşümünün niteliği üzerinde önemli bir etkisi vardır. $S_k(\omega)$, $\omega(k-n)s(n)$ dizisinin normal *Fourier* dönüşümü olarak kabul edilirse ve bir konuşma dizisinin normal *Fourier* dönüşümünün ve uygulanan pencere fonksiyonunun sırasıyla (5.11) ve (5.12)'deki gibi ifade edilebildiği kabul edilirse,

$$S(\omega) = \sum_{n=-\infty}^{\infty} s(n) e^{-jn\omega} \quad (5.11)$$

$$W(\omega) = \sum_{n=-\infty}^{\infty} w(n) e^{-jn\omega} \quad (5.12)$$

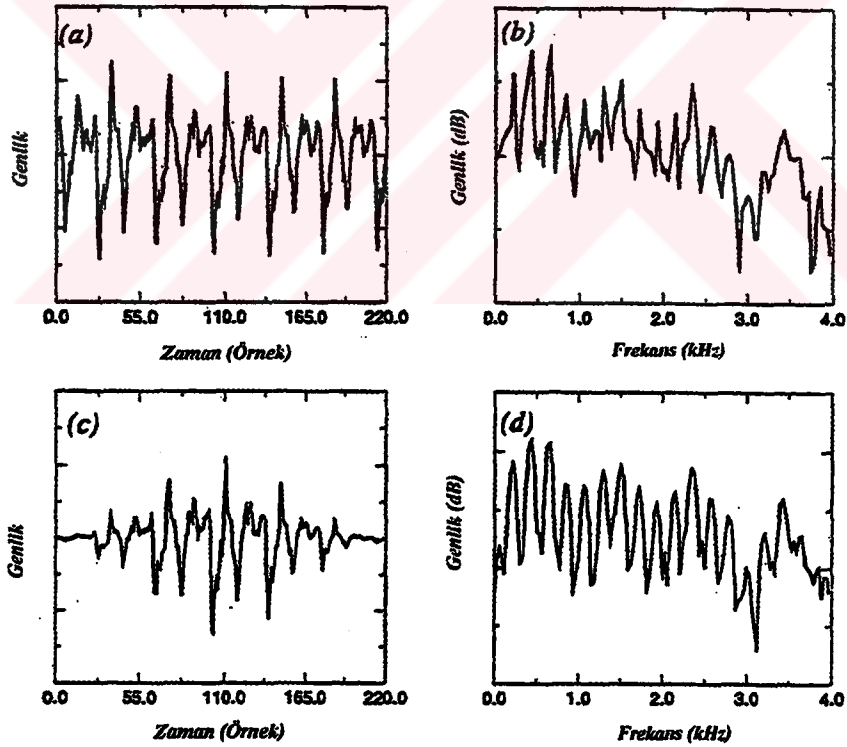
sabit k değeri için $\omega(k-n)s(n)$ çarpımının normal *Fourier* dönüşümü, $\omega(k-n)$ ve $s(n)$ 'in ayrı ayrı *Fourier* dönüşümlerinin konvolüsyonuna eşittir. Sabit k değeri için $\omega(k-n)$ 'nin *Fourier* dönüşümü $W(\omega)e^{-jk\omega}$ olduğundan $S_k(\omega)$ aşağıdaki gibi yazılabilir [25]:

$$\begin{aligned} S_k(\omega) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} W(\Omega) e^{jk\Omega} S(\omega - \Omega) d\Omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} W(\Omega) e^{jk\Omega} S(\omega + \Omega) d\Omega \end{aligned} \quad (5.13)$$

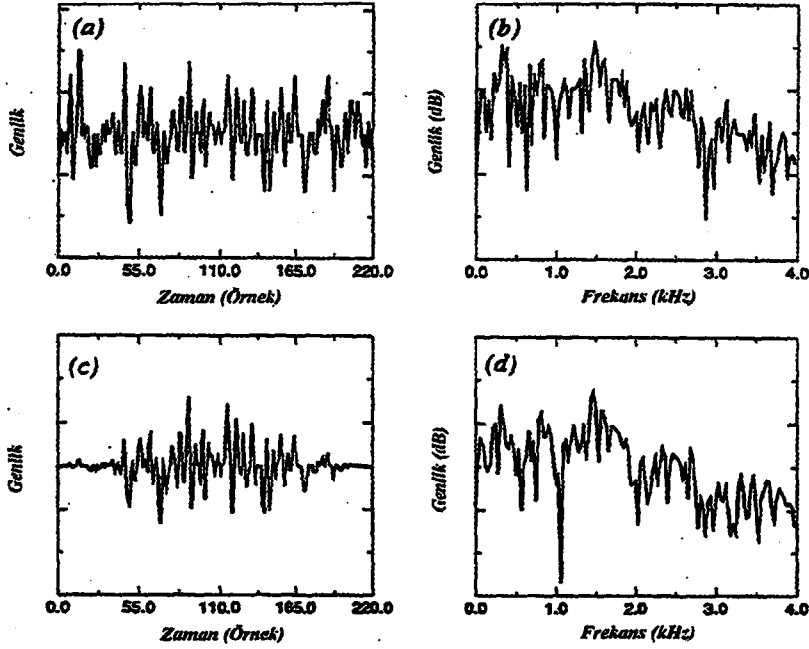
Bu eşitliklerden, $s(n)$ dizisinin *Fourier* dönüşümünün $-\infty < n < \infty$ aralığında, kaydırılmış pencere dizisinin *Fourier* dönüşümüyle konvolüsyonunun zamana bağlı *Fourier* dönüşümünü oluşturduğu açıktır. $w(k-n)$ pencere fonksiyonu, n 'in k civarındaki sınırlı bir aralığın dışındaki değerleri için sıfırdır. Bu durumda, pencerenin içindeki $s(n)$ konuşma işaretinin özelliklerinin pencerenin dışında da devam ettiği kolayca kabul edilebilir [27]. Örneğin, pencere içindeki konuşma işareti ünlü veya ünsüz olursa, aynı ünlü veya ünsüz özelliklerin pencere dışında da mevcut olduğu kabul edilebilir. Bu nedenle, konuşma işaretinin temel özelliklerinin pencerenin dışında ya devam ettiğini, ya da sıfır olduğunu kabul etmek yerinde olur. Böylece, zamana bağlı *Fourier* dönüşümü, işaretin pencere içindeki kısmının *Fourier* dönüşümünün düzgün bir modeli olarak yorumlanabilir. Sonuç olarak, $W(\Omega)$ penceresinin *Fourier* dönüşümünün özellikleri önem kazanır. (5.13) eşitliğinden, $S_k(\omega)$ 'nın $S(\omega)$ 'nın özelliklerini doğru şekilde içermesi için $W(\Omega)$ fonksiyonunun idealde $S(\omega)$ 'ya göre bir dürtü olarak görünmesi gerektiği açıktır. Konuşma spektrumu analizi için çeşitli pencere uzunlukları kullanmanın etkileri Şekil 5.3, 5.4 ve 5.5'de gösterilmiştir.

Bu şekillerin her birinde (a) dikdörtgen bir pencere fonksiyonuyla çarpılan konuşma işaretini, (b) dikdörtgen pencereyle elde edilen konuşma spektrumu genliğini dB olarak, (c) (a)'daki konuşma işaretinin aynısının bir Hamming pencere fonksiyonuyla çarpılmış şeklini, (d) Hamming penceresiyle elde edilen konuşma spektrumu genliğini dB cinsinden göstermektedir. Şekil 5.3'de, ünlü bir konuşma işareti 220 örnekli pencere süresiyle (8 kHz örnekleme hızında 27,5 msan) gösterilmiştir. Konuşma işaretinin periyodu Şekil 5.3c ve 5.3d'de açık olarak verilmektedir. Şekil 5.3d'de kısa süreli *Fourier* dönüşümünün ardışık iki harmoniği arasında, temel frekans ve onun harmoniklerinde birbirlerine göre sabit frekans mesafeli dar tepeler veya loblar vardır. Şekil 5.3d'de gösterilen spektrumda, 500 Hz civarında güçlü bir birinci rezonans tepesi ve 1700 Hz ve 2500 Hz civarında da sırasıyla ikinci ve üçüncü rezonans tepeleri vardır. 3500 Hz civarındaki başka bir rezonans tepesi de açık olarak görülmektedir. Bu spektrumda ses oluşum boşluğunun darbe spektrumunun alçak geçiren özelliğinden dolayı, daha yüksek frekanslarda ani düşme eğilimi de görülmektedir.

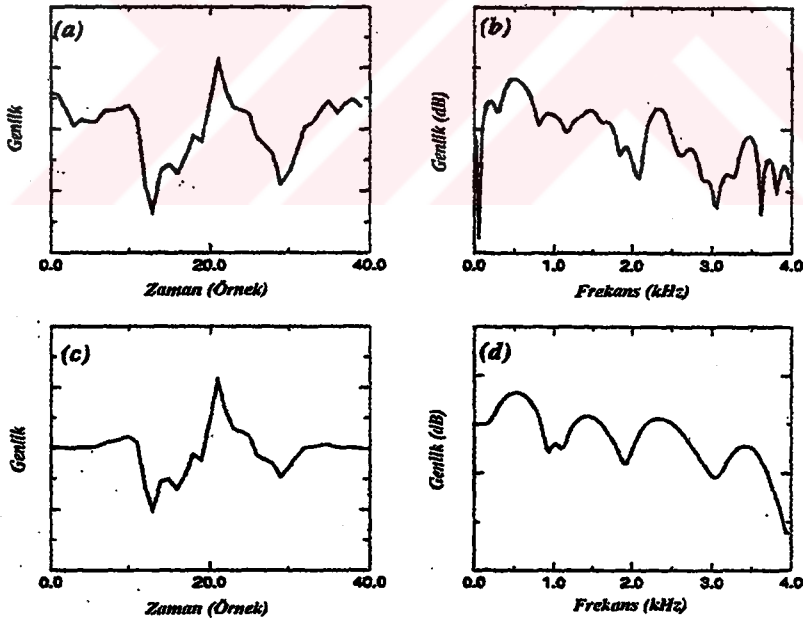
Şekil 5.3b (dikdörtgen pencere) ve 5.3d (*Hamming* penceresi) arasında yapılan bir karşılaştırma, perde harmonikleri, rezonans yapısı ve genel spektrum şekli açısından büyük ölçüde bir genel benzerlik ortaya koymaktadır. Bu iki şekil arasındaki en belirgin fark, Şekil 5.3b'deki perde harmoniklerinin daha keskin olmasıdır. Bunun nedeni, dikdörtgen pencerenin *Hamming* penceresine göre frekans çözünürlüğünün daha yüksek olmasıdır. Bu iki spektrum arasındaki başka bir fark da, bir dikdörtgen pencerenin nispeten büyük yan loblara sahip olması nedeniyle bu yan lobların yarattığı bozulmanın, gürültülü bir spektruma neden olmasıdır. Bu da yan lobların bitişik harmonikler arasındaki boşlukta etkileşimi nedeniyle ve zaman zaman harmonikler arasında rastgele bir değişime sebep olur. Bitişik harmonikler arasındaki bu etki "kaçak" olarak adlandırılır, istenmeyen bir durumdur ve dikdörtgen pencerenin ana lobunun sağladığı avantajları yok etme eğilimindedir. Bu nedenle, bu gibi pencereler konuşma spektrumu analizinde nadiren kullanılır.



Şekil 5.3 Ünlü bir konuşma parçasının 27,5 msan'lik bir pencere kullanılarak yapılan spektral analizi (a ve c, zaman dalga şekilleri, b ve d bunlara karşılık gelen spektrumlar)



Şekil 5.4 Ünsüz bir konuşma parçasının 27,5 msan'lik bir pencere kullanılarak yapılan spektrum analizi (a ve c, zaman dalga şekilleri, b ve d bunlara karşılık gelen spektrumlar)



Şekil 5.5 Ünlü bir konuşma parçasının 5 msan'lik bir pencere kullanılarak yapılan spektrum analizi (a ve c, zaman dalga şekilleri, b ve d bunlara karşılık gelen spektrumlar)

Şekil 5.5'de, 40 örnekli (5 msan) bir ünlü konuşma parçası için benzer bir karşılaştırma seti verilmiştir. Bu gibi kısa analiz pencereleri için konuşmanın periyodu, ne Şekil 5.5a (dikdörtgen) ve 5.5c (Hamming)'de gösterilen zaman esaslı konuşma işaretlerinde, ne de Şekil 5.5b (dikdörtgen) ve 5.5d (Hamming)'de gösterilen konuşma spektrumunda görülebilmektedir. Şekil 5.5'de gösterilen spektrumda 500, 1500, 2400 ve 3400 Hz'de birkaç geniş rezonans tepesi bulunmaktadır. Şekil 5.5b ve 5.5d arasında yapılan karşılaştırmada, yine, dikdörtgen pencerede daha yüksek bir frekans çözünürlüğü elde edilmektedir.

Pencerelerin, 220 örnek uzunluğundaki bir ünsüz konuşma parçası üzerindeki etkileri Şekil 5.4'de gösterilmiştir. Bu şekilde, ünsüz konuşma spektrumunda bir dizi keskin rezonans tepesi ve sıfırının yavaş yavaş üst üste gelme eğilimi açıkça görülmektedir. Ünsüz konuşmanın rastgele niteliğine bağlı olarak konuşma spektrumunda gürültülü bir görünüm bulunmaktadır. Bu şekilden, Hamming penceresinin bir dikdörtgen pencereden daha düzgün bir spektrum ortaya koyduğu görülmektedir.

Pencerenin süresi ile kısa süreli Fourier dönüşümünün özellikleri arasındaki temel ilişki Şekil 5.3, 5.4 ve 5.5'de açıkça gösterilmiştir. Sonuç olarak, bir konuşma işaretinin frekans çözünürlüğü pencerenin uzunluğu ile ters orantılı olarak değişir. Spektrum analizi sırasında pencere kullanmanın amacı, analiz edilecek zaman aralığını, dalga şeklinin özelliklerinde önemli bir değişiklik olmayacak şekilde sınırlamaktır. Örneğin, Şekil 5.3c'de, rezonans frekanslarının 27,5 msan aralığında belirgin şekilde değiştiği açıktır. Çerçeve boyunca bu geçici değişimi göstermek için daha kısa bir analiz aralığı gereklidir. 27,5 msan aralığının başına ve sonuna 5 msan süreli pencereler yerleştirildiğinde bu iki kısa süreli Fourier dönüşümü arasında belirgin bir fark olur. Bu sebeple, iyi bir geçici çözünürlük kısa bir pencere ile sağlanabilirken, sürekli iyi bir frekans çözünürlüğü için uzun bir pencere seçilmelidir.

5.3 KONUŞMA İŞARETİ SPEKTRUMU ZARF ÖNGÖRÜSÜ

Doğrusal öngörü, düşük bit hızlı konuşma kodlaması için günümüzün en popüler tekniğidir ve konuşma analizinde çok önemli bir araç haline gelmiştir. Doğrusal

Öngörülü Kodlama'nın (LPC) popülerliği, konuşma spektral genliğinin kısa ve hassas gösteriminden, ayrıca görelî hesaplama basitliğinden kaynaklanmaktadır. LPC, spektral kutup ve sıfırların (yani rezonansların) frekanslarının ve bant genişliklerinin öngörülmesinde kullanılmakta, fakat esas olarak ağızdaki ses oluşumunu sağlayan rezonans boşluğunun yapısı hakkındaki bilgiyi içeren parametre setini oluşturmaktadır. Analizin karmaşıklığını en az düzeye indirmek için konuşma işaretinin genelde tam kutuplu bir kaynaktan geldiği kabul edilir, ki bu kabul de çoğu uygulamada büyük güçlülere sebep olmaz.

Spektrum zarfı öngörüsü için çok çeşitli LPC analiz yöntemleri vardır. *Itakura* ve *Saito* [28], LPC teknikleri ile spektrum zarfı öngörüsünün temel teorik esasının maksimum benzerlik (*ML-maximum likelihood*) öngörü tekniğinde bulunduğunu göstermiştir. Hattâ, *Itakura-Saito* kıstası temelli bir spektrum eşleme yorumu geliştirmek için de bu tekniği kullanmışlardır. Kullandıkları temel matematik model, özbağımlı (*autoregressive*) rastgele bir yordamın örnek fonksiyonları olan konuşma dalga şekilleriyle ilgilidir. Bu model, ünsüz konuşma sesleri için uygun olmakla birlikte, aynı zamanda ünlü konuşma sesleri için de bu durumda konuşma dalga şekilleri periyodik olduğu için geçerlidir.

5.3.1 Maksimum Benzerlik Yöntemi ile Spektrum Zarfı Öngörüsü

Sabit bir konuşma işareti $s(n)$, T periyodu ile periyodik ise, özilişki $R(m) = \varepsilon[s(n)s(n+m)]$ fonksiyonu da T periyodu ile periyodiktir. Bu durumda, $R(m)$ özilişkisi, *Fourier* serisiyle aşağıdaki gibi gösterilebilir :

$$R(m) = \sum_{k=-T/2}^{T/2} P_k e^{jm\omega_k} \quad (5.14)$$

Burada, $\omega_k = 2\pi k / T$ dır ve:

$$P_k = \frac{1}{T} \sum_{m=-T/2}^{T/2} R(m) e^{-jm\omega_k} \quad (5.15)$$

ifadesi periyodik konuşma işaretinin ayrık zamanlı güç spektrumunu gösterir. Spektrum öngörü problemi, $s(n)$ orijinal konuşma dalga şeklinden $\{P_k\}_{k=-T/2}^{T/2}$ 'yi belirlemektir. Spektrumda modelleme kısıtlamaları olmadığında çözüm açıktır. Ancak, bazı fonksiyonel modellerle ayrık spektrumu hassas olarak göstermek çok karmaşıktır. Bu problemin bir çözümü, doğrusal öngörü tekniklerini kullanmaktır. Daha genel olarak, spektrum, $P_n(a_k)$ model spektrumu hesaplama kuralı vasıtasıyla $\{a_k\}_{k=1}^p$ parametreleri kullanılarak modellenenir. Bu problem, orijinal konuşma dalga şeklindeki istatistiksel bilgi kullanılarak çözülebilir (bu durumda, orijinal $s(n)$ veya konuşma spektrumundaki eşdeğeri $\{S_k\}$, $\{a_k\}_{k=1}^p$ spektral parametrelerinin öngörülmesi için kullanılabilir). Model parametrelerini elde etmek için başka bir yaklaşım da, asimptotik optimallik özelliklerine sahip olduğu bilinen maksimum benzerlik yöntemini kullanmaktır. Maksimum benzerlik model parametrelerinin hesaplanmasında orijinal konuşma spektrumu için bir olasılık modeli gereklidir. Bu olasılık modelinin bir *Gauss* modeline eş olduğu gösterilmiştir [29]. Bu nedenle, $\{S_k\}$ konuşma spektrumu değişkenlerinin sıfır ortalamayla karmaşık bir *Gauss* dağılımına sahip olduğu kabul edilir. Konuşma spektrumu bileşenlerinin birbirinden bağımsız olduğu gösterildiğinden [30] *Gauss* kabulü bunların da bağımsız olduğunu gösterir. Konuşma spektrumu zarfını bir kaç parametreyle hassas şekilde modellemek için, konuşma spektrumunun tepe değerleri ve bunlara karşılık gelen frekanslar hesaplanır ve maksimum benzerlik spektrum eşleme kistasında kullanılır. Ortalama temel frekansta konuşma spektrumundan tepe değerlerin genliklerini elde etmek için bir tepe değer bulma algoritmasından yararlanılır [31]. Bu nedenle, maksimum benzerlik tekniğinde kullanılan $\{X_k\}_{k=0}^{N-1}$ güç ölçümleri (N , 4 kHz'lik bir konuşma spektrumundaki tepe değer veya harmonik sayısıdır) ve bunlara karşılık gelen ω_k frekansları, tepe değer alma tekniği kullanılarak elde edilenler. Bu tepe değer genlikleri genelde harmonik olmamasına karşın, maksimum benzerlik spektrum eşleme kistası, spektrumun tepe değerlerindeki güç ölçümlerine uygun bir sentetik spektrum zarfı belirler.

Maksimum benzerlik yönteminin olasılık yoğunluk fonksiyonu (PDF), bu nedenle, aşağıdaki gibi yazılabilir :

$$\begin{aligned}
 PDF(X_n, a_k) &= \prod_{n=0}^{N-1} \frac{1}{\pi P_n(a_k)} e^{-x^2} \\
 &= \pi^{-N} e^{-\sum_{n=0}^{N-1} (x^2 + \log[P_n(a_k)])}
 \end{aligned} \tag{5.16}$$

Burada, $x^2 = |X_n|^2 / P_n(a_k)$ 'dır. $\{a_k\}$ 'nın maksimum benzerlik öngörülürü, PDF tepe değeri maksimize edilerek veya PDF genişliğinin ana lobu minimize edilerek bulunur. Bu işlem, PDF 'in logaritmasının negatifinin minimize edilmesine eşdeğerdir ve bu da benzerlik işlevi olarak adlandırılır [28]:

$$\begin{aligned}
 MI &= -\log[PDF(X_n, a_k)] \\
 &= N \log(\pi) + \sum_{n=0}^{N-1} (x^2 + \log[P_n(a_k)])
 \end{aligned} \tag{5.17}$$

Frekans örnekleme yeterince sık olursa (N çok büyük) toplamının yerini bir integral alabilir. Ortaya çıkan eşitlik maksimum benzerlik spektrum zarfı öngörü kistasını oluşturur ve sürekli spektrum için *Itakura* ile *Saito*'nun yöntemine çok benzer [28]. Ancak N , işaretin bant genişliği içindeki tepe değer genliklerinin sayısı olduğundan, genelde bu yaklaşıma izin verecek kadar büyük değildir. (5.4) eşitliğinden, ilk olarak *Itakura* ve *Saito* tarafından önerilen fikir [28], maksimum benzerlik kıstası için bir spektrum eşleme yorumu geliştirmek için kullanılabilir. Öncelikle, benzerlik fonksiyonu üzerinde daha düşük bir sınır elde edilir. Bu durum için, çözülecek N adet bilinmeyen P_k parametresi bulunur. Maksimum benzerlik spektrum zarfı öngörülerinin $P_k = |X_k|^2$ olduğu kolayca gösterilebilir. $\{a_k\}$ değerleri için bir diferansiyel benzerlik işlevi aşağıdaki gibi tanımlanır :

$$MI = \sum_{n=0}^{N-1} \{E(\omega_n) - E(\omega_n) - 1\} \tag{5.18}$$

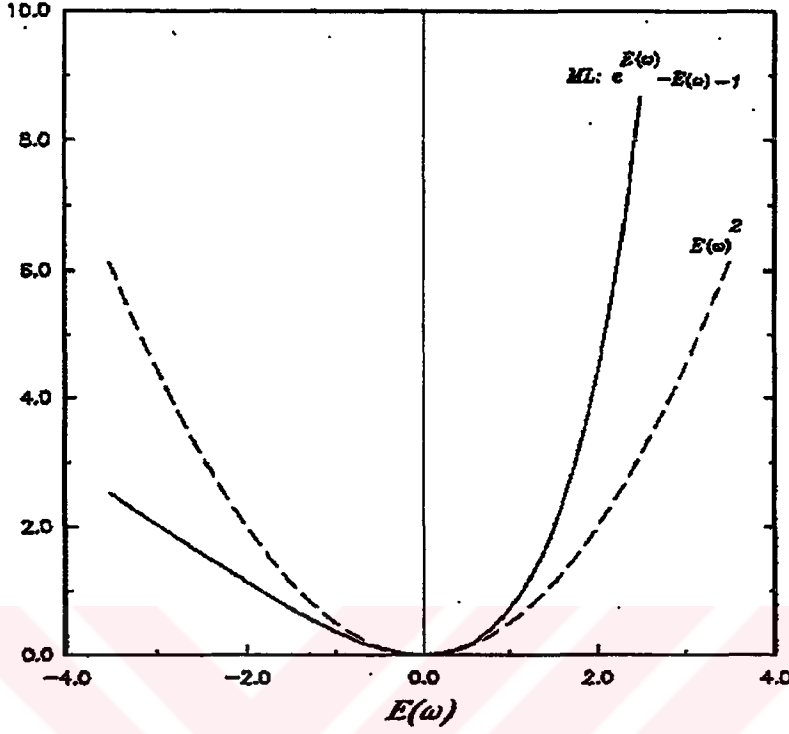
Burada,

$$E(\omega_n) = \log(x^2) = \log|X_n|^2 - \log P_n(a_k) \quad (5.19)$$

$|X_n|^2$ orijinal güç spektrumu için ω_n frekansındaki güç ile, model spektrumu tarafından $P_n(a_k)$ olarak öngörülen güç arasındaki hatayı dB cinsinden vermektedir. *Itakura* ve *Saito* hata kıstasını takiben [28], bu sonuç, aşağıdaki eşitlikle verilen karesel dB hatası kıstasında elde edilen sonuçla karşılaştırılırsa :

$$M^m = \sum_{n=0}^{N-1} [E(\omega_n)]^2 \quad (5.20)$$

(5.18) ve (5.20) hata kıstasları Şekil 5.6'da gösterilmiştir. (5.19)'daki $E(\omega_n)$, temel frekansın harmoniklerindeki hata gücü olarak tanımlanır. Şekil 5.6'da görüldüğü gibi, (5.20) eşitliği, ölçülen güç büyüklüklerinin üstünde ve altında dB modeli hatalarına eşit ağırlık verirken, (5.18) nolu eşitlikteki maksimum benzerlik kıstası, model spektrumu ölçülen güç büyüklüklerinin altında bulunduğu ortaya çıkan hatalara çok daha fazla ağırlık verir. Bu, maksimum benzerlik spektrum zarfı öngörü tekniğinin orijinal spektrum büyüklüklerinin üstünde bir sentetik model spektrumu vermesini sağlar. Yukarıdaki analiz, doğrusal olmayan spektrum eşleme yorumunun, kabul edilen spektrum modelinden bağımsız olarak özbağımlı (tam kutuplu) bir yordamla modellenebilecek hem periyodik, hem de periyodik olmayan konuşma işaretleri için geçerli olduğunu göstermektedir [28]. Hattâ, periyodik durumda, model spektrumu sadece ayrı harmonik frekanslarda ölçülen güce uygun hale getirilmelidir.



Şekil 5.6 Maksimum benzerlik spektrum eşleme kısıtı

5.3.1.2 Tam Kutuplu Modelleme ile Spektral Öngörü

Yukarıda, maksimum benzerlik kısıtlarının tarifi yapılırken, bu kısıtın model spektrumu parametrelerinin öngörüsünde fonksiyonel biçimde kullanılabileceği belirtilmiştir. Tam kutuplu model, konuşma spektrumu zarfını belirlemek için özbağlanımlı yordamlara denk olduğundan özel bir ilgi odağı olmuştur. Özellikle konuşma analizi için, ağızdaki ses oluşumunu sağlayan rezonans boşluğu filtresinin yaklaşık tam kutuplu lineer bir filtre olduğu gösterilmiştir. Kay, periyodik olmayan bir özbağlanımlı süreç için çözümü maksimum benzerlik problemine yaklaştıran özyinelemeli (*recursive*) bir teknik geliştirmiştir [29]. Kay'ın fikirlerinin pek çoğu, McAulay tarafından benzer bir çözüm üretmekte kullanılmıştır [30].

Tam kutuplu spektrum, aşağıdaki eşitliklere göre özbağlanımlı özilişki işleviyle bağlantılıdır:

$$R(n) = \sum_{\omega=0}^{N-1} \frac{\sigma^2}{|A_p(\omega)|^2} e^{jn\omega} \quad (5.21)$$

$$A_p(\omega) = 1 + \sum_{k=1}^p a_k^p e^{-jk\omega} \quad (5.22)$$

Burada, $\{a_k^p\}$, p 'inci model dizisi için özbağlanım parametreleridir. N süresi, tam kutuplu model spektrumunun $2\pi/N$ frekans kademesinde yavaş yavaş değişmesi için yeterince büyük olursa, $\sigma^2/|A_p(\omega)|^2$ işlevine göre, karmaşık üstel ifadeler ilişki matrisinin yaklaşık karakteristik vektörleridir ve karakteristik değerler tam kutuplu spektrum modelinin harmonik örnekleriyle yaklaşık hale getirilebilir [29]. Bu nedenle ayrık model spektral yoğunluğu, aşağıdaki gibi özbağlanımlı ilişki matrisinin karakteristik değerleri olarak tanımlanabilir :

$$P_k(a_k) = \frac{\sigma^2}{|A_p(\omega_k)|^2} \quad (5.23)$$

Tam kutuplu bir spektral zarf öngörü yordamında $A_p(\omega_k)$, yansıma katsayıları (ya da özbağlanım parametreleri) cinsinden gösterilebilir:

$$A_k(\omega_k) = A_{k-1}(\omega_k) + K_k B_{k-1}(\omega_k) \quad (5.24)$$

$$B_k(\omega_k) = [B_{k-1}(\omega_k) + K_k A_{k-1}(\omega_k)] e^{-j\omega_k} \quad (5.25)$$

$A_0(\omega_k)=1$ ve $B_0(\omega_k)=e^{-j\omega_k}$ 'dır. Benzerlik işlevinin (5.24) ve (5.25) eşitlikleriyle bir araya getirilmesi, bu işlevinin sadece σ^2 ve $\{K_k\}_{k=1}^p$ parametrelerine bağlı olduğunu gösterir.

5.3.1.3 Özyinelemeli Maksimum Benzerlik (RML) Algoritması

Özyinelemeli Maksimum Benzerlik (Recursive Maximum Likelihood- RML) algoritmasında, maksimum benzerlik, σ^2 ve $\{K_k\}_{k=1}^p$ 'e göre minimize edilir. Maksimum benzerlik kazancı doğrudan $\partial MI / \partial \sigma^2 = 0$ 'dan belirlenebilir ve bu eşitlik aşağıdaki sonucu doğurur [30]:

$$\hat{\sigma}_p^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2 |A_p(\omega_k)|^2 \quad (5.26)$$

Kazancın optimize edildiği benzerlik işlevi aşağıdaki gibi olur :

$$MI = N \log \hat{\sigma}_p^2 - \sum_{m=1}^p m \log(1 - K_m^2) + N \log(\pi e) \quad (5.27)$$

Optimum değerler $\{\partial MI / \partial K_m = 0\}_{m=1}^p$ eşitlikleri aynı anda çözülerek elde edilebilir. Sonuçta ortaya çıkan lineer olmayan eşitlikleri çözmek çok karmaşıktır. (5.27) eşitliğindeki ikinci terimden, optimizasyon probleminin standart LPC analizine benzer olduğu açıktır. (5.24) ve (5.25) eşitlikleri kullanarak herhangi bir yansıma katsayıları seti için bir kafes (*lattice*) filtre tanımlanabilir. Problem, şu anda, (5.27) eşitliğinde tanımlanan hata kistasını en aza indiren özel kafes filtreyi bulmaktır. *Makhoul*, belirli durumlarda global olarak optimum çözüme götüren ileri ve geri öngörü hatalarının çeşitli bileşimlerini en aza indiren kafes filtre tasarımları önermiştir [32]. Bütün durumlarda çözümler, hataların artan kafes kademeleriyle birlikte özyinelemeli olduğu gerçeğine dayanır. Bu nedenle, *McAulay*, benzerlik işlevi için kafes filtre yöntemi özelliklerinden yararlanan özyinelemeli bir biçim önermiştir. Bu yöntemde, $A_p(\omega_k)$ ve $B_p(\omega_k)$ vektörleri aşağıdaki gibi tanımlanır:

$$\langle A_p, B_p \rangle = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2 A_p(\omega_k) B_p^*(\omega_k) \quad (5.28)$$

Değişkenler tanımlanarak,

$$\alpha_p = \text{Re} \langle A_p \cdot B_p \rangle \quad (5.29)$$

$$\rho_p = \frac{\alpha_p}{\sigma_p^2} \quad (5.30)$$

kazanç faktörünün aşağıdaki gibi ifade edilebileceği gösterilebilir:

$$\hat{\sigma}_p^2 = \hat{\sigma}_{p-1}^2 (1 + 2\rho_{p-1}K_p + K_p^2) - p \log(1 - K_p^2) \quad (5.31)$$

Yukarıdaki formülde ilk değer olarak aşağıdaki değerler kullanılır:

$$\hat{\sigma}_0^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2 \quad (5.32)$$

$$\alpha_0 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2 \cos(\omega_k) \quad (5.33)$$

Kafes filtre tasarım yönteminde [30][32], maksimum benzerlik kıstası kademeli olarak minimize edilir. $\{K_m\}_{m=1}^{p-1}$ değerleri elde edildikten sonra kafesin p'inci kademesinin oluşturduğu hata $\partial M / \partial K_p = 0$ eşitliği çözülerek minimize edilir. Genel olarak bu eşitliklerin çözümü için üç kök vardır. Bunlardan biri $K_p > 1$ 'de, diğeri $K_p < -1$ 'de, üçüncüsü de $|K_p| < 1$ 'de olabilir. Ancak, (5.30) eşitliğinden, ρ_{p-1} 'in büyüklüğünün biri aşmadığı açıktır. Bu sebeple, $[-1, 1]$ aralığında daima bir kök bulunur. (5.29)'da α_p tanımını kullanarak öngörülen yansıma katsayılarını aşağıdaki gibi ifade edilebilir:

$$\hat{K}_p = - \frac{\text{Re} \langle A_{p-1} B_{p-1} \rangle}{\hat{\sigma}_{p-1}^2} \quad (5.34)$$

$N \gg p$ olan limit durumunda global maksimum benzerlik optimizasyon problemi, kafes filtrenin her bir kademesinde yaratılan birinci derece yerel minimizasyon problemi olarak çözülür. Diğer bir deyişle, özyinelemeli maksimum benzerlik yöntemi ve maksimum benzerlik yöntemi bu özel sınırlayıcı durumda özdeşdir [30].

5.3.2 Zaman Bölgesinde Doğrusal Öngörü Analizi

Doğrusal öngörü konuşma işaretleri için bir analiz-sentez sistemi sağlamaktadır. Sentez modeli, $\hat{S}$ çıkış konuşma işaretini üretmek üzere, $H(z)$ spektrum şekillendirme filtresi ve $U(z)$ uyarma kaynağından oluşur. $U(z)$ ve $H(z)$, $\hat{S}$ 'nin mümkün olduğunca $S(z)$ orijinal konuşma işaretine benzemesini sağlayacak şekilde belirli kısıtlar kullanılarak seçilir. Modelleme problemini basitleştirmek için $U(z)$ düz bir spektrum zarfına sahip olacak şekilde seçilir; böylelikle en yakın spektrum detayı $H(z)$ 'yle sınırlanmış olur. Düz spektrum mantıklı bir kabuldür, çünkü ünsüz sesler için uyarma beyaz gürültüye benzer. Ünlü sesler için kaynak, düzgün (*uniform*), P örneklerinde periyodik olan (perde periyodu) bir örnek dizisi şeklindedir ve düzgün alan harmonikleri bulunan bir çizgisel spektruma sahiptir.

Konuşma işareti $s(n)$ için $H(z)$ 'yi elde etmeyi basitleştirmek amacıyla, konuşmanın, N örneklik pencere süresince durağan (*stationary*) olduğu kabul edilir. Bu, $H(z)$ filtresinin, her bir konuşma penceresinde sabit katsayılarla modellenmesini mümkün kılar. $H(z)$ 'nin genel kutup-sıfır durumunda p tane kutba ve q tane sıfıra sahip olduğu kabul edilir. Böylece bir sentetik konuşma örneği, bir LPC sentezleyicinin önceki p çıkış örnekleri ile önceki $q+1$ giriş örneklerinin doğrusal bir kombinasyonuyla modellenebilir :

$$H(z) = \frac{\hat{S}(z)}{U(z)} = G \frac{1 + \sum_{i=1}^q b_i z^{-i}}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (5.35)$$

Burada, G kazanç faktörüdür ve $b_o=1$ alınır. Çoğu LPC modelinde $q=0$ olan bir tam kutup modeli kabul edilir. Sıfırlar atılır, çünkü bunlar spektrumun büyüklüğüne çok küçük bir katkıda bulunur ve zamandaki basit gecikmelerden kaynaklandıkları için sadece doğrusal faz ilave ederler. Bu sebeple, $H(z)$ aşağıdaki gibi tam kutuplu bir modelle ifade edilebilir:

$$H(z) = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (5.36)$$

(5.36) eşitliği örneklenmiş zaman bölgesine dönüştürülürse:

$$s(n) = Gu(n) + \sum_{k=1}^p a_k s(n-k) \quad (5.37)$$

eşitliği elde edilir. (5.37) güncel $s(n)$ çıkış değerinin, ağırlıklandırılmış güncel giriş $Gu(n)$ ve önceki çıkış örneklerinin ağırlıklandırılmış toplamı ile belirlenebileceğini gösteren, iyi bilinen bir LPC diferansiyel eşitliğidir. Buradan, LPC analizindeki problemin verilen $s(n)$ işaret ölçümleri için $\{a_k\}_{k=1}^p$ katsayılarının belirlenmesi olduğu çıkarılabilir. Elde edilen katsayıların daha sonra LPC modeli $H(z)$ transfer fonksiyonunun katsayıları olduğu kabul edilir. Hata veya arta kalan işaret aşağıdaki eşitlikle verilebilir:

$$e(n) = s(n) - \sum_{k=1}^p a_k s(n-k) \quad (5.38)$$

Ortalama öngörü hatasının karesini en aza indirmek için a_k öngörü katsayıları aşağıdaki gibi hesaplanır:

$$E_n = \sum_n e^2(n) = \sum_n \left[s(n) - \sum_{k=1}^p a_k s(n-k) \right]^2 \quad (5.39)$$

Daha sonra hatanın katsayılar göre türevi sıfıra eşitlenerek E_n minimize edilir ve aşağıdaki denklem bulunur:

$$\sum_n s(n-i)s(n) = \sum_{k=1}^p a_k \sum_n s(n-i)s(n-k) \quad 1 \leq i \leq p. \quad (5.40)$$

Aşağıdaki eşitlik tanımlandığı takdirde :

$$\phi(i, k) = \sum_n s(n-i)s(n-k) \quad (5.41)$$

(5.40) eşitliği daha kısa olarak yazılabilir :

$$\sum_{k=1}^p a_k \phi(i, k) = \phi(i, 0) \quad i = 1, 2, \dots, p. \quad (5.42)$$

p bilinmeyenli bu p denklem, $s(n)$ konuşma kısmı için E_n ortalama öngörü hatasının karesini en aza indiren katsayıları bulmak için basit şekilde çözülebilir. (5.41) eşitliği iki şekillerde yorumlanabilir ve bunlar eşitliğin çözülmesinde mümkün iki yolu (yani, özilişki (*autocorrelation*) ve ortak değişken (*covariance*) yöntemleri) gösterir [4].

5.3.2.1 Özilişki Yöntemi

(5.39) - (5.41) eşitliklerindeki toplamaların limitlerini belirlemek için konuşmanın $-N/2 \leq n \leq N/2$ aralığı dışındaki kısmının sıfır olduğu kabul edilir. (5.41) eşitliğindeki limitler $\phi(i, k)$ için aşağıdaki gibi gösterilebilir :

$$\phi(i, k) = \sum_{n=-N/2}^{N/2+p-1} s(n-i)s(n-k) \quad i = 1, \dots, p; \quad k = 0, \dots, p \quad (5.43)$$

(5.43) eşitliği, aşağıdaki gibi daha açık bir şekilde ifade edilebilen bir kısa zaman özilişkisi halinde sadeleştirilebilir :

$$\varnothing(i, k) = R(|i - k|) \quad i = 1, \dots, p; \quad k = 0, \dots, p \quad (5.44)$$

Burada,

$$R(k) = \sum_{n=-N/2}^{N/2-k} s(n)s(n+k) \quad (5.45)$$

(5.29) 'daki sonuçlar kullanıldığında ortaya aşağıdaki eşitlik çıkar:

$$\sum_{k=1}^p a_k R(|i - k|) = R(i) \quad i = 1, 2, \dots, p \quad (5.46)$$

Benzer şekilde, minimum ortalama öngörü hatasının karesi aşağıdaki eşitlikten bulunur :

$$E_n = R(0) - \sum_{k=1}^p a_k R(k) \quad (5.47)$$

(5.46) eşitliğinde verilen denklem takımı aşağıdaki gibi matris biçiminde ifade edilebilir :

$$\begin{bmatrix} R(0) & R(1) & \dots & R(p-1) \\ R(1) & R(0) & \dots & R(p-2) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ R(p-1) & R(p-2) & R(0) & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \cdot \\ \cdot \\ \cdot \\ R_{(p)} \end{bmatrix} \quad (5.48)$$

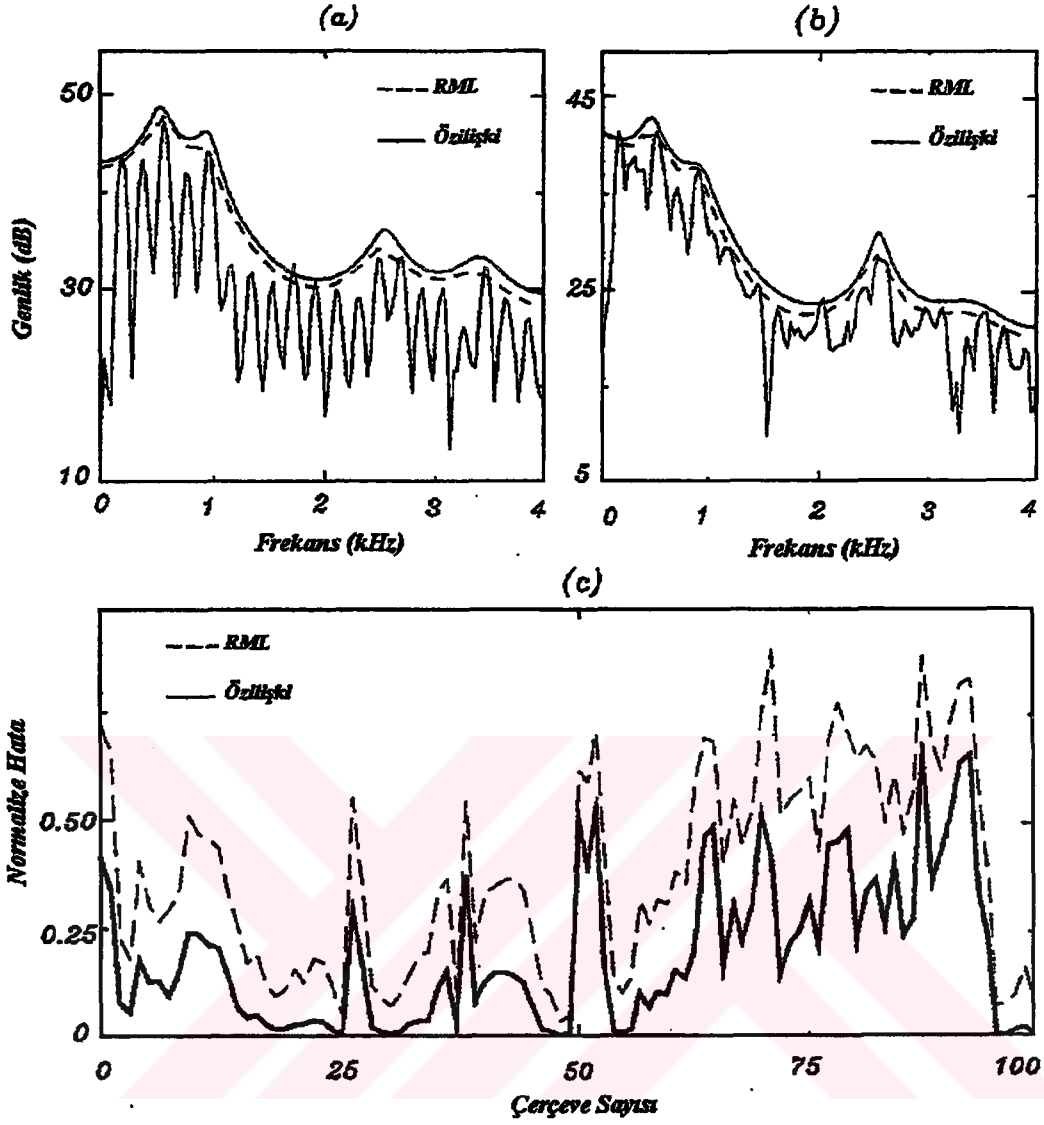
$p \times p$ matrisi, verilen herhangi bir diyagonale göre bütün elemanlarıyla simetriktr. Bu nedenle, bir *Toeplitz* matrisidir. Yukarıdaki eşitliğin çözümü, yuvarlama hatalarının

ortaya çıkabileceği güç bir hesaplama çalışmasıdır. Çözüm yöntemlerinden en yaygın kullanılanı *Durbins* algoritmasıdır ve aşağıdaki gibi özetlenebilir :

$$\begin{aligned}
 E^{(0)} &= R(0) \\
 k_i &= \left[R(i) - \sum_{j=1}^{i-1} a_j^{(i-1)} R(i-j) \right] / E^{(i-1)}, 1 \leq i \leq p \\
 a_i^{(i)} &= k_i \\
 a_j^{(i)} &= a_j^{(i-1)} - k_i a_{i-j}^{(i-1)}, 1 \leq j \leq i-1 \\
 E^{(i)} &= (1 - k_i^2) E^{(i-1)}
 \end{aligned} \tag{5.49}$$

Yukarıdaki eşitlikler, $i=1,2,...,p$ için özyinelemeli olarak nihai çözüme giden biçimde yani $\{a_j = a_j^{(p)}\}_{j=1}^p$ alınarak, çözülür.

LPC'nın tam kutuplu $H(e^{j\omega})$ spektrumu, kullanılan kutup sayısı ile sınırlıdır. Tipik olarak 10 kutuplu bir model için en çok 5 rezonans hassas olarak gösterilebilir. Harmoniklere bağlı hızlı frekans değişimine ve yanısıra rezonans yapısına bağlı daha yavaş değişimlere sahip kısa süreli, ünsüz konuşma spektrumu, böyle bir $H(e^{j\omega})$ ile tam olarak modellenemez. Düz LPC spektrumunun kaba konuşma spektrumu altındaki sapmaları, zıt yönde fark yaratmaktan ziyade asgari düzeye indirilecek artık (*residual*) hatalara katkıda bulunduğundan, $H(e^{j\omega})$, harmonikler arasındaki spektral düzlükleri harmonik tepelerden daha az takip etme eğilimindedir. Ayrıca, rezonanslar arasındaki düzlükler rezonans bölgelerine göre daha kaba olarak modellebilir. Şekil 5.2a ve 5.2b'de sırasıyla ünlü ve ünsüz konuşma kısımları için 27,5 msan'lik bir konuşma uzunluğu kullanılarak özilişki ve özyinelemeli maksimum benzerlik yöntemi LPC analiziyle elde edilen spektrum zarfları gösterilmiştir. Bu durumda, özyinelemeli maksimum benzerlik yönteminin özilişki yöntemiyle elde edilenden daha iyi bir spektrum zarfı eşlemesi oluşturduğu açık olarak görülmektedir. Yukarıda tarif edilen her bir yöntem için harmonik büyüklüklerdeki hesaplanmış normalize hatalar Şekil 5.7c'de verilmiştir. Bu şekil, ayrıca, özyinelemeli maksimum benzerlik yöntemiyle elde edilen performansın özilişki yöntemiyle elde edilen performanstan daha iyi olduğunu göstermektedir.

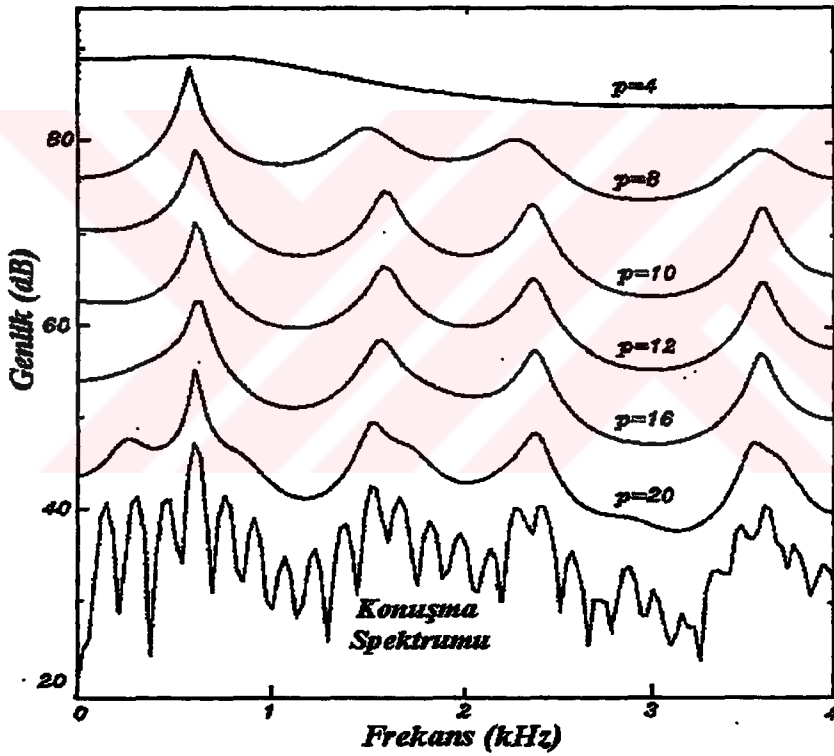


Şekil 5.7 Özilişki ve özyinelemeli maksimum benzerlik yöntemleriyle elde edilen konuşma spektrumları ve bunların zarfları : (a) ünlü konuşma için, (b) ünsüz konuşma için, (c) normalize hata

5.3.2.2 LPC Modelinin Derecesi

LPC modeli için seçilecek p derecesi, spektral duyarlılık ile hesaplama süresi/belleği ve transmisyon bant genişliği arttıkça, fazlalaşır. $p \rightarrow \infty$ şeklindeki bir limitte, $H(e^{j\omega})$ LPC spektral zarfı, $S(e^{j\omega})$ konuşma spektrumuyla tam olarak eşleşir (Şekil 5.8). Fakat, bununla birlikte gerekli bellek ve hesap yoğunluğu da artar. Genelde, işaret bant

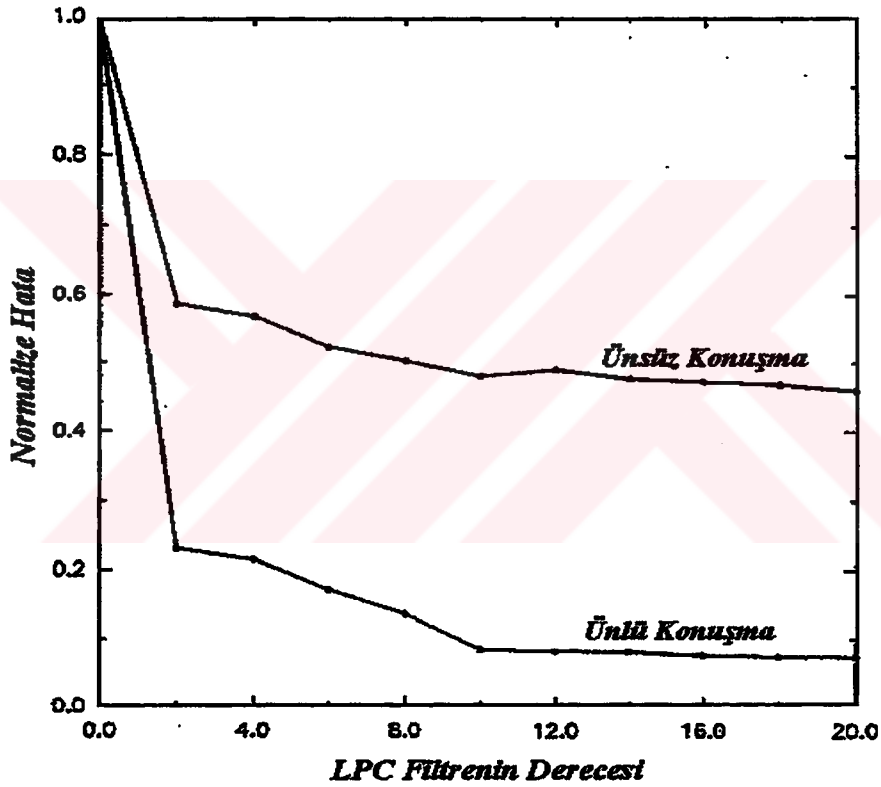
genişliğindeki bütün rezonans frekanslarını göstermek için yeterli sayıda kutup ve ilaveten spektrumdaki muhtemel sıfırlar [33] ve genel şekillendirme için 2-4 kutup daha bulunmalıdır (örneğin, 8 kHz örneklenmiş konuşma için 10 kutup). Söz konusu ikinci etki esas olarak dudak yayını ile birlikte ses telleri arası (*glottal*) dalga biçiminin spektrumundan çıkar, fakat aynı zamanda burundan çıkan (*nasal*) ve ünsüz seslerden kaynaklanan sıfırlar da gelir. Bu gibi konuşmalardaki burundan çıkan ekstra rezonanslara rağmen, burundan çıkan ses modeline daha fazla kutup eklemek genelde gereksizdir; çünkü bu tip seslerdeki yüksek frekanslı rezonanslar büyük bant genişliğine ve hassas spektral modelleme için daha az önem taşıyan çok küçük bir enerjiye sahiptir.



Şekil 5.8 Çeşitli LPC modeli dereceleri için konuşma spektrumu ve LPC spektral zarfları

Konuşma enerjisine bölünen LPC artık enerjisi (normalize hata), p öngörü dizisiyle düzenli olarak azalır (Şekil 5.4). Ünlü konuşmada, rezonans yapısını modellemek için yeterli sayıda kutup bulunduğu (yani, $p=10$ ilâ 12) ilave kutuplar spektral yapıya pek az katkıda bulunur; fakat hesaplama ve bit hızına önemli katkıları

vardır. Uyarma işareti darbeli yapıda olmadığından, ünsüz konuşma daha büyük normalize hataya sebep olur. Bu farkın nedeni, LPC katsayılarının hesaplanmasında genelde sıfırların etkisinin dikkate alınmamasıdır. Ünsüz sesler durumunda, LPC uyarmanın analiz çerçevesi üzerinde nispeten sabit bir enerji dağılımı vardır; sesli konuşmada ise LPC uyarmanın enerjisi her bir perde periyodunun başlangıcı üzerinde yoğunlaşır ve bu da konuşma örneklerinin çoğunluğu için uyarmanın dikkate alınmamasına neden olur. Bu yüzden, LPC modeli ünlü konuşmaya ünsüz konuşmadan daha uygundur, zira ünlü konuşmada, 5.39 nolu eşitlikte daha fazla zaman örneği için LPC uyarması dikkate alınmaz.



Şekil 5.9 LPC modeli dizisinin bir işlevi olarak normalize doğrusal öngörü hatası

5.4 PERDE ÖNGÖRÜSÜ

Perde (temel frekans ω_0 veya perde periyodu T_0), akustik konuşma işaretinde önemli rol oynar. Bir konuşmaya ait aksan bilgisi esas olarak bu parametreyle belirlenir. Kulak, temel frekansa, diğer konuşma işareti parametrelerindeki değişikliklerden daha duyarlıdır. Perde ölçütünün kalitesi ve hatasızlığı, kodlanan konuşmanın kalitesini belirler. Bu nedenle perde parametresinin belirlenmesi için ölçümde iyi ve güvenilir yöntemler kullanılması gerekir.

Bir konuşma işaretinin perde periyodunun tek başına akustik basıncın dalga biçiminden hassas ve güvenilir olarak belirlenmesi için aşağıdaki yorumlar geçerlidir:

- i) Ses telleri arası uyarma işareti kusursuz bir periyodik darbe dizisi değildir. Kusursuz bir periyodik dalga biçiminin periyodunun bulunması kolay olduğu halde, hem periyodu, hem de periyot içindeki dalga biçiminin yapısı değişiklik gösteren bir konuşma dalga biçiminin periyodunun ölçülmesi oldukça zordur.
- ii) Perde periyodunun ölçümündeki bir başka güçlük de, ses kanalı ile ses telleri arası uyarma arasındaki etkileşimdir. Ses kanalının rezonansları bazı açılardan ses telleri arası dalga biçiminin yapısını önemli ölçüde değiştirir ve gerçek perde periyodunu tespit etmek güç olur. Bu gibi etkileşimler, genelde, rezonansların da hızla değiştiği hızlı dil hareketi sırasında perde tespitini çok zorlaştırır.
- iii) Güvenilir perde ölçümü için, konuşmanın ünlü kısımları sırasında her bir perde periyodunun başını ve sonunu tam olarak belirlemek gerekir ki, bu da oldukça güçtür.

- iv) Ünsüz konuşma ile alçak seviyeli ünlü konuşmayı birbirinden ayırmak bazen oldukça güç olabilir. Pek çok durumda, ünsüz konuşma kısımları ile alçak seviyeli ünlü konuşma kısımları arasındaki geçiş çok keskindir ve bu sebeple de belirlenmesi son derece zordur.

Perde ölçümündeki bu güçlüklerin bir sonucu olarak çok çeşitli perde tespit yöntemleri geliştirilmiştir. Temelde, bir perde dedektörü, ünlü/ünsüz kararı veren bir birimdir ve ünlü konuşma kısımlarında perde periyodunun ölçümünü sağlar. Ancak, bazı perde tespit algoritmaları sadece konuşmanın ünlü kısımlarındaki perde frekansını tespit eder ve ünlü/ünsüz kararı için başka bazı teknikleri esas alır. Perde tespit algoritmaları kabaca aşağıdaki üç kategoriye ayrılabilir :

- i) Konuşma işaretlerinin frekans esaslı özelliklerini kullanan grup
- ii) Konuşma işaretlerinin zaman esaslı özelliklerini kullanan grup
- iii) Konuşma işaretlerinin hem frekans, hem de zaman esaslı özelliklerini kullanan grup

Perde öngörüsüyle ilgili yaklaşımın amacı, hem zaman, hem de frekans esaslı teknikleri kullanarak her bir aday temel frekans için orijinal ve sentetik konuşma spektrumu arasındaki hatayı en aza indirmektir. Her bir aday temel frekans için optimum bir sentetik konuşma dalga biçimi veya spektrumu ve buna karşılık gelen bir hata bulunur. Aday temel frekanslardan oluşan bir grupta bu hatayı değerlendirilerek en iyi perde ve spektral zarf öngörüsü bulunabilir. Bu teknikler ilerdeki bölümlerde detaylı olarak ele alınmaktadır.

5.4.1 Frekans Bölgesinde Perde Öngörüsü

Frekans bölgesinde çalışan perde dedektörleri, periyodik işaretlerin, temel frekans ve harmoniğe sahip bir dizi darbeden (veya sinüs işaretinden) oluşan bir spektrumu

bulunması özelliğinden yararlanır. Böylelikle, konuşma işaretinin periyodunu öngörmek için işaretin frekans spektrumu üzerinde basit bir ölçüm yapılabilir.

Periyodik bir işaretin temel frekans bölgesi özelliği harmonik yapısıdır. Bu harmonikler arasındaki uzaklık ise perde periyoduna karşılık düşer (*reciprocal*). Frekans bölgesinde uygulanan yöntemlerin temel sorunu yüksek hesap yükleridir. Buna rağmen modern sayısal işaret işleme teknikleri ile bu sorunun üstesinden gelinmektedir.

5.4.1.1 Harmonik Perde Öngörüsü

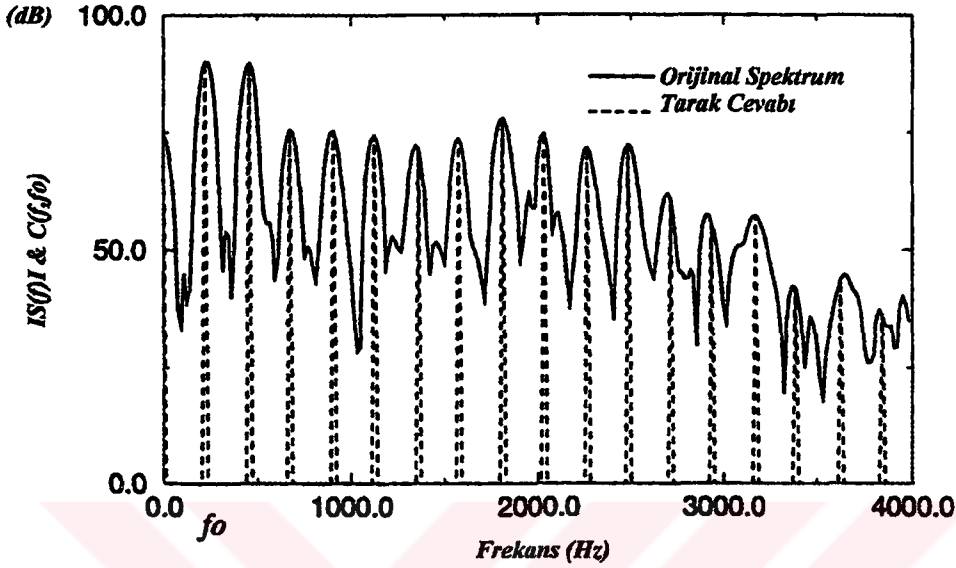
Frekans bölgesinde perde tesbitinin temel yordamlarından biri, temel frekanstaki spektral tepe değerin elde edilmesidir. Bu yaklaşım temel harmoniğin bulunmasını gerektirir ki, bu da ön filtrelemeden dolayı beklenemez. Daha pratik bir metod olarak tüm harmonik değerler bulunarak, arka arka gelen harmonikler arasındaki mesafe elde edilir (ortalama değer alınır) ve buradan da perde frekansı tesbit edilir. Bunu gerçekleştirmek için,

$$C(\omega, \omega_0) = \begin{cases} W(k\omega_0) & ; \quad \omega = k\omega_0, k = 1, 2, \dots, \Omega_m / \omega_0 \\ 0 & ; \quad \text{diğer} \end{cases} \quad (5.50)$$

şeklinde tanımlanan bir tarak (*comb*) filtre kullanılır ve bu filtre ile konuşma spektrumunun ilişkisi (*correlation*) bulunur. İşlem sonucunda bulunan çıktı ağırlıklı tarak tepe değerlerinin toplamından oluşur.

$$A_c(\omega_0) = \frac{\omega_0}{\Omega_m} \sum_{k=1}^{\Omega_m / \omega_0} S(k\omega_0) W(k\omega_0) \quad 2\pi / \tau_{\min} \leq \omega_0 \leq 2\pi / \tau_{\max} \quad (5.51)$$

Burada Ω_m konuşma spektrumundaki en büyük frekans değeridir. Eğer ω_0 temel frekansa eşit ise, orijinal spektrumun tarak cevabı Şekil 5.10'da görüldüğü gibi harmonik tepe değerlerini verecektir.



Şekil 5.10 Harmonik perde öngörüsü metodu

Daha iyi bir sonuç elde etmek için, her bir tarak değeri için artan frekansla birlikte azalan bir ağırlıklandırma katsayısı kullanılabilir.

5.4.1.2 Spektrum Temelli Perde Öngörüsü

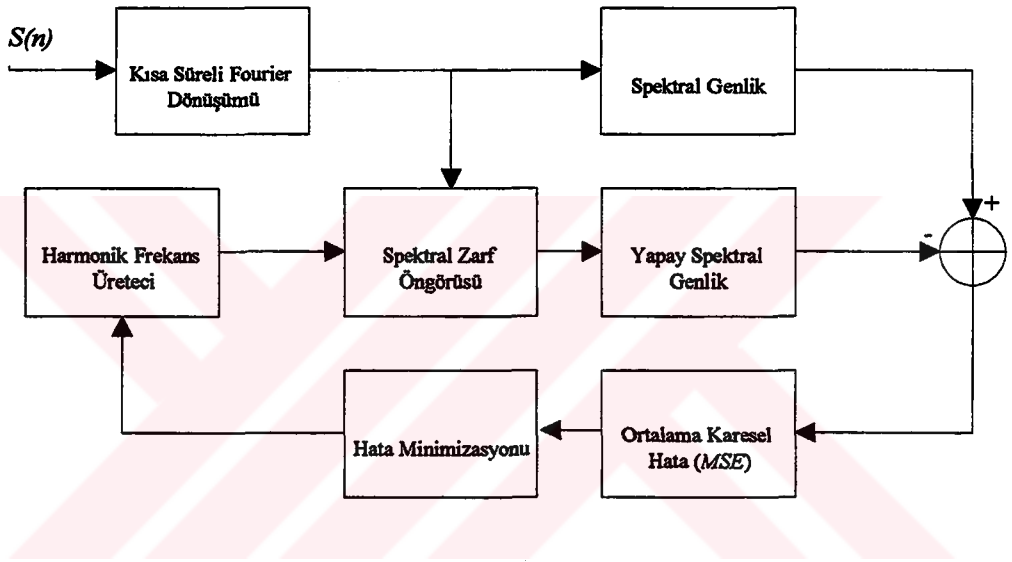
Spektrum temelli perde öngörüsü algoritmasının sadeleştirilmiş blok şeması Şekil 5.11'de görülmektedir.

Amaçlanan, aşağıda verildiği gibi, orijinal ve yapay konuşma spektrumu arasında minimum ortalama karesel hatayı (*Mean Squared Error-MSE*) oluşturan optimum temel frekansı seçmektir :

$$E(\omega_0) = \sum_{\omega=0}^{N-1} G(\omega) \left| S(\omega) - S(\omega, \omega_0) \right|^2 \quad (5.52)$$

$\hat{S}(\omega, \omega_0)$ yapay konuşma spektrumu ve $G(\omega)$ 'da frekansa bağlı ağırlık ölçüm işlevidir. Bu formülde, yapay konuşma spektrumu her bir aday temel frekans ve konuşmanın tamamen ünlü olduğu kabulüne dayanan eşdeğer harmonik spektrum büyüklükleri kullanılarak üretilir. Yapay konuşma spektrumu aşağıdaki eşitlik kullanılarak hesaplanabilir :

$$\hat{S}(\omega, \omega_0) = \sum_{m=0}^L A_m W(\omega - m\omega_0) \quad (5.53)$$



Şekil 5.11 Spektrum temelli perde öngörü tekniğinin blok şeması

$\{A_m\}_{m=0}^L$ karmaşık spektrum genlikleri, L harmonik sayısı ve $W(\omega)$, $\omega(n)$ pencere işlevinin frekans cevabıdır. Hamming penceresi benzeri pencerelerin frekans cevabı, genişliği gerçek temel frekanstan az olan etkin bir ana loba sahiptir. Bu nedenle, iki bitişik harmonik arasındaki örtüşme (*aliasing*) dikkate alınmayabilir. $W(\omega - m\omega_0)$ 'ın impuls niteliği bulunduğundan, konuşma spektrum zarfının bir küme L karmaşık spektrum genlikleri grubuyla, yani $\{A_m\}_{m=0}^L$ ile yaklaşık hale getirilmesi mümkün olur. Bu genlikler, her bir harmonik loba karşılık gelen hata en aza indirilerek hesaplanır. Böylelikle m 'inci harmonik lobdaki hata aşağıdaki gibi ifade edilebilir :

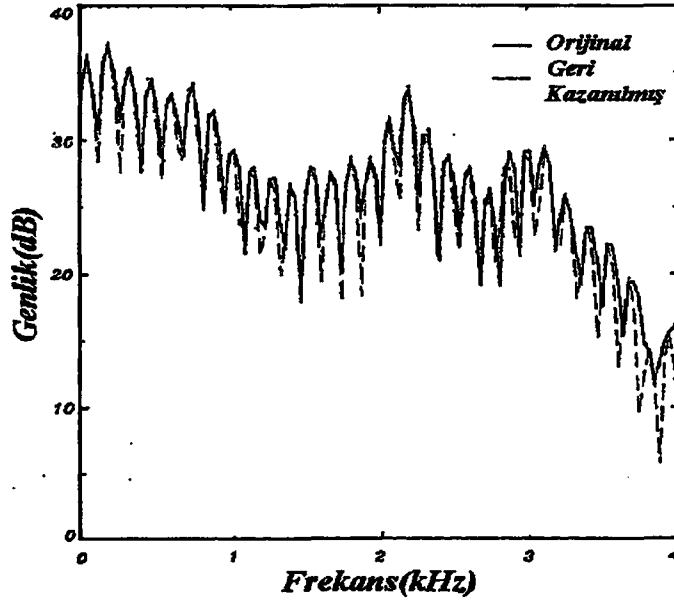
$$E_m(\omega_0) = \sum_{\omega=a_m}^{b_m} G(\omega) |S(\omega) - A_m W(\omega - m\omega_0)|^2 \quad (5.54)$$

Her bir harmonik lobun $m\omega_0$ 'da ($0 \leq m < L$) merkezlenen ω_0 (toplama aralığı) genişliği vardır. Bundan dolayı, her bir harmonik lobun üst ve alt limiti sırasıyla, $a_m = (m-1/2)\omega_0$ ve $b_m = (m+1/2)\omega_0$ 'dır. Optimum spektrum genliklerini elde etmek için 5.41 eşitliği A_m 'ye göre türetilir. Elde edilen optimum spektrum genlikleri :

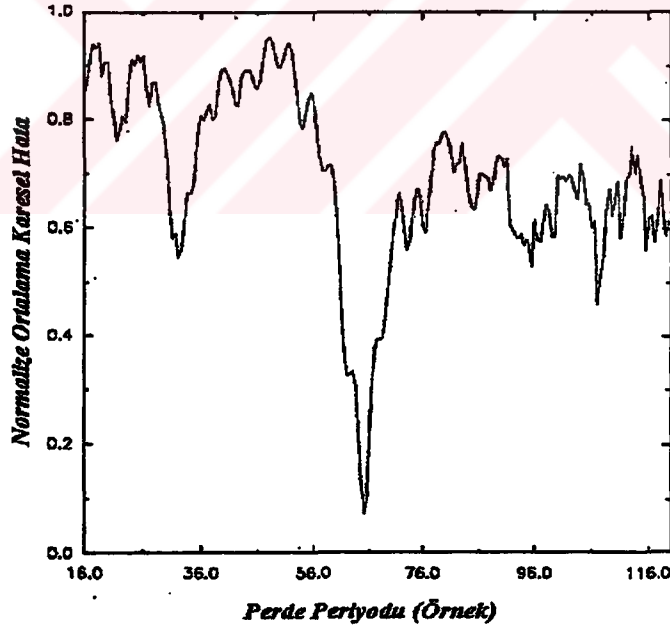
$$A_m = \frac{\sum_{\omega=a_m}^{b_m} G(\omega) S(\omega) W^*(\omega - m\omega_0)}{\sum_{\omega=a_m}^{b_m} G(\omega) |W(\omega - m\omega_0)|^2} \quad (5.55)$$

şeklinde olacaktır. Her bir ω_0 aday temel frekansı için spektrum genlikleri, belirli bir harmonik lob genişliğinde hatayı en aza indiren (5.55) eşitliği kullanılarak üretilir. Bir konuşma spektrumu harmoniğinin periyodik enerjisi varsa, enerji karakteristik pencere frekans cevabı şekline sahip temel harmonikte merkezlenir. Sonuç olarak, m 'inci konuşma spektrumu harmoniğini yeniden oluşturmak için A_m spektrum genliğiyle ölçülen bu frekans cevabı kullanılırsa, ilgili harmonik lob için iyi bir eşleme elde edilir. Bir konuşma spektrumu harmoniğinin periyodik olmayan bir enerjisi varsa, bu harmonik için karakteristik şekil olmaz ve yeniden oluşturulan konuşma spektrumu olarak pencere frekans cevabı kullanıldığında iyi bir eşleme sağlanamaz. Bu, optimum perde frekansı kullanılarak Şekil 5.12'de gösterilmiştir. Sözkonusu yordam, her bir aday temel frekansın bütün harmonikleri için tekrarlanır ve harmonikler için yaratılan hatalar toplanarak, mümkün olan her aday temel frekans için tüm spektrumda $E_T(\omega_0)$ toplam hata bulunur. $E_T(\omega_0)$ 'nin minimum değeri, aday temel frekans grubundaki en iyi temel frekansa karşılık gelir. Deneysel olarak $E_T(\omega_0)$ hatası, Şekil 5.13'de gösterilen perde periyoduyla yavaş yavaş değişir. Bu özellik, hatayı kaba bir ızgara (*grid*) üzerinde hesaplayarak, elde edilecek global minimum değere yakın perde periyodunun başlangıç öngörüsünü elde etmede kullanılabilir. Pratikte, başlangıç öngörüsü, tamsayı perde periyotları için $E_T(\omega_0)$ hatası hesaplanarak elde edilir. Bu kaba perde periyodu başlangıç öngörüsünde,

yüksek frekans harmonikleri iyi eşleşemez ve bu nedenle de yüksek frekansların önemini azaltmak için frekans ağırlıklandırılmış $G(\omega)$ fonksiyonu seçilir.



Şekil 5.12 Optimum perde frekansı için orijinal ve yeniden oluşturulmuş spektrumlar



Şekil 5.13 Normalize ortalama karesel hata ile perde periyodu ilişkisi

5.4.2 Zaman Bölgesinde Perde Öngörüsü

Zaman bölgesinde perde dedektörleri periyodu öngörmek için doğrudan konuşmanın dalga şekli üzerinde çalışır. Bu tip perde dedektörlerinde yapılan ölçümler genellikle işaret dalga şekli için tepe ve vadi ölçümleri, sıfır geçiş ölçümleri ve özilişki ölçümleridir. Bütün durumlarda, rezonans yapısının etkilerini en aza indirmek için bir yarı periyodik işaretin uygun şekilde işlenmesi halinde, basit zaman bölgesi ölçümlerinin iyi periyot öngörülerini sağlayacağı kabulü yapılır. Bu tipte en popüler algoritma, aşağıda gösterilen özilişki esaslı perde öngörüsüdür.

5.4.2.1 Özilişki Esaslı Perde Öngörüsü

Zaman bölgesinde özilişki esaslı perde öngörü algoritmasının temel avantajı hesaplanmasının kolay olmasıdır. Zaman esaslı yaklaşım, tamsayı perde periyotları için frekans esaslı yaklaşıma hemen hemen eşittir. Zaman esaslı özilişki işlevinin aşağıdaki gibi ifade edilebilir:

$$\psi(P) = P \sum_{k=-N/2}^{N/2} \Phi(kP) \quad (5.56)$$

Burada, $\Phi(k)$ (tamsayı k değerleri için) $s(n)\omega^2(n)$ 'nin özilişki işlevidir ve aşağıdaki gibi yazılabilir :

$$\Phi(k) = \sum_{n=-N/2}^{N/2} s(n)\omega^2(n)s(n-k)\omega^2(n-k) \quad (5.57)$$

$\Phi(k)$ fonksiyonu aradeğerleme (*interpolation*) ile tamsayı olmayan k değerleri için tamamlanır:

$$\Phi(k) = (1 + \lfloor k \rfloor - k)\Phi(\lfloor k \rfloor) + (k - \lfloor k \rfloor)\Phi(\lfloor k \rfloor + 1) \quad (5.58)$$

$\lfloor k \rfloor$, k 'ya eşit veya daha küçük olan en büyük tamsayıya eşittir. Bu sebeple, $\Psi(P)$ 'nin maksimumu, yaklaşık olarak analiz penceresinin karesiyle çarpılan işaretin özilişki işlevine ait bir işlevin maksimumuna eşittir. Bu teknik özilişki yöntemine benzer, fakat sadece perde periyodundaki tepelerle değil, perde periyodunun katlarındaki tepelerle de ilgilenir. Böylelikle, Hızlı Fourier Dönüşümü (*Fast Fourier Transform-FFT*)'ne benzer bir yordam kullanarak özilişki işlevini hesaplamak ve sonra perde periyoduyla yerleştirilen örnekleri toplamak suretiyle bütün tamsayı perde periyotlarında $\Psi(P)$ 'nin maksimumu için etkin bir hesaplama yöntemi elde edilmiş olur.

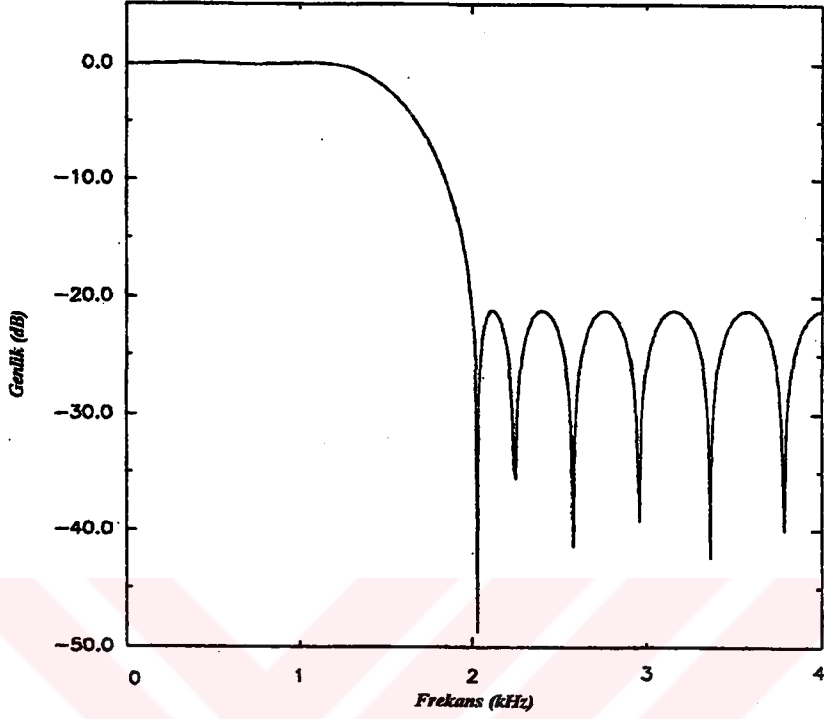
Tamsayı olmayan P perde periyotlarında $\Psi(P)$ 'nin maksimizasyonu daha hassas perde periyodu elde edilebilir. Bu yöntemde, özilişki fonksiyonunda perde periyodunun katlarında tepelerin katkılarına bağlı olarak daha yüksek duyarlık elde edilir. Frekans esaslı perde öngörüsüne eşdeğer olan $E_T(P)$ hata fonksiyonu aşağıda verilen özilişki yaklaşımı kullanılarak hesaplanabilir :

$$E_T(P) = \sum_{n=-N/2}^{N/2} [s(n)\omega(n)]^2 - P \sum_{k=-N/2}^{N/2} \Phi(kP) \quad (5.59)$$

Optimum perde periyodunun kolay tespit edildiği düzgün bir $E_T(P)$ hata işlevi elde etmek için $s(n)$ konuşma işareti aşağıdaki gibi alçak geçiren filtreden geçirilir:

$$s(n) = \sum_{k=-10}^{10} s_0(n-k)LPF(k) \quad (5.60)$$

Burada, $s_0(n)$ orijinal konuşma işareti ve $LPF(k)$ 21. dereceden FIR filtresidir . Bu filtrenin frekans cevabı Şekil 5.14'de gösterilmiştir ve kesim frekansı 1574 Hz'dir.



Şekil 5.14 21. kademe FIR filtresinin frekans cevabı

Elde edilen $E_T(P)$ hata fonksiyonu, perde periyodu için ilk öngöründe kullanılabilir. Bu ilk öngörü oluşturmada kullanılabilecek basit bir yaklaşım, $E_T(P)$ 'nin minimum değerini veren perde periyodunun seçilmesidir. Ancak, $E_T(P)$ 'de bulunan bilgiyi perde sürekliliğindeki bazı sınırlamalarla bir araya getirmek ve böylece aşağıda tarif edilen şekilde perde öngörüsünü geliştirmek daha iyi bir tekniktir.

5.4.3 Perde Pekiştirme

Hassas ve hatasız bir perde öngörüsü için, perde periyodu elde edilmeden önce, daha önce ele alınan algoritmalar kullanılarak hesaplanan perde periyodu bazı ek işlemlere tabi tutulmalıdır. Bu ek işlemler arasında yanlılık (öngörü değeri ile gerçek

parametre değeri arasındaki fark-bias) giderme, perde izleme (*tracking*) ve perde astkat (*sub-multiple*) kontrolü bulunmaktadır.

5.4.3.1 Yanlılık Giderme

(5.54) eşitliğindeki MSE fonksiyonunun, uzun perde periyotları için beklenen küçük bir değere sahiptir. Bu etki Şekil 5.16'de açık olarak görülmektedir. Bu şekilden, uzun perde periyodu seçme olasılığının yüksek olduğu görülebilir. Genelde, uzun perde periyotları için hata azalır. Böylece, MSE hata işlevi $E_T(P)$, uzun perde periyotları kısa perde periyotlarına tercih edilecek şekilde bir eğilim gösterir. Konuşmanın beyaz gürültü şeklinde bir periyodik işaret olarak modellenmesi halinde MSE hata işlevi $E_T(P)$ 'nin beklenen değeri aşağıdaki eşitlikle ifade edilebilir:

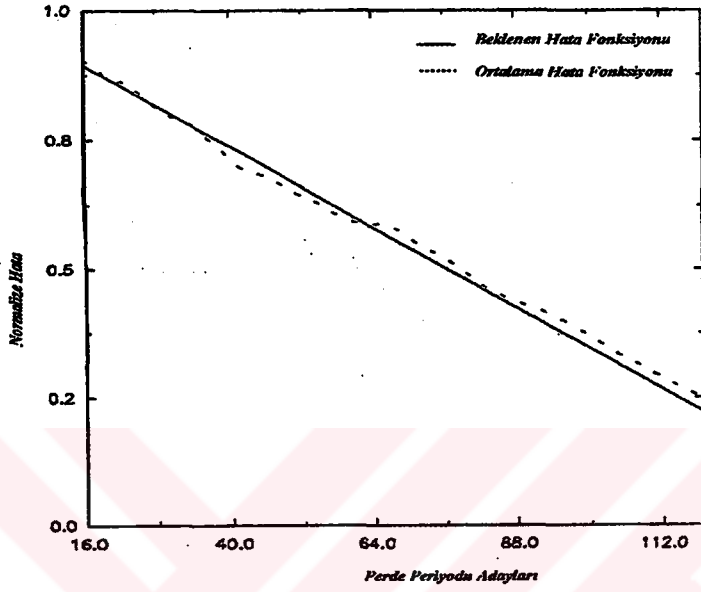
$$E[E_T(P)] \approx \sigma^2 \left(1 - P \sum_{n=-N/2}^{N/2} \omega^4(n) \right) \quad (5.61)$$

Burada, σ , beyaz Gauss gürültüsünün standart sapmasıdır, $\omega(n)$ ise $\sum_{n=-N/2}^{N/2} \omega^2(n) = 1$ sınırlamasını sağlayacak şekilde seçilmiştir. (5.61) eşitliği ile verilen bu yanlılık yaklaşımının hassasiyetini belirlemek için Şekil 5.15'de görüldüğü gibi $E[E_T(P)]$ eğrileri ve ortalama hataya (3500 pencerenin ortalaması) karşılık aday perde periyotları elde edilir. Bu şekilden görülebileceği gibi, yanlılık yaklaşımı ortalama hata eğrisine çok yakındır.

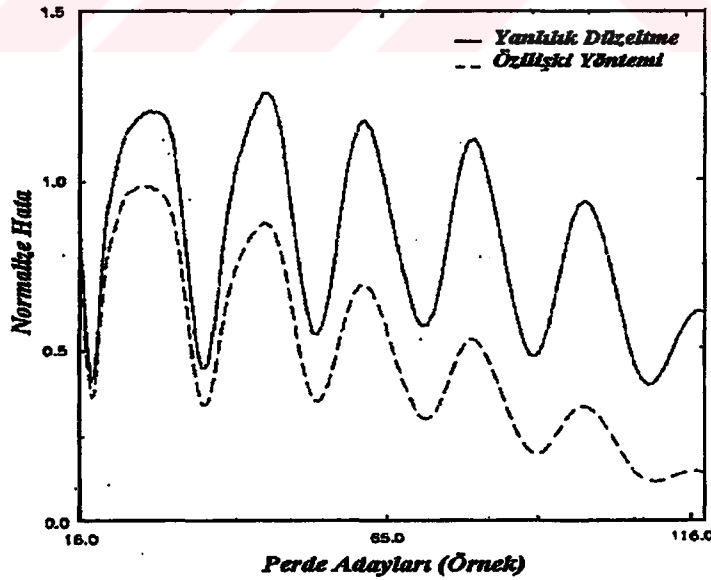
Sürekli olarak kısa perde periyotlarından çok uzun perde periyotlarının seçilmesi problemini gidermek için, tam periyodik bir işarette sifıra veya bir gürültü işaretinde bire yakın değerlere sahip olacak şekilde, enerji normalizasyonu yapılmış bir yanlılık düzeltmeli hata kıstası geliştirilmiştir. Bu hata kıstası aşağıdaki gibi ifade edilebilir:

$$E_T(P) \approx \frac{E_T(P)}{\sum_{n=-N/2}^{N/2} [s(n)\omega(n)]^2 \left(1 - P \sum_{n=-N/2}^{N/2} \omega^4(n) \right)} \quad (5.62)$$

(5.62) eşitliğinden görülebileceği gibi, ifade, σ^2 gürültü varyansından bağımsızdır ($\sigma^2=1$ kabul edilmektedir). Bu nedenle, bu varyansın öngörüsüne gerek yoktur. Yanlılık düzeltmesinin etkisi Şekil 5.16'da gösterilmiştir ve bu düzeltme, perde periyodu olarak seçilecek her bir aday periyod için hemen hemen eşit olasılık sağlar.



Şekil 5.15 Perde öngörüsü için gerekli yanlılık düzeltmesi



Şekil 5.16 Perde öngörüsü için yanlılık düzeltmenin etkisi

5.4.3.2 Perde İzleme

Bir çerçeveden diğerine düzgün geçiş sağlamak için bir perde izleme algoritması kullanılır. Bu düzgünleştirme işlemi, genel perde hatalarının azaltılmasında önemli rol oynar. Bu teknikle ilgili temel problem, düzgünleştirme işleminde perde periyodu öngörü hassasiyetinin temiz konuşmada bile bozulmasıdır. Dikkate alınacak üç perde izleme koşulu vardır :

- i) Perde izlemesinin mevcut çerçeveden başlaması
- ii) Perde izlemesinin mevcut çerçevede sona ermesi
- iii) Perde izlemesinin mevcut çerçevede devam etmesi

Bu koşullardan anlaşılabilceği gibi, üçüncü koşul ilk iki koşulun biriyle uygun şekilde modellenir. Mevcut çerçeveden başlayan veya orada sona eren en iyi perde izlemesinin bulunması istenmektedir. Bu amaçla, M 'nin önemsiz bir gecikme oluşacak kadar küçük olduğu durumlarda, M çerçeve kadar ileri ve geri bakılır (tipik olarak $M=3$).

Daha sonra, geçmişteki M çerçeveden mevcut çerçeveye kadar ve gelecekteki M çerçeveden mevcut çerçeveye kadar minimum toplam hata seçilir. En küçük toplam hataya sahip yol belirlenir. Belirlenen minimum toplam hata, mevcut konuşma çerçevesi için perde periyodunun (veya temel frekansın) bir başlangıç öngörüsüne karşılık gelir. Bu “geriye doğru” ve “ileriye doğru” perde izleme algoritmaları aşağıdaki gibi açıklanabilir :

- Geriye Doğru Perde İzleme:

Geriye doğru perde izleme algoritmasında, önceki konuşma çerçevelerine göre perde sürekli kabul edildiğinden, mevcut konuşma çerçevesi için perde periyodu, önceki çerçevenin (P_{-1}) ilk perde periyodu civarındaki bir aralıkta hesaplanır. $E_T(P)$ hata işlevinin aşağıdaki perde aralığına düşen her bir P değerinde olduğu düşünülür:

$$(1 - a)P_{-1} \leq P \leq (1 + a)P_{-1} \quad (5.63)$$

Burada, α , aramanın perde aralığını belirler (tipik olarak $\alpha=0,2$). Yukarıda verilen aralıktaki $E_T(P)$ değerleri karşılaştırılır ve P_b geriye doğru tizliği, tanımlanan perde arama aralığındaki $E_T(P)$ 'yi en aza indiren P değeri olarak belirlenir. Daha sonra $C_b P_b$ geriye doğru toplam hata hesaplanır :

$$C_b(P_b) = \hat{E}_T(P_b) + \sum_{i=1}^{M-1} \hat{E}_{T-i}(P_{-i}) \quad (5.64)$$

Burada, $\left\{P_{-i}, \hat{E}_{T-i}(P_{-i})\right\}_{i=1}^{M-1}$ ilk perde öngörülleri ve bunlara karşılık gelen hata değerleridir (önceki $M-1$ çerçeveler için). Geriye doğru (*backward*) toplam hata, geriye doğru perde öngörüsü için güvenli bir yaklaşım sağlar. Bunun nedeni, önceki konuşma çerçevelerinin ilk perde periyotlarının bilinmesidir.

- İleriye Doğru Perde İzleme:

İleriye doğru (*forward*) perde izleme yordamı gelecekteki konuşma çerçeveleri arasındaki perde sürekliliğiyle ilgilidir. Gelecekteki $M-1$ çerçeve için perde belirlenmediğinden, ileriye doğru perde izleme algoritması bu çerçevelerin perde frekansını seçmek durumundadır. Bu işlem, şu şekilde yapılır : önce, P (mevcut çerçeve)'nin sabit olduğu kabul edilir. Sonra, (5.65) ifadesinde verilen aralıkta

$\left\{\hat{E}_{T_i}(P_i)\right\}_{i=1}^{M-1}$ terimini en aza indirmek için $\{P_i\}_{i=1}^{M-1}$ bulunur :

$$(1-a)P_i \leq P_{i+1} \leq (1+a)P_i; \quad i = 0,1,\dots,M-1. \quad (5.65)$$

$\{P_i\}_{i=1}^{M-1}$ hesaplandığında geriye doğru perde izleme yordamında olduğu gibi $C_f(P)$ ileriye doğru toplam hata işlevi hesaplanır :

$$C_f(P) = \hat{E}_T(P) + \sum_{i=1}^{M-1} \hat{E}_{T_i}(P_i) \quad (5.66)$$

Bu yordam her aday P için tekrarlanır. Denk düşen $C_f(P)$ değerleri karşılaştırılır ve perde tarama aralığında minimum $C_f(P)$ değerini oluşturan P_f ileriye doğru perde olarak seçilir.

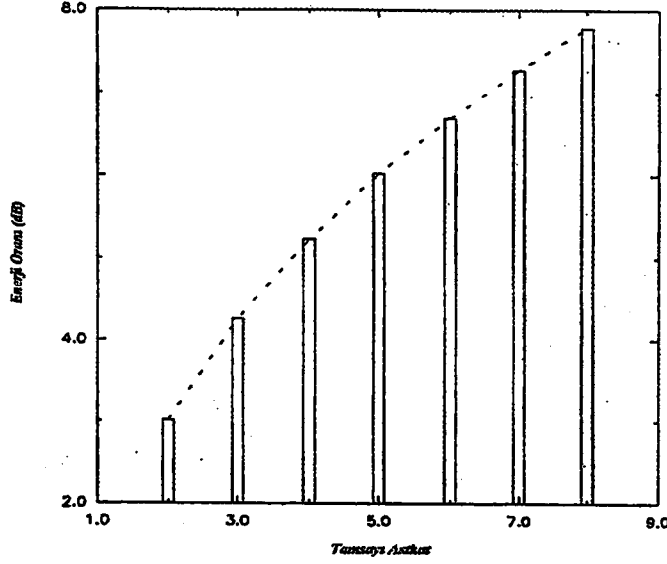
$C_b(P_b)$ geri toplam hatası ileriye doğru toplam hatayla karşılaştırılır. Bu karşılaştırma, mevcut çerçeve için P_o ilk perde öngörüsü olarak ileri perde öngörüsünün veya geri perde öngörüsünün seçilip seçilmemesini belirler.

5.4.3.3 Perde Astkat Kontrolü

İlk perde öngörüsü sırasında perde çiftleme hataları olabilir ve bunlar da sentezlenen konuşmada harmonik distorsiyon oluşmasına neden olur. Perde çiftleme hatalarının meydana gelme oranını azaltmak için yeni bir algoritma geliştirilmiştir. Yeni algoritmanın ilk aşaması, ilk tizliğin (P_o) tamsayı astkatlarının ($P_o/2, P_o/3, \dots, P_o/n$) bulunmasıdır ve bunlar perde tarama algoritmasının değer kümesinin içindedir. Bir sonraki aşama ise her bir astkat adayı için ortalama harmonik enerjinin aşağıdaki gibi hesaplanmasıdır :

$$E(\omega_k) = \frac{1}{L_k} \sum_{i=1}^{L_k} A(i\omega_k) ; k = 1, 2, 3, \dots, n. \quad (5.67)$$

Burada, L_k 4 kHz konuşma bandı genişliğindeki harmonik sayısı, $A(i\omega_k)$ harmonik büyüklükler ve $\omega_k = 2\pi/(P_o/k)$ 'da tizliğin k 'ıncı astkatının frekansıdır. Daha sonra, en küçük astkatın enerjisi ile birinci astkatın (P_o) enerjisi arasındaki oran hesaplanır ve Şekil 5.17'de görüldüğü gibi her bir astkat için değişen bir adaptif eşikle karşılaştırılır. Bu oran denk düşen eşikten büyük olursa, ilk perde öngörüsü olarak seçilir. Aksi takdirde, bir sonraki en büyük astkat yukarıdaki yordama göre kontrol edilir. Bu yordam, bütün astkatlar bu koşul açısından kontrol edilene kadar devam eder. İlk tizliğin astkatlarının hiçbiri bu koşulu yerine getirmese ilk perde öngörüsü olarak P_o seçilir.

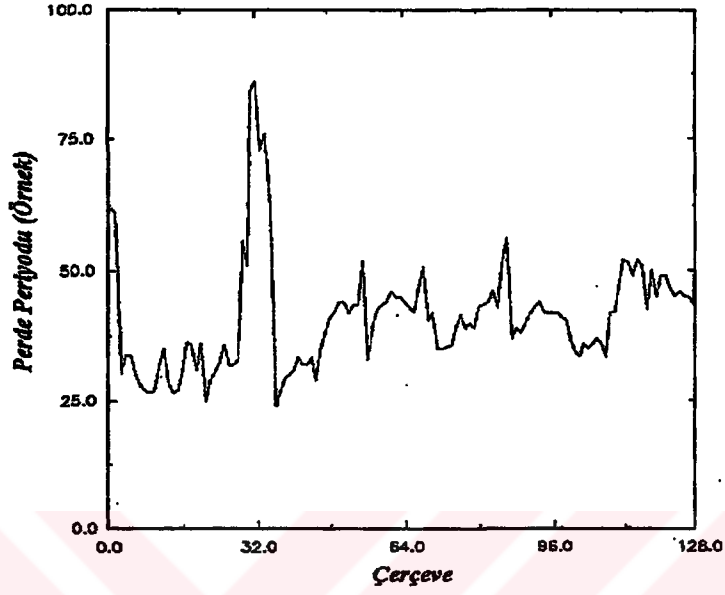


Şekil 5.17 Adaptif eşik fonksiyonu

5.4.3.4 Perde Öngörüsünde Duyarlılık Artışı

Gerçek perde periyodunun katları atıldığında yeni perde öngörüsünün hassasiyeti artırılmalıdır. Daha önce de bahsedildiği gibi, yüksek kaliteli düşük bit hızlı kodlama sistemleri için, perde periyoduna ters orantılı olan temel frekans hassas şekilde bilinmelidir. Özilişki perde öngörü algoritması esas olarak tamsayı perde periyotlarıyla sınırlı olduğundan yeterli hassasiyet sağlamayabilir. Daha hassas perde öngörüsü için hata işlevi, ilk perde periyodu öngörüsü civarında küçük bir bantta, daha küçük aralıklarda yeniden değerlendirilir. Bu aralıkta minimum hataya sebep olan temel frekans, hassas perde öngörüsü olarak alınır. Bu yordam, istenen hassasiyet elde edilene kadar daha küçük aralıklarda da tekrarlanabilir. Başlıca problem ilk öngörünün elde edilmesindedir. İlk öngöründe hata, çok sayıda aday temel frekans için değerlendirilmelidir; ikinci kademedede ise hatanın sadece yaklaşık 5-10 aday temel frekansta değerlendirilmesi gerekir. Öngörünün kalıcı hassasiyeti ikinci kademedede değerlendirildiğinden, sistemin performansını olumsuz yönde etkilemeden ilk öngörünün elde edilmesinde daha düşük hassasiyette bir teknik kullanılabilir.

Şekil 5.18’de 128 çerçevede perde periyodu değişimi gösterilmiştir. Bu şekilden, bütün perde geliştirme teknikleri uygulandığında, bir çerçeveden diğerine daha yavaş perde periyodu değişimleri elde edildiği açıkça görülmektedir.



Şekil 5.18 Bir çerçeveden diğerine perde periyodu değişimi

BÖLÜM 6

SENTEZLEYEREK ANALİZ

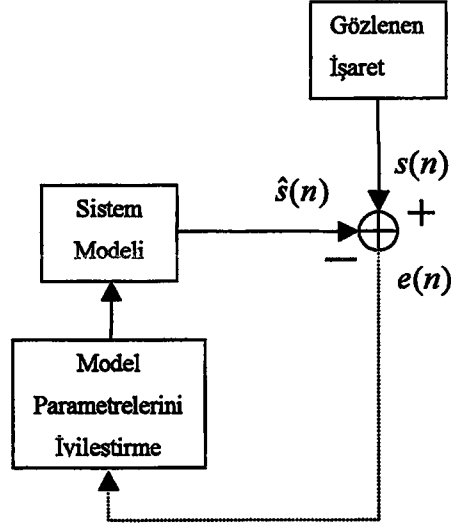
6.1 SENTEZLEYEREK ANALİZ YAKLAŞIMI

4,8-16 kbit/san arasındaki konuşma kodlama tekniklerinde; analiz ve sentez (*Analysis and Synthesis-AaS*) yöntemi ve sentezleyerek analiz (*Analysis by Synthesis-AbS*) yöntemi olmak üzere iki temel yaklaşım bulunmaktadır. AaS yönteminde konuşma işareti analiz edilerek, işaret içindeki artık (*redundant*) bilginin atılması için gerekli olan parametreler seçilip çıkartılır ve kalan işaret nicemlenir. Alıcıda ise aynı işlemin tersi yapılır, yani nicemlenmiş artık işaret seçilip çıkartılan parametreler yardımıyla sentezlenerek orijinal işaret yeniden elde edilir. Bu yöntem parametre seçip çıkartma ve nicemleme işlemlerini ayırmakta ve bundan dolayı da sistemde ortaya çıkan bozulmalara karşı kontrol, alt-sistemler ile sınırlı kalmaktadır. Tüm kodlama süreci üzerinde daha fazla kontrole sahip olmak, yani yapay konuşma işaretindeki tüm hatayı minimize etmek için sentezleyerek analiz teknikleri önerilmektedir. Sentezleyerek analiz yönteminde her alt-sistem, yapay işaretin tamamında en az bozulma olacak şekilde, birlikte optimize edilir. Bu, verici kodlayıcıda bir kodçözücü bulundurmak suretiyle, sentezlenen işaretin aynısı vericide de oluşturulup analiz için kullanılmasını sağlayarak gerçekleştirilebilmektedir. RELP [32], APC [33], ATC [33] ve SBC [33] gibi AaS yaklaşımları 9,6-16 kbit/san civarındaki hızlarda çok başarılı olmasına rağmen 9,6 kbit/san'nin altındaki hızlarda kaliteli konuşma üretemez. Bu kısıtın iki ana nedeni vardır :

- (i) kodlama prosedürünün etkin şekilde çalışıp çalışmadığını görmek için kodlanmış konuşma analiz edilemez, yani yeniden oluşturulan konuşmanın bozulmaları üzerinde kontrol yoktur,
- (ii) adaptif yöntemlerde önceki çerçevelerden ortaya çıkan hatalar mevcut analiz çerçevesinde dikkate alınmaz ve dolayısıyla, hatalar, herhangi bir engel olmaksızın sonraki çerçevelere yayılır.

Özellikle AbS-LPC yöntemi [34, 35] olmak üzere AbS yöntemlerinde bu iki faktör bir arada bulunur. AbS-LPC kodlama sistemlerinde, “filtre modeli”ni uyarmak için kullanıldığında optimum yeniden üretilen konuşma işaretini üreten uyarma işaretini belirlemek için bir “kapalı çevrim” optimizasyon yordamı kullanılır. AbS-LPC kodlama yöntemlerinin 4,8-9,6 kbit/san’de APC ve RELP gibi klasik AaS yapılarından çok daha başarılı olmasını sağlayan bu “kapalı çevrim” yaklaşımıdır.

AbS yöntemi, konuşma kodlamada tek yöntem değildir, fakat diğer tahmin ve tanımlama alanlarında kullanılan genel bir tekniktir. AbS’nin temelinde yatan genel yaklaşım şudur: Öncelikle, işaretin zaman veya frekans bölgesinde bir şekilde gözlenebileceği ve gösterilebileceği kabul edilir. Daha sonra, işaret üretim modelinin Şekil 6.1’de gösterilen bir teorik şekli kabul edilir. Model, gözlenebilen işaretin değişik aralıklarını üretmek üzere değiştirilebilecek bir dizi parametreye sahiptir. Gerçek işaret modeliyle aynı şekilde bir model gösterimi ortaya koymak için bir deneme-yanılma yöntemi uygulanabilir. Modelin parametrelerini sistematik bir yöntemle değiştirerek, en az hata ile gerçek işarete eş bir yapay işaret üretebilen bir parametre kümesi bulmak mümkündür. Bu nedenle, bu gibi bir eşleme hesaplandığında, modelin parametrelerinin gerçek işaretin parametreleri olduğu kabul edilir.



Şekil 6.1 AbS kapalı çevrim analizi blok yapısı

Daha eski uygulamalarda konuşma işleme için, rezonans (*formant*) öngörülerinde [4] yukarıda tanımlanan AbS prosedürü uygulanmış; fakat belirgin bir hesap karmaşıklığı bulunduğundan düşük bit hızlı kodlama için *Atal* tarafından MPLPC [36] tarif edilene kadar bir daha kullanılmamıştır. *Atal*'ın çalışmasında konuşmanın zaman bölgesinde gösterimi kullanılmıştır ve klasik “kaynak-filtre” modeline çok benzer bir model seçilmiştir. Ancak, başka esas ve modellere sahip AbS yordamları da eşit derecede uygulanabilirliğe sahiptir [37].

6.2 GENEL ABS KODLAMASI

AbS-LPC kodlama sisteminin temel yapısı Şekil 6.2’de gösterilmiştir. Modelde, gerçek model ile eşleşecek şekilde değiştirilebilen ve böylece kaliteli yapay konuşma işareti elde edilebilen üç alt blok vardır :

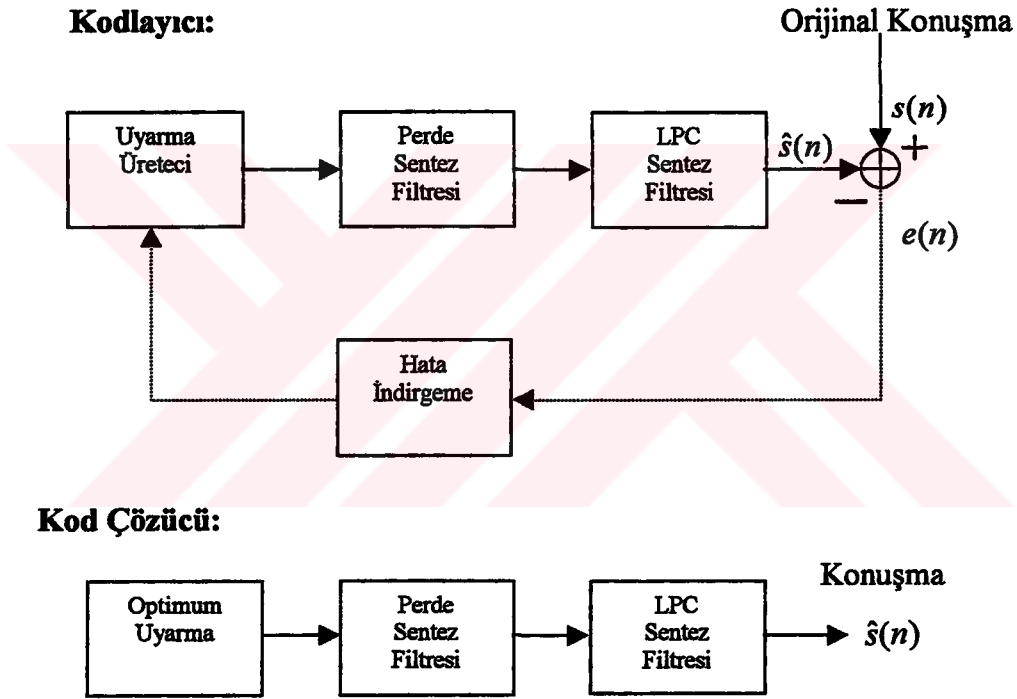
- Zamana göre değişen filtre
- Uyarma işareti
- Algılama (*perception*) esaslı minimizasyon yordamı

Orijinal işarete yaklaşabilmek için modelde sık sık parametre güncellemesi gerektiğinden sistemin analiz işlemi bloklar halinde yapılır, yani giriş konuşması uygun örnek bloklarına ayrılır. Analiz bloğunun veya çerçevenin uzunluğu ve güncelliği, kodlama yöntemlerinin bit hızını ve kapasitesini belirler. Bir AbS-LPC yapısının temel işleyişi aşağıdaki gibidir :

- 1) LPC ve perde filtresinin (zamana göre değişen filtre) başlangıç içeriği önceden belirlenen değerlere getirilir (genelde sıfır veya düşük seviyeli rastgele gürültüye ayarlanır).
- 2) Bir konuşma örnek çerçevesi tampon belleğe alınır ve konuşma çerçevesi üzerinde LPC analizi kullanılarak bir LPC katsayıları kümesi hesaplanır.
- 3) Hesaplanan LPC katsayıları kullanılarak nicemlenmemiş ilk artık değeri hesaplamak için bir ters LPC filtresi oluşturulur. Bu adım kapalı çevrim perde araması kullanılırsa gerekli değildir.
- 4) Uyarmayı belirlemek amacıyla etkin analiz için LPC çerçevesi genelde çok büyük olduğundan, bu çerçeve, tamsayılardan oluşan alt çerçevelere ayrılır.
- 5) Her bir alt çerçeve için :
 - a) Açık çevrim veya kapalı çevrim yöntemini kullanarak perde filtresi [uzun dönemli öngörü (LTP)] parametrelerini, yani gecikme ve ilgili ölçekleme faktörünü hesaplanır.
 - b) Perde filtresi parametreleri bulunduğunda, perde ve LPC sentez filtresi bir araya getirilerek peşpeşe (*cascade*) bir filtre oluşturulabilir. Bu peşpeşe filtre kullanılarak yapay konuşma ile orijinal konuşma arasındaki fark en aza inecek şekilde en iyi ikincil uyarma belirlenir.
- 6) Sonuçta oluşan yapay konuşma, optimum ikincil uyarma peşpeşe filtreden geçirilerek ve filtrelerin ilk bellek içerikleri (önceki alt çerçeve sentezinden kalan) tekrar oluşturularak, elde edilir.

7) Sonraki çerçeveler için adım 2-6 tekrar edilir.

Hem kodlayıcıda, hem de kod çözücüde yapay konuşmanın üretilmesi önemlidir. Bu, hem kodlayıcıda, hem de kod çözücüde filtre belleklerinde tekrar koşulları bulunacak şekilde, belleğin içeriğini güncellemek için gereklidir. Aksi takdirde, herşeyi eşzamanlı tutmak için belleğin içeriğinin kod çözücüye iletilmesi gerekir. Gerçekte, AbS-LPC ile ilgili en önemli nokta, iletim ortamı kusursuz olmadığında (örneğin, hata oranlarının çok yüksek olabileceği mobil bağlantılar) bu özdeş koşulun hem kodlayıcıda, hem de kod çözücüde nasıl korunacağıdır.



Şekil 6.2 AbS-LPC yönteminin blok yapısı

Yukarıdaki tariften, AbS-LPC yönteminin gerçekten senteze göre analiz olmadığı görülebilir. Bunun sebebi, yaklaşımın gerçekte sıralı olması, yani filtre parametrelerinin önce hesaplanıp sabit tutulması ve ardından da ikincil uyarmanın sentezleyerek analiz yöntemiyle hesaplanmasıdır. Sonuç olarak, orijinal referans işaretine göre optimum ikincil uyarma elde edilmesine rağmen, bu optimum

kullanılan filtreler ile sınırlıdır. Bu nedenle, ideal halde, bütün parametrelerin paralel optimizasyonu anlamına gelen en iyi uyarma ve filtre kombinasyonu istenir. Açıkça görüldüğü gibi, bu kombinasyonu bulma işlemi çok karmaşıktır ve çok hesap yükü gerektirir ve bu sebeple de yukarıda açıklanan adımlara ayrılır.

Bu modelin klasik “kaynak-filtre” tipi ses kodlayıcı modeline çok benzer olması ilginçtir [4]. Ancak, temel ses kodlayıcılar ile AbS-LPC kodlayıcılar arasında büyük bir fark vardır. Klasik ses kodlayıcılarda kaynak uyarması ünlü (darbe uyarması) ve ünsüz (rastgele gürültü uyarması) olarak sınıflandırılır ve bu da modelin hassas olmamasının ana sebeplerinden biridir. Ancak, AbS-LPC’de bu sınıflandırma kesin değildir; bu nedenle de uyarma işareti, darbe benzeri ile gürültü benzeri arasında herhangi bir işaret olabilir ve böylece çok daha iyi kalitede konuşma sentezlenmesine olanak sağlar.

6.2.1 Zamanla Değişen Filtreler

Modeldeki zamanla değişen filtreleri gösteren blok genelde iki lineer öngörücüden oluşur : LPC öngörücü veya kısa vadeli öngörücü (*Short Time Predictor-STP*) ve perde öngörücü veya uzun vadeli öngörücü (*Long Time Predictor-LTP*). Çoklu darbeleri LPC’de (MPLPC) ve düzgün darbe uyarmalı LPC’de (RPELPC), LTP, sadece uyarımayla perde darbeleri üretebileceğinden dikkate alınmayabilir. STP, konuşma işaretindeki kısa dönemli ilişkiyi modeller (spektrum zarfı) ve aşağıdaki gibi ifade edilebilir:

$$\frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^p a_i z^{-i}} \quad (6.1)$$

Burada, a_i , STP katsayıları, p ise filtrenin mertebesidir. STP, esasında temel LPC analiz filtresidir. Konuşma spektrumundaki değişimi yansıtmak için zaman göre değişir ve tipik uyum hızı yaklaşık 20-30 msan’dır. Filtrenin mertebesi (p) genelde 8-12 arasında seçilir.

LTP filtre, konuşmadaki uzun dönemli ilişkiyi modeller (hassas spektrum yapısı) ve aşağıdaki gibi ifade edilebilir :

$$\frac{1}{P(z)} = \frac{1}{1 - \sum_{i=-1}^I b_i z^{-(D+i)}} \quad (6.2)$$

Burada, D , uzun dönemli ilişkiyi gösterir ve genelde perde periyoduna veya onun katlarına karşılık gelir, b_i ise LTP kazanç katsayılarıdır. Yine, bu filtre de zamanla değişir ve genelde STP'den daha yüksek uyum hızlarına sahiptir (örneğin, 5-10 msan). Filtre ara-uç (*tap*) sayısı tipik olarak $I=0$ (yani 1 ara-uçlu) ve $I=1$ (yani 3 ara-uçlu) şeklinde seçilir. Filtreler Şekil 6.2'de gösterilen sırayla bağlanmasına rağmen birbiriyle değiştirilebilir ve bazı hallerde, sadeleştirilmiş MPLPC yapısında olduğu gibi LTP tamamıyla kullanılmaz. İki filtrenin özyinelemeli niteliğinden dolayı her ikisinin de bellekleri, önceki analiz çerçevesinden nakledilen çalışma tampon bellekleri içinde bulunur. Bu filtre belleğinin korunması ve AbS analizine dahil edilmesi çok önemlidir, çünkü analizin geçmişini, yani ilişki sürelerini yansıtır ve önceki çerçevelerde ortaya çıkan hataları içerir. Ayrıca, blok yönlendirmeli analizin sebep olduğu distorsiyonlar için bir yumuşatma etkisi oluşturur.

6.2.2 Algılama Esaslı Minimizasyon İşlemi

Şekil 6.2'de gösterilen AbS-LPC kodlayıcı, uyarma işareti ile STP ve LTP filtrelerini değiştirerek uygun bir hata kıstasına göre $s(n)$ orijinal işaret ile $\hat{s}(n)$ sentezlenmiş işaret arasındaki hatayı en aza indirir. Daha önce tarif edildiği gibi, bu minimizasyon bir sıralı yaklaşımla elde edilir. Önce zamana göre değişen filtre parametreleri belirlenir ve sonra bunlar sabit tutularak uyarma optimize edilir.

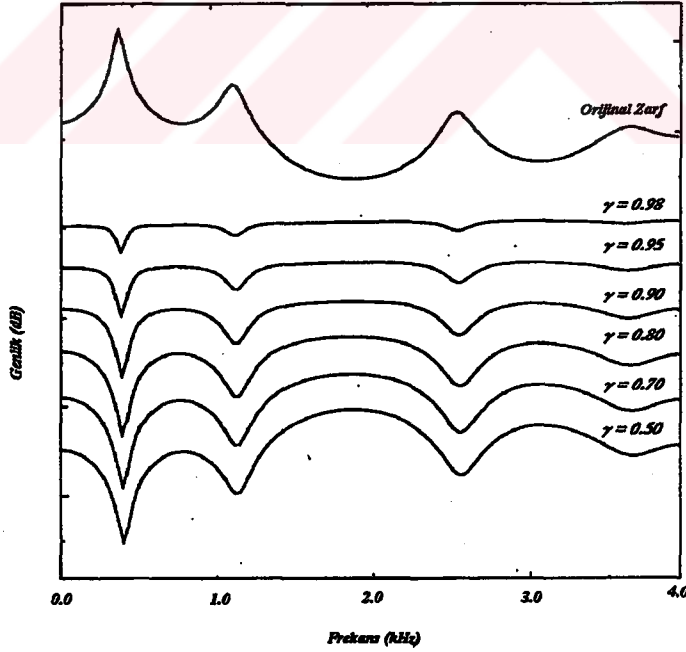
Her iki yaklaşım için kullanılan optimizasyon kıstası, genelde kullanılan ortalama karesel hatadır ve basitlik ile uygun performans sağlar. Ancak, düşük bit hızlarında örnek başına sadece bir veya daha az bit vardır ve bu sebeple, örneğin, örnek başına 1'den fazla bit bulunan 16 kbit/san yapılarında dalga biçimini yakın şekilde eşlemek

zordur. Sonuç olarak, orijinal işaret ile yeniden oluşturulan işaret arasındaki ortalama karesel hata daha az önemlidir ve yeterli değerin altındadır. Gerekli olan, insan algılama kıstasına daha yakın bir hata kıstasıdır. Ses algılamasıyla ilgili pek çok çalışma yürütülmekte olmasına rağmen şu ana kadar tatmin edici bir hata kıstası belirlenmemiştir. Ancak, AbS-LPC yordamlarında ağırlıklandırma filtresi kullanımı popüler, fakat tam tatmin edici olmayan bir yöntemdir. Bu filtre aşağıdaki formülle ifade edilir :

$$W(z) = \frac{A(z)}{A(z/\gamma)} \quad (6.3)$$

$$= \frac{1 - \sum_{i=1}^p a_i z^{-i}}{1 - \sum_{i=1}^p a_i \gamma^i z^{-i}} \quad 0 \leq \gamma \leq 1$$

Bu ağırlıklandırma filtresi, *Atal*'ın APC için teklif ettiği filtrenin aynısıdır ve frekans cevabı Şekil 6.3'de gösterilmiştir.



Şekil 6.3 Tipik ağırlıklandırma filtresi spektrumları ve orijinal konuşma zarfıyla karşılaştırılması

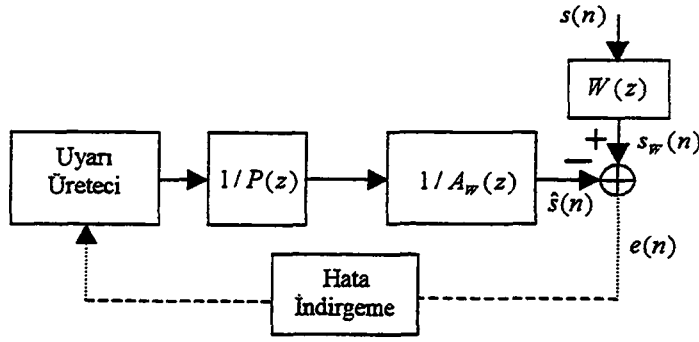
γ çarpanının etkisi merkez rezonans frekanslarını değiştirmez, fakat rezonansların bant genişliğini, Δf kadar genişletir :

$$\Delta f = -\frac{f_s}{\pi} \ln \gamma \text{ (Hz)} \quad (6.4)$$

Burada, f_s , Hz cinsinden örnekleme frekansıdır. Şekil 6.3'den görülebileceği gibi, ağırlıklandırma filtresi, LPC analiziyle belirlenen rezonanslara karşılık gelen frekans bölgelerinde yumuşatma yapar. Rezonans bölgelerinde daha büyük distorsiyona müsaade edilerek rezonans sıfırlarında daha rahatsız edici olan gürültü azaltılabilir. Son dengeleme miktarı γ çarpanıyla kontrol edilir ve bu çarpan 0 ile 1 arasında bir genişletme etkisi yaratır. En uygun γ değeri dinleme yoluyla subjektif olarak seçilir, 8 kHz örnekleme için γ genelde 0,8-0,9 civarında bulunmaktadır.

Ağırlıklandırma filtresi normal durumda ($s(n)$ - $\hat{s}(n)$ işleminden sonra) olduğu gibi kullanılabilmesine rağmen, Şekil 6.4'de gösterildiği gibi çıkarma işlemine katkıda bulunan iki dala ayrılarak daha elverişli bir hesap yöntemiyle değiştirilebilir. Bu, bir giriş örnekleri bloğunda AbS aramasından önce sadece bir kez ağırlıklandırma yapılmasını sağlar. Aynı zamanda, $W(z)$ ile STP filtresi bir araya getirilerek değiştirilmiş tam kutuplu sentez filtresi oluşturulur:

$$\begin{aligned} \frac{1}{Aw(z)} &= \frac{1}{A(z)} \cdot W(z) \\ &= \frac{1}{1 - \sum_{i=1}^p a_i \gamma^i z^{-i}} \end{aligned} \quad (6.5)$$



Şekil 6.4 AbS-LPC kodlama yapısının değiştirilmiş kısmı (ağırlıklandırma filtresi, hata minimizasyon yordamının iki koluna taşınmış)

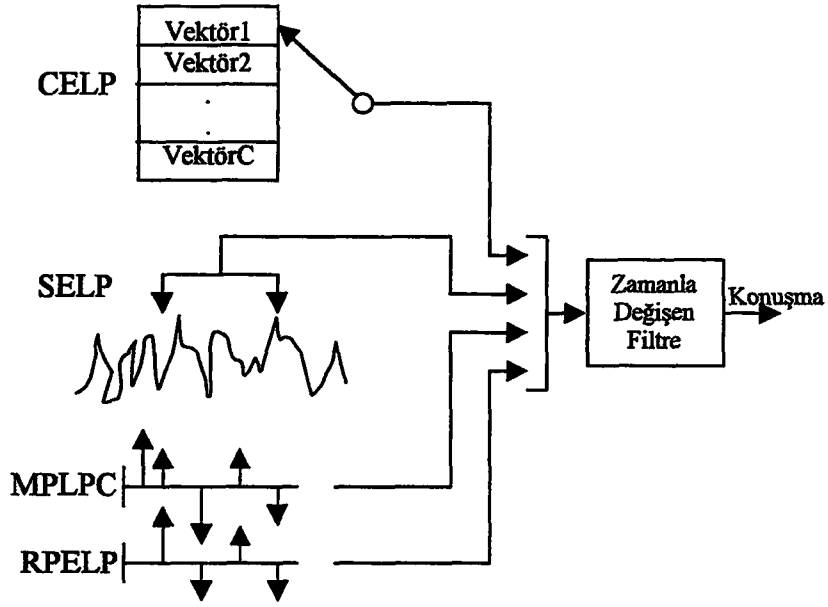
6.2.3 Uyarma İşareti

$x(n)$ işareti, AbS-LPC modeline uyarma girişini ifade eder ve bu nedenle, belki de, Şekil 6.2’de gösterilen modelin en önemli bileşenidir. Bunun sebebi, kaynakta, zamanla değişen filtrelerin spektral modeliyle ifade edilmeyen artık yapılar bulunmasıdır. Bunlar arasında, LTP kapsamında bulunmayan önemli ilişki (örneğin, LTP gecikme aralığından çok daha büyük ilişkiler) sergileyen perdeye veya uzun döneme bağlı yapılar yer alır. Ayrıca, belirleyici yöntemlerle etkin şekilde modellenemeyen rastgele yapıları da içerir. LTP filtre belleği ölçekli $x(n)$ modellerinden oluştuğu için uygun bir uyarma modeli çok önem taşır. Uyarma modeli saf darbeler ile rastgele gürültü arasında değişebileceği için, bloğun tamamının analizi (6.6) eşitliği ile ifade edilebilir.

$$U_i = g_i X_i \quad (6.6)$$

Burada, U_i , L boyutlu i ’inci uyarma vektörü, X_i , M adet L boyutlu “şekil” vektörleri, g_i ise X_i şekliyle ilgili M boyutlu kazanç veya ölçek vektörüdür.

$x(n)$ ’e ait temel istatistiksel bilgileri yansıtan çeşitli şekil vektörleri uygun şekilde seçilerek uyarma herhangi bir biçimde olabilir. Kullanılan şekiller arasında Çoklu Darbeyle, Düzgün Darbeyle, Kod Kitabıyla ve Kendinden Uyarılan modeller yer almaktadır. Değişik uyarma tipleri için AbS-LPC blok şeması Şekil 6.5’de gösterilmiştir.



Şekil 6.5 AbS-LPC kodlayıcının değişik uyarma tiplerini gösteren genel blok yapısı

6.2.3.1 Kod Kitabıyla Uyarma

Kod kitabıyla uyarmada da (CELP) [38] uyarma vektörü, ilgili bir ölçekleme veya kazanç vektörüne sahip muhtemel C stokastik diziden oluşan, önceden belirlenen bir grup içinden seçilir. Ölçekleme vektörü genelde sadece skalar bir ölçek faktörü olmasına rağmen vektörün çeşitli parçalarındaki uyarma vektör elemanlarını daha hassas şekilde ölçeklemek için birden fazla da olabilir. Bu sebeple, genel geçerlilik için, c^k kod kitabı vektörü, her biri L/M uzunlukta M eşit parçaya bölünebilir ve her bir k indisi ($1 \leq k \leq C$) için optimum kazançlar ($g_{k1}, \dots, g_{kM}$) hesaplanabilir. Böylelikle kod kitabıyla uyarma için U uyarması aşağıdaki gibi yazılabilir :

$$U_k = g_k X_k \quad , \quad k = 1, \dots, C \quad (6.7)$$

Burada,

$$g_k = [g_{k1}^i, \dots, g_{kM}^i] \quad , \quad i = 1, \dots, L/M \quad (6.8)$$

$$x_k = \begin{cases} c_j^k & , \quad i = 1, 2, \dots, L-1 \\ 0 & , \quad \text{aksi halde} \end{cases} \quad (6.9)$$

olarak ifade edilebilir. AbS yordamında muhtemel C adet c dizisi sistematik olarak $H_c(z)$ sentez filtresinden geçirilir ve en düşük hatayı üreten vektör istenen dizi olur. Dizi kümesi hem kodlayıcıda, hem de kod çözücünde bulunduğundan, kod kitabına ait sadece k indisinin iletilmesi gerekir. Bu nedenle, örnek başına 1'den az bit ile kodlama mümkün olabilmektedir.

Kod kitabı sonlu boyutta olduğundan, kodlanacak uyarmayı iyi şekilde temsil eden vektörler kitaba yerleştirilmelidir. *Atal*'ın yaklaşımında, birim varyanslı beyaz *Gauss* şeklinde rastgele sayılar kullanılmış ve bu seçimin çok iyi sonuçlar verdiği rapor edilmiştir. Sonuçlar konuşmanın hem STP, hem de LTP filtrelerinden geçirilerek ters filtrelemesiyle elde edilen tahmin hata örneklerinin olasılık yoğunluk fonksiyonunun *Gauss* yöntemine çok yakın olmasıyla kısmen açıklanabilir.

6.2.3.2 Kendinden Uyarma

Kendinden uyarmada (SELP) [39], $x(n)$ işareti LTP'ye benzer bir yapı kullanılarak kodlu uyarma fonksiyonunun geçmişinden türetilir. SELP'de birden fazla SE-LTP (kendinden uyarmalı LTP) kullanılabilmesine rağmen burada sadece bir SE-LTP dikkate alınmıştır. SE-LTP, SE-LTP belleğinin bazı rastgele içeriklerle doldurulmasıyla başlatılır. Sonra her sentezleyerek analiz yordamında L blok uzunluğuna eşit ve zaman içinde k ile indislenmiş bir dizi seçilir ve $H_c(z)$ 'den geçirilir. En iyi k_{opt} indisli vektör, fark hatasını en aza indiren vektördür. En iyi vektör bulunduğu ve mevcut bloğu sentezlemede kullanıldığında, atılan en eski L örnekler SE-LTP'ye geri beslenir. SE-LTP, adaptif kod kitabına sahip bir CELP, kodlayıcıdır ve benzer şekilde formüle edilebilir. Ancak, SELP'de, muhtemel C

dizileri kod kitabı girdileri değil, sadece daha önce bulunan U_{opt} optimum uyarmalarını (doğru ölçekli önceki uyarma vektörleri) içeren aşağıda ifadesi verilen B 'nin pencerelemiş şeklidir.

$$B = \left[{}^U(-C-L), {}^U(-C-L+1), \dots, {}^U(-1) \right] \quad (6.10)$$

$$x^k(n) = B(n-L-k), \quad 0 \leq n < L, \quad k = 1, \dots, C \quad (6.11)$$

6.2.3.3 Çoklu Darbeli ve Düzenli Uyarma

Çok Darbeli LPC (MPLPC) [36, 40], AbS-LPC kodlama yapılarının ilk uygulamalarındandır. MPLPC'de uyarma işaretinin niteliğiyle ilgili olarak önceden kabul yapılmayarak uyarmanın ünlü ve ünsüz sınıflarına ayrılmasından kaçınılır. MPLPC'de, uyarma, CELP'de olduğu gibi bir kod kitabından ya da SELP'de olduğu gibi belirli bir analiz çerçevesindeki bir sabit diziler kümesinden optimum bir dizi seçiminden ziyade, eşit olmayan aralıklarda değişik genliklere sahip küçük bir darbe dizisiyle belirlenir. Kodlamada, minimum hata üreten darbe pozisyonlarının ve uyarma genliklerinin belirlenmesi yapılır. Gerekli ön bilgi sadece analiz bloğu başına ihtiyaç duyulan darbe sayısıdır. İyi kaliteli konuşma sentezi için tipik darbe hızı 8 örnek başına 1 darbe civarındadır. MPLPC, verilen M -darbeli n_i darbe yeri kümesi için $M \times L$ boyutlu bir X_M matrisi ile ifade edilebilir.

$$U_M = g_M X_M \quad (6.12)$$

Burada,

$$g_M = [g_{n_1}, \dots, g_{n_M}] \quad (6.13)$$

$$x_M = \begin{cases} 1, & j = n_{i+1}, \quad 0 \leq i \leq M-1 \\ 0, & \text{aksi halde}, \quad 0 \leq j \leq L \end{cases} \quad (6.14)$$

olarak yazılabilir. Yukarıdaki formülden, MPLPC, büyüklüğü darbe sayısı ile ve darbe genliğini hesaplamada kullanılan bit sayısı ile belirlenen çok büyük kod kitabına sahip bir CELP sistemi olarak yorumlanabilir.

MPLPC ifadesi, darbe sayısının önceden belirlenmesi zorunluluğu haricinde darbelerin aralıkları veya dağılımı üzerinde herhangi bir sınırlama getirmez. Bu düzen, görüldüğü gibi, darbe pozisyonlarının kodlanması için çok sayıda bit gerektirir. Bu nedenle, hem bit tasarrufu, hem de optimum pozisyon belirleme karmaşıklığı açısından daha iyi yapı bir darbe atama istenir. Seyrek dağılımlı kod kitaplı CELP, darbelerin hem pozisyonu, hem de genliği üzerinde ciddi sınırlamalara sahip bir MPLPC'dir. Bunun çok kuvvetli bir temel oluşturduğu açıkça görülmektedir, fakat yapı sadece darbe pozisyonu üzerinde etkili olursa genlikler değişebilir. Bu, Şekil 6.6'da görüldüğü gibi, normal darbe ikazlı LPC'nin (RPELPC) yapısıdır. Bir RPELPC uyarma çerçevesinde darbeler N eşit aralıkları ile yer alır ve pozisyonları tamamen birinci darbenin pozisyonuyla belirlenir. Bu, uyarma çerçevesi büyüklüğünün L 'nin 40 ve N 'nin de 4 olduğu Şekil 6.6'da gösterilmiştir. Çerçeve başına toplam darbe sayısı $M=L/N$ dir ve RPELPC aşağıdaki gibi formüle edilebilir :

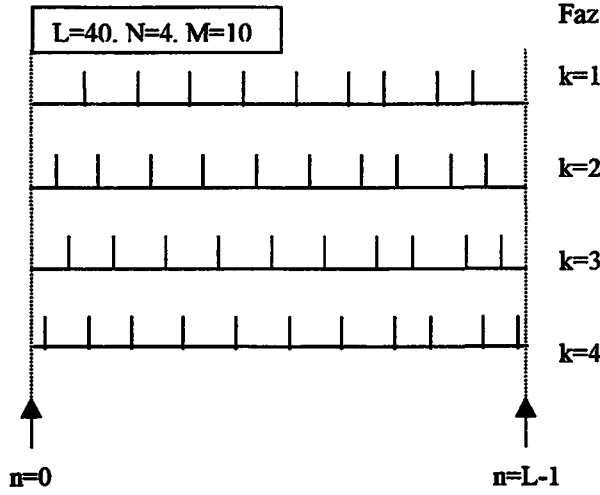
$$U_k = g_k x_k, \quad k = 1, 2, \dots, N \quad (6.15)$$

Burada,

$$g_k = [g_{k1}, \dots, g_{kM}] \quad (6.16)$$

$$x_M = \begin{cases} 1, & j = iN + k - 1, \quad 0 \leq i \leq M - 1 \\ 0, & \text{aksi halde, } 0 \leq j \leq L \end{cases} \quad (6.17)$$

ifadeleri yazılabilir. Böylelikle RPELPC'de, $H_c(z)$ 'den N dizileri geçirilir ve hatayı en aza indiren dizi en iyi dizi olarak seçilir.



Şekil 6.6 Tipik bir RPELPC darbe pozisyonu yapısı

6.2.4 En Uygun Uyarma Dizisinin Tesbiti

Önceki bölümde değişik uyarma işareti biçimleri tarif edilmiş, bunların belirlenmesine ilişkin detaylara girilmemiştir. Yukarıda sözü edilen kaynak kodlama teknikleri, artık yapı tiplerini en iyi şekilde modellemekle birlikte her biri için optimum uyarma dizisini belirlemede kullanılan formül aynıdır [35]. Tek fark, bir optimum uyarma biçiminin seçilebileceği fonksiyon aralığıdır. CELP’de, kod kitabında bulunan L noktalı *Gauss* rastgele dizilerine karşılık gelen örnek rastgele fonksiyonlar aralığı taranır. SELP, uyarma işaretinin geçmişiyle SE-LTP gecikmeleri aralığında zaman içinde tarama gerektirir. MPLPC ve RPELPC’de tarama yine zaman içinde ve fakat bir dizi gecikmeli dürtü (*impulse*) cevabı fonksiyonuyla yapılır. Söz konusu teknikler için optimum uyarma fonksiyonunu bulma kistası aynıdır. Amaç, gX ’in Şekil 6.4’de görülen $e(n)$ ağırlıklandırılmış hatayı en aza indiren yapay bir işaret üretmesi için, X şekil matrisini ve ilgili g kazancını belirlemektir.

$$e_k = s_w - \hat{s}_k \quad (6.18)$$

Burada, s_w , ağırlıklandırılmış orijinal referans işaret, s_k , sentezlenmiş işaret, k ise ilgili uyardır. H , $L \times L$ boyutunda j ’inci sırasında LTP’nin (kırpılmış) birleşik dürtü

cevabı $h(n)$ ve birim dürtü $\delta(n-j)$ 'in oluşturduğu değiştirilmiş STP bulunan bir matris olsun.

$$H = \begin{bmatrix} h(0)h(1)...h(L-1) \\ 0 h(0)...h(L-2) \\ \dots \dots \dots \\ 0 0 \dots h(0) \end{bmatrix} \quad (6.19)$$

s_m , sıfır girişli kaskad LTP ve STP filtrelerinin çıkışını (daha önce sentezlenen çerçevelerden kalan bellek) gösterirse, eşlenecek $\tilde{s}$ referans işareti aşağıdaki gibi tanımlanabilir:

$$\tilde{s} = s_w - s_m \quad (6.20)$$

$$\Rightarrow e_k = \tilde{s} - U_k H \quad (6.21)$$

$$= \tilde{s} - g_k X_k H \quad (6.22)$$

$$= \tilde{s} - g_k \hat{s}_k \quad (6.23)$$

Burada,

$$\hat{s}_k = X_k H \quad (6.24)$$

Burada ölçüt hatanın karesinin minimum olmasıdır; bu sebeple amaç E_k 'yi en aza indirmektir. Burada,

$$E_k = e_k e_k^T \quad (6.25)$$

T ise yer deęiřtirmeyi gösterir. k 'inci aday uyarma için g_k optimum genlik vektörü, e_k hatasının s_k tahminine ortogonal olması kořuluyla (6.23) ve (6.25) eřitliklerinden hesaplanabilir.

$$e_k \hat{s}_k^T = 0 \quad (6.26)$$

Bu nedenle,

$$(\tilde{s} - g_k \hat{s}_k) \hat{s}_k^T = 0 \quad (6.27)$$

$$\Rightarrow g_k = \tilde{s} \hat{s}_k^T [\hat{s}_k \hat{s}_k^T]^{-1} \quad (6.28)$$

(6.28) eřitlięini (6.23) eřitlięinde yerine koyduęumuzda (6.25) eřitlięi ařaęıdaki gibi yazılabilir:

$$E_k = \tilde{s} \left[I - \hat{s}_k^T [\hat{s}_k \hat{s}_k^T]^{-1} \hat{s}_k \right] \tilde{s}^T \quad (6.29)$$

Burada, I özdeř matristir. Böylece, tüm k 'larda minimum E_k deęerini veren g_k vektörü ve X_k matrisi en uygun uyarma olarak seęilir.

Yukarıdaki E_k ifadesi bütün mümkün uyarma biçimleri için genelleřtirilir ve bu sebeple uygulamada gerekenden daha karmařıktır. Örneęin, $M=1$ olan bir CELP kodlayıcı ele alındıęı takdirde yukarıda gösterildięi gibi, $[\hat{s}_k \hat{s}_k^T]^{-1}$ dönüşümü gereksizdir.

$$\tilde{s}_k \tilde{s}_k^T = \sigma(\text{skalar}) \quad (6.30)$$

$$g_k = \frac{\tilde{s} \hat{s}_k^T}{\sigma} = g_k(\text{skalar}) \quad (6.31)$$

$$\Rightarrow E_k = \tilde{s} \tilde{s}^T - g_k \hat{s}_k \tilde{s}^T \quad (6.32)$$

$$= \tilde{s}\tilde{s}^T - Q_k \quad (6.33)$$

(6.31) ve (6.32) nolu eşitlikleri zaman bölgesindeki örnekler biçiminde yeniden yazılırsa

$$g_k = \frac{\sum_{i=0}^{L-1} \tilde{s}(i) \hat{s}_{k(i)}}{\sum_{i=0}^{L-1} \hat{s}_k^2(i)} \quad (6.34)$$

$$E_k = \sum_{i=0}^{L-1} \tilde{s}^2(i) - g_k \sum_{i=0}^{L-1} \hat{s}_k(i) \tilde{s}(i) \quad (6.35)$$

g_k (6.35)'de yerine koyulursa, (6.33) aşağıdaki gibi yeniden yazılabilir:

$$E_k = \sum_{i=0}^{L-1} \hat{s}_k^{-2}(i) - \frac{\left[\sum_{i=0}^{L-1} \hat{s}_k(i) \tilde{s}(i) \right]^2}{\sum_{i=0}^{L-1} \hat{s}_k^2(i)} \quad (6.36)$$

ve Q_k ise aşağıdaki formülle ifade edilebilir:

$$Q_k = \frac{\left[\sum_{i=0}^{L-1} \hat{s}_k(i) \tilde{s}(i) \right]^2}{\sum_{i=0}^{L-1} \hat{s}_k^2(i)} \quad (6.37)$$

g_k skalar faktörü; basitçe ağırlıklandırılmış konuşma ile uyarma cevabının çapraz ilişkisinin, uyarma cevabının karesel toplamına bölümüdür. Karesel hataların toplamı E_k ise ağırlıklandırılmış konuşmanın enerjisi ile uyarma cevabı ve ağırlıklandırılmış konuşma işaretinin karesel çapraz ilişkisi arasındaki farktır. Pratikte, en iyi uyarma seçimi için maksimum Q_k bulunmaktadır.

MPLPC’de uyarma şeklini belirlemek için kullanılan yukarıdaki yöntem pratik değildir, çünkü bütün muhtemel darbe yeri kombinasyonlarının taranmasını gerektirir (örneğin, $L=40$, $M=4$, darbe pozisyon vektörü sayısı = 91390). Bu nedenle, hesaplamayı sadeleştirmek için çeşitli yaklaşımlar kullanılır. Basit ve popüler bir yöntem de her seferinde bir darbe yeri belirlenmesidir. Bu darbeler için en uygun yer, verilen bir L aralığında bütün muhtemel darbe yerleri için hata hesaplanarak ve minimum hatanın yeri belirlenerek bulunur (yani, en iyi k pozisyonunu belirlemek için $M=1$ alınır ve (6.33) eşitliği kullanılır). Pozisyon bilindiğinde k pozisyonundaki bu darbenin katkısı referans işaretinden çıkarılır ((6.20)’ye benzer) ve bütün darbeler bulunana kadar bu yaklaşım tekrarlanır. Bu yöntem, aşağıdaki gibi özetlenebilir:

- 1) $s_0 = s_w - s_m$ kabul edilsin ve H_k ise, s_m ’nin STP filtre belleği cevabı olduğu k ’ncı darbe ile ölçeklenmiş STP dürtü cevabı olsun.
- 2) $i = 1, \dots, M$ için her seferinde
 - i) $k=1, \dots, L$ için $E_i^k = \min \{ \tilde{s}_i \tilde{s}^T - g_k H_k \tilde{s}_i^T \}$ bulunur,
 - ii) $\tilde{s}_{i+1} = \tilde{s}_i - g_{i,opt} H_i$
- 3) Son olarak, $\hat{s}_{mp} = \sum_{i=1}^M g_{i,opt} H_i$ bulunur

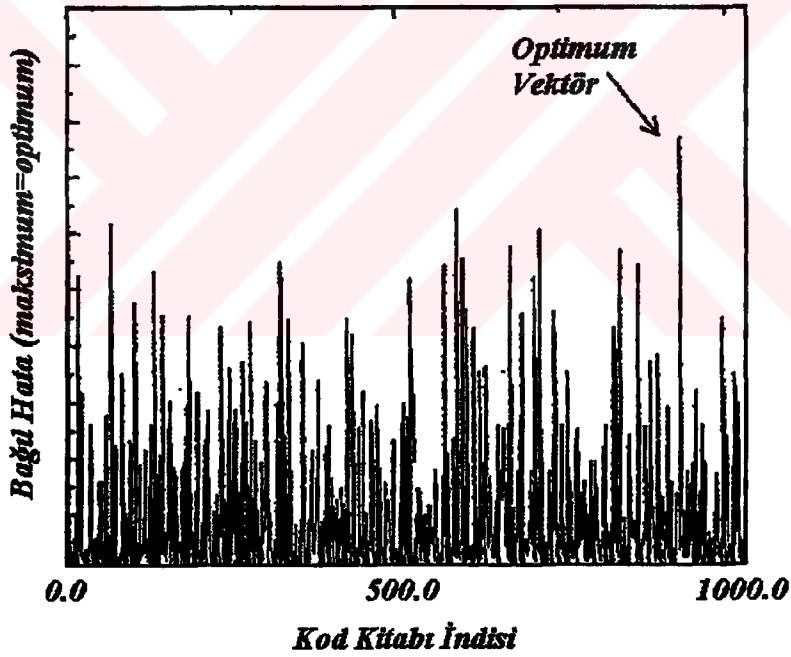
Yukarıda bahsedilen, her seferinde bir darbe bulunması yaklaşımının optimal çözüme oldukça yakın olduğu ortadadır ve daha fazla iyileştirme getirmeye çalışan yordamlar uygulandığı sıkça rapor edilmektedir. Örneğin, popüler bir son işleme (*post-processing*) yöntemi, bir $M \times M$ matris dönüşümü yaparak darbe pozisyonları bulunduktan sonra darbe genliklerinin yeniden optimize edilmesidir. RPELPC’de darbelerin pozisyonu sabittir, bu nedenle de her seferinde bir darbe belirleme gereksinimi yoktur. RPELPC’de, genlik vektörü, N kez (muhtemel genlik dizisi sayısı) tekrarlanan tek basamakta birlikte optimize edilir ve ortalama karesel hatayı en aza indiren vektör seçilir.

6.2.5 AbS-LPC Yapılarının Özellikleri

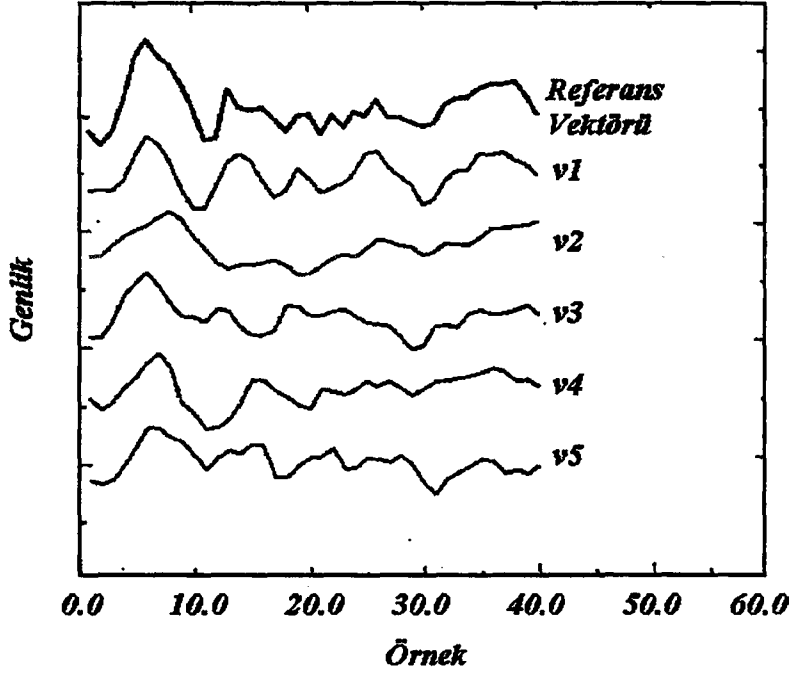
İki ana AbS-LPC yaklaşımı olan MPLPC ve CELP'yi daha detaylı olarak incelemeden önce değişik AbS-LPC yapılarının uygulandığı yöntemlerdeki bazı benzerlik ve farklılıkları vurgulamak yerinde olur. Bu benzerlik ve farklılıklar esas olarak aşağıda tarif edilen uyarma davranışı özelliklerinde bulunmaktadır.

6.2.5.1 CELP

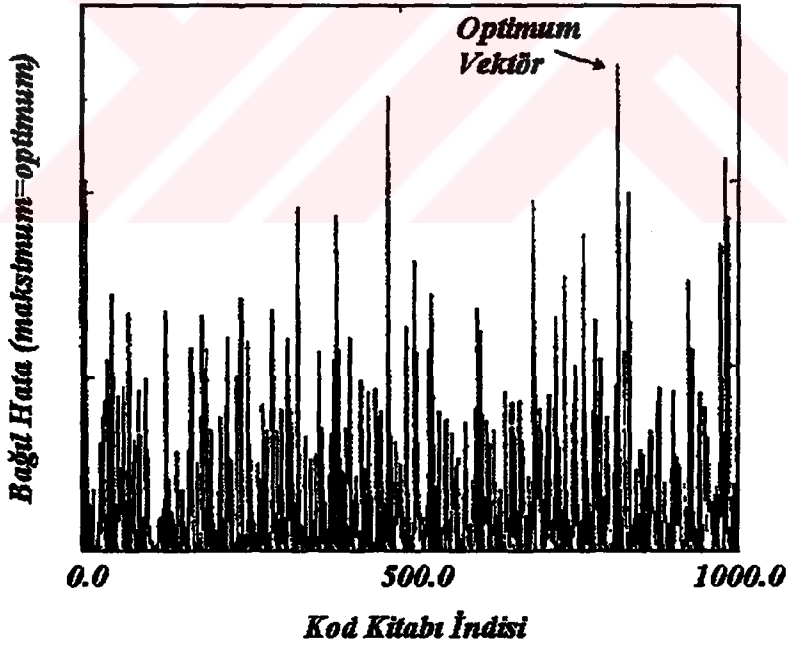
CELP sisteminde amaç kod kitabından orijinal referans vektörüne en iyi uyan vektörü seçmektir. İki ardışık vektör aramasının tipik grafikleri Şekil 6.7, 6.8, 6.9 ve 6.10'da gösterilmiştir.



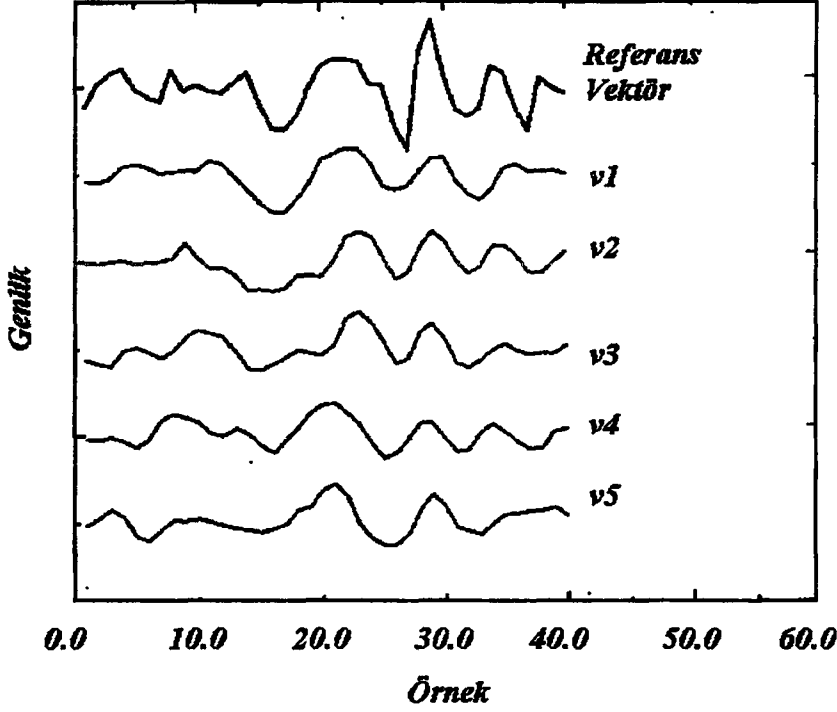
Şekil 6.7 CELP için kod kitabı indisine ilişkin hata grafiği (ayırddedici vektör)



Şekil 6.8 Orijinal ile karşılaştırılmış sentezlenmiş en iyi kod kitabı girişleri
(ayırmedici vektör)



Şekil 6.9 CELP için kod kitabı indisine ilişkin hata grafiği
(daha az ayırmedici vektör)



Şekil 6.10 Orijinal ile karşılaştırılmış sentezlenmiş en iyi kod kitabı girişleri (daha az ayırtedici vektör)

Bu grafiklerde çıkış hatası (gerçekte, her bir kod kitabı vektörü için Q_k) gösterilmektedir. Birinci vektörden görülebileceği gibi kod kitabındaki en uygun $v1$ vektörü diğer vektörlerden oldukça farklıdır. Ancak, bir sonraki kod kitabı araması için hatanın daha düşük düzeyde farklılık gösterdiği bulunmuş ve birinci, ikinci ve üçüncü en uygun vektörlerdeki benzerlikle gösterilmiştir. Bu gösterim, ortalama karesel hatanın AbS-LPC kodlama yapılarında (gerçekte, genelde konuşma kodlamasında) kıstas seçimi olarak kullanılmasının uygunsuzluğunu açıkça ortaya koymaktadır; yani, en uygun aday seçimi süreci hiçbir şekilde kolaylıkla kontrol edilemez. Birinci referans vektöründe $v1$ seçimi subjektif olarak da doğru olmasına rağmen, ikinci referans vektörü için $v1$ seçimi bunun kadar kesin değildir. İkinci en iyi vektör seçilmiş olduğunda subjektif farkın ne olacağını incelemek ilginçtir. Bu denendiği takdirde işlenen konuşmanın kalitesinde beklendiği gibi önemli bir bozulma olmadığı görülebilir. Bu, birinci ve ikinci en iyi seçilen vektörler arasında hem objektif, hem de subjektif açıdan belirgin bir farklılık olacak şekilde kod kitabı vektörlerinin daha iyi optimize edilip edilemeyeceği sorusunu gündeme

getirmektedir. Maalesef bilinen kod kitapları bu konuda hiç başarılı olamamıştır. Bunlara, çoklu-darbe özellik kod kitapları, ses telleri arası (*glottal*) darbe kod kitapları, v.b. dahildir.

6.2.5.2 SELP

SELP kodlaması için o andaki en iyi uyarma geçmiş uyarmalardan elde edilir. Anlatılan durumda ikincil uzun süreli öngörücüler (LTP) 1 olarak saptanmıştır. İkincil LTP ile (LTP2), geçmiş uyarı hafıza LTP'sinde daha kısa olan birincil LTP tarafından modellenmemiş, uzun süreli herhangi bir ek ortak ilişkiyi modellemeye çalışılmaktadır. Şekil 6.11'de bir kadın sesi için LTP1 ve LTP2 zaman gecikme indisi dağılım grafiği gösterilmektedir. LTP1'in histogramından, test için kullanılan konuşmacının 42 örnek civarında bir perde tepe noktası olduğu açıktır. Ancak, LTP2 için grafikte benzer çok az yapı görülmektedir. Test konuşmasında fazladan uzun süreli yapı bulunmuşsa SELP'in bunu uygun biçimde modellemediği görülmektedir.

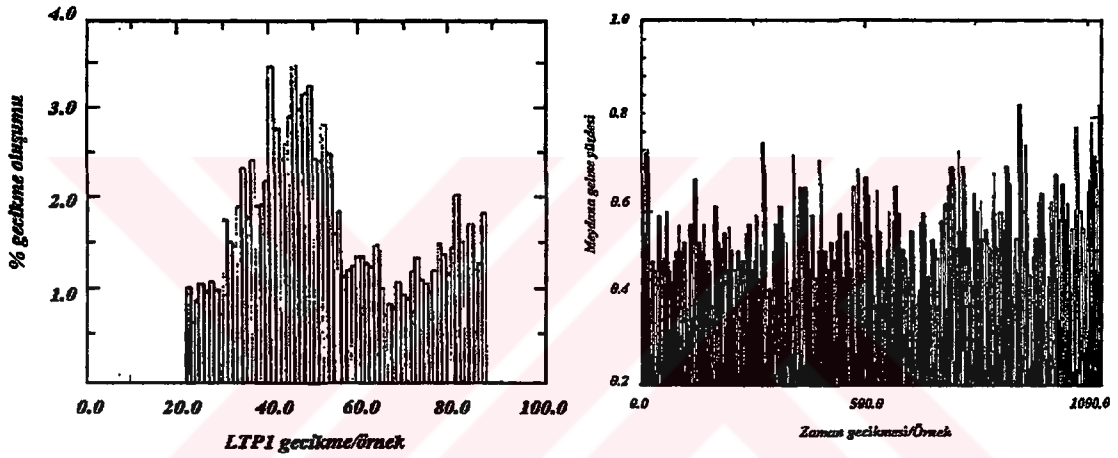
6.2.5.3 MPLPC

MPLPC'de, çıkıştaki hatayı en aza indiren rastgele dizilmiş darbelerin en iyi kombinasyonu seçilir. Darbeleri elde etmek için genellikle üç tür arama metodu uygulanır:

1. Yöntem : j kademesinden $j-1$ kademesine kadar olan tüm darbe genlik ve konumlarının bilindiği varsayılır ve yalnız n_j ve g_j genlikleri hesaplanır.
2. Yöntem : Her j kademesinde yalnız $n_1, n_2, \dots, n_{j-1}$ konumlarının sabit kalıp g_{j-1} 'e kadarki darbe genliklerinin optimize edilmesi dışında 1. yöntemde kullanılan yöntemlerin aynıları kullanılır. Tüm darbe genlikleri, daha önceki darbelerdeki hatalardan dolayı oluşan duyarsızlıkların kompensasyonunda kullanılmak üzere değiştirilebileceği için, darbeler yakın aralıklı olsalar dahi duyarlılık sağlanabilir.

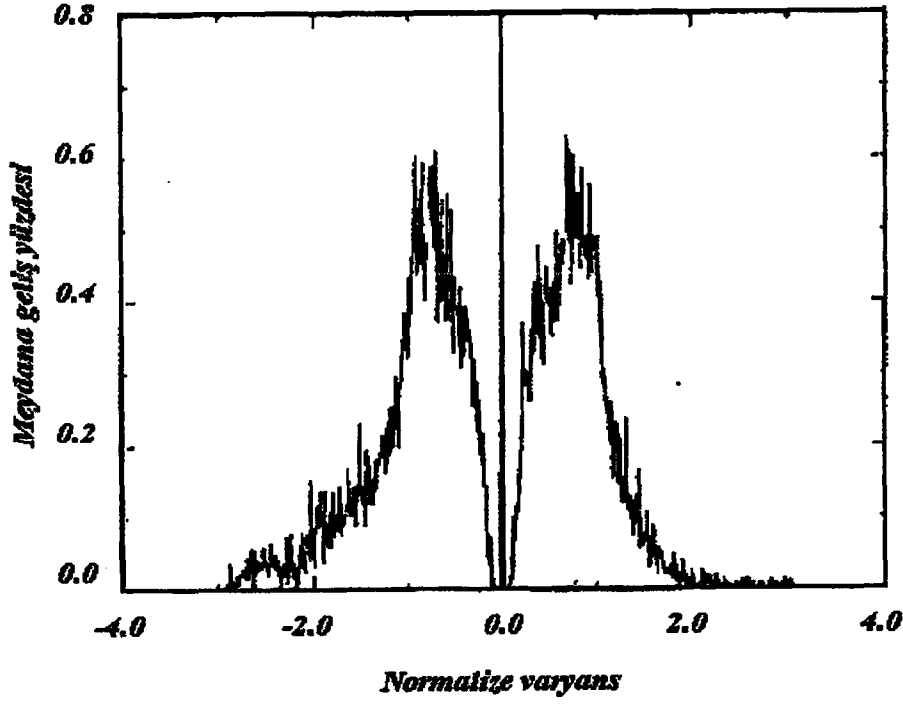
3. Yöntem : Darbe konumları sabit kalırken tüm $g_1, g_2, \dots, g_m$ genliklerinin yalnızca son aşamanın ardından tekrar optimize edilmesi dışında 2. yöntem gibidir.

Beklendiği gibi 2. yöntem, analiz edilen bloğun boyu ve darbe sayısında bağlı olarak hem objektif hem subjektif en iyi performansı sergiler. Ancak buradaki kazanç nicelemlere uygulandığında azalır. Bu nedenle, karmaşıklık açısından bakıldığında 3. yöntem tercih edilir, zira performans yönünden mukayese edildiğinde, devrenin yalnız bir kere tekrar optimize edilmesi hariç 2. yönteme çok benzemektedir.

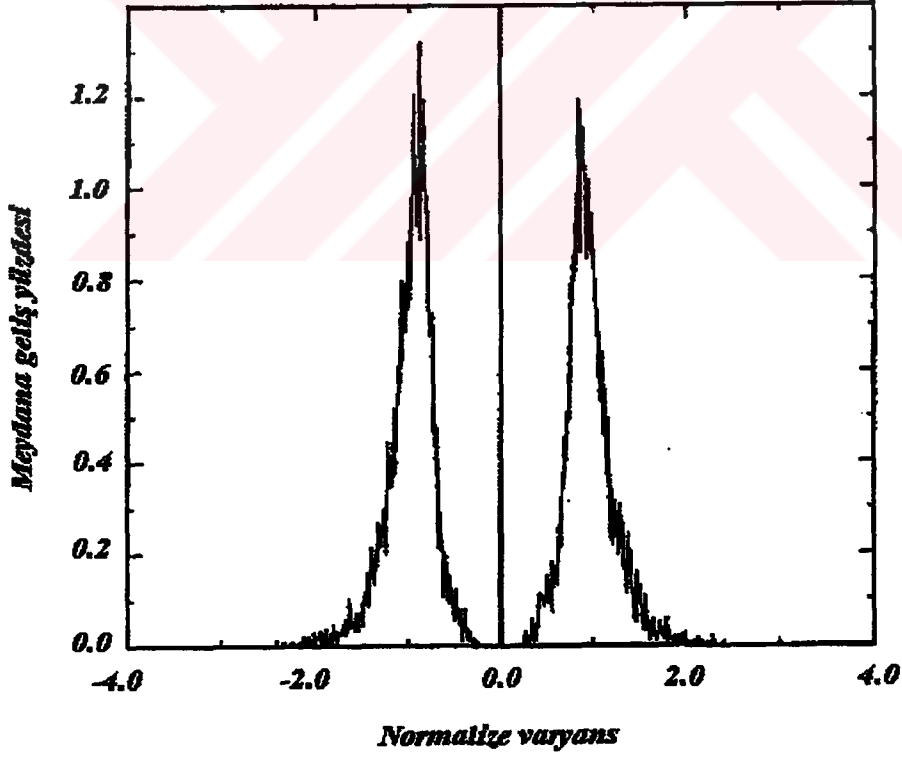


Şekil 6.11 SELP için LTP1 ve LTP2 gecikmelerinin dağılımı

2. yöntem için, her çerçeve için birim değişim bazında normalize edilmiş darbe genliklerindeki değişim Şekil 6.12'de gösterilmiştir. LTP hariç grafikte de görüleceği gibi histogram, sıfır etrafında düşük genlikli ve genliklerin çoğunluğu $\pm 3\sigma$ içinde olmak üzere simetriktir.



MPLPC darbelerinin LTP'siz daėılımı



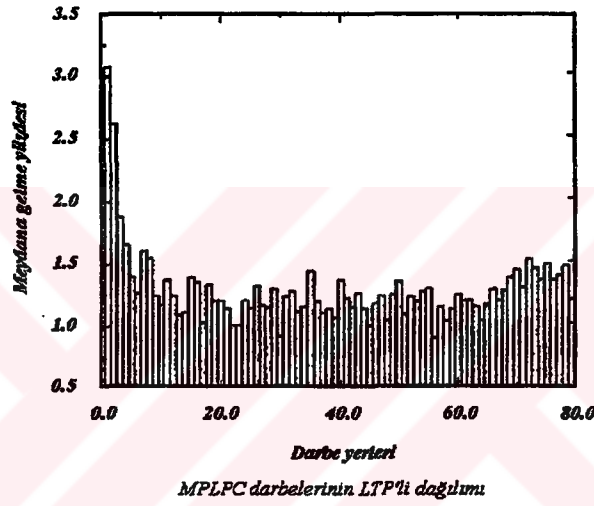
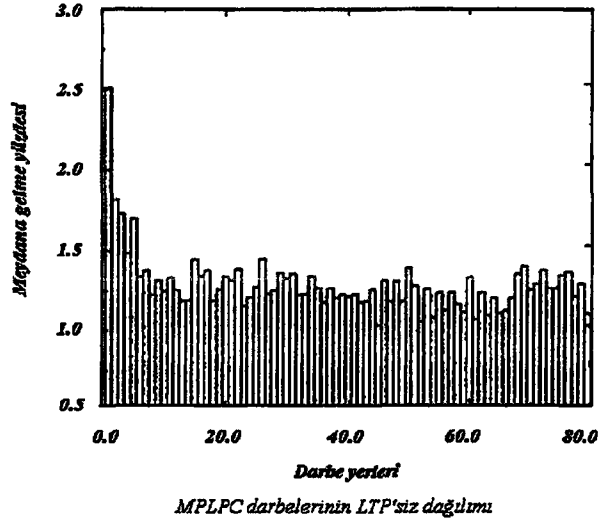
MPLPC darbelerinin LTP'li daėılımı

řekil 6.12 MPLPC darbe genliklerinin daėılım grafiėi

Değerlerin dağılımının daha fazla sınırlanmış olması, MPLPC'deki LTP'li darbelerin nicemlenmesinin, LTP'siz durumdan daha etkili olacağı anlamına gelir. Darbe nicemlemesindeki bu etkinlik çok önemlidir, zira LTP'li veya LTP'siz benzer bit hızlı MPLPC yaklaşımları için LTP'li MPLPC, LTP düzgün ve doğal oldukça, objektif ve subjektif olarak tercih edilmektedir.

Darbe konumlarının histogramı da Şekil 6.13'de gösterilmiştir. Çerçevenin başındaki darbe konumları, diğer konumlardan daha fazla gözetilmektedir, zira bunlar çerçeve içindeki hataların azalmasını sağlar. Türetmede kullanılan özilişki tipi analizlerden kaynaklanan düzgün olmayan darbe pozisyonu dağılımlarının üstesinden gelmek için, analizin ortak değişkenli tipi tavsiye edilmektedir.

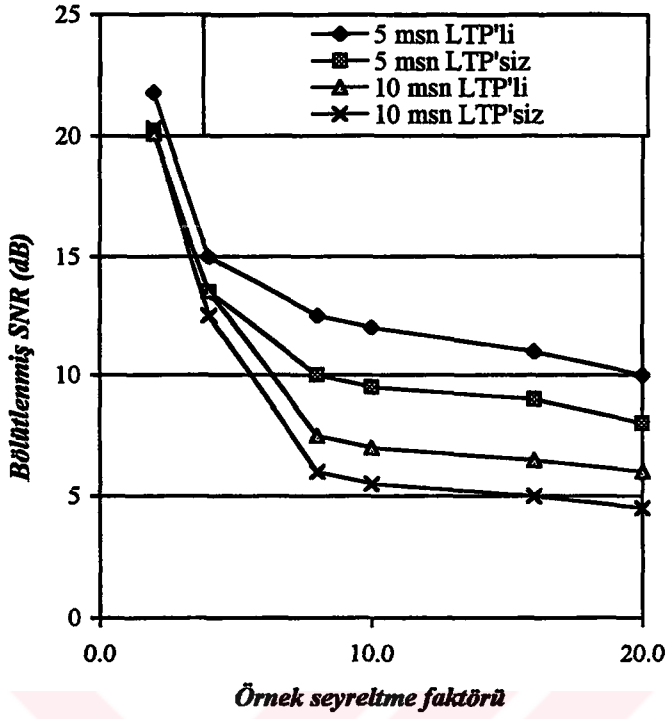
Ortak değişken metodu, kaskad filtrenin dürtü tepkisinin analiz bloğunun dışına dökülen kısmını dikkate alarak otomatik ortak ilişki analizinin blok kenar tesirlerinin yerini almayı dener. Darbelerin konumları da alt çerçeve boylarının seçimine etki eder. İdeal olarak geniş alt çerçeve boyları MPLPC'ye daha uygundur zira sınırlı sayıda darbe en faydalı konumlara, örneğin perde konumlarına yerleştirilebilir. Ancak, küçük alt çerçevelerle darbeler, konuşma işaretinin nispeten daha önemsiz ayrıntıları için bile görevlendirilir ki bu kodlama etkinliğini düşürür. Geniş alt çerçevenin dezavantajı karmaşıklığın artmasıdır.



Şekil 6.13 80 örneklilik alt çerçeve boyulu MPLPC darbe konumlarının dağılım grafiği

6.2.5.4 RPELPC

Daha önce de anlatıldığı üzere, darbe konumlarının önceden biçimlendirilmiş olmaları dışında RPELPC, MPLPC'ye benzer. Açık olarak biçimlendirmenin miktarı referans işaretini tahmin etmek için darbelerin serbestlik miktarını belirler. LTP'li ve LTP'siz 40 ve 80 örneklilik çerçevelerin analizi için RPELPC'in performansı üzerinde örnek seyreltme (*decimation*) oranının tesiri Şekil 6.14'de gösterilmiştir. Bağlı SNR farkları kısmen konuşma işaretinin kendisi ile ilgilidir. Ancak, LTP'nin eklenmesi, RPELPC'in performansını, özellikle de daha yüksek serbestlik oranlarında artırır.

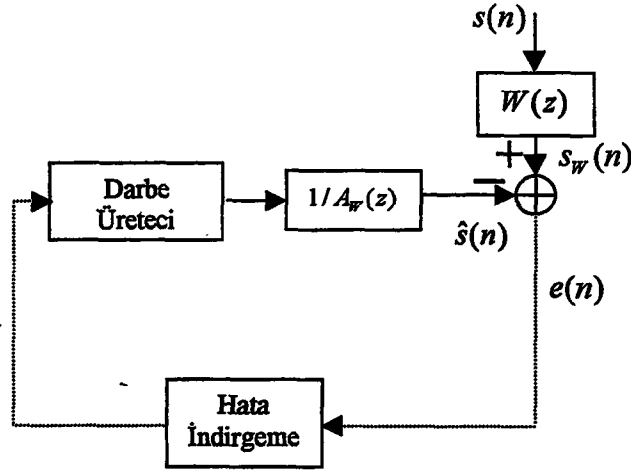


Şekil 6.14 Değişik yapıdaki RPELPC kodlayıcıların işaret gürültü oranları

Grafikten de görülebileceği gibi daha alttaki analiz çerçevesinde performans daha iyidir. Bunun sebebi, referans vektörüyle eşleşmek için uyarıyı değiştirmede daha fazla serbestiye sahip olunmasıdır.

6.3 ÇOKLU DARBELİ LPC (MPLPC)

Daha önce de anlatıldığı gibi, karmaşıklığı az olan bir MPLPC kodlayıcısındaki uyarma darbelerinin genlikleri, orijinal ile sentezlenmiş konuşma arasındaki hatanın azaltılması için ardışıl olarak tespit edilir. Darbelerin en uygun konumları, bir analiz bloğundaki tüm mümkün darbe pozisyonları ve genlikleri için hatanın hesaplanması ve en az hataya yol açacak biçimde izin verilebilir sayıda konum ve genliğin seçilmesi yoluyla bulunur. Basit bir MPLPC diyagramı Şekil 6.15'de gösterilmiştir.



Şekil 6.15 Basit bir MPLPC kodlayıcının blok diyagramı

STP sentez filtresinin ağırlıklı dürtü tepkisinin $h_w(n)$ olacağını varsayarak; m_i konumunda, g_i genliğindeki tek bir darbe uyarması için ağırlıklandırılmış karesel hata şu şekilde yazılabilir:

$$E_w = \sum_{n=0}^{L-1} [\tilde{s}(n) - g_i h_w(n - m_i)]^2 \quad (6.38)$$

Burada $\tilde{s}(n)$, STP filtresinin bellek etkisi çıkartılmış, algılama ile ağırlıklandırılmış hedef işarettir. En uygun darbe konumu, (6.38) denkleminin g_i 'ye göre türevi alınıp, türevi sıfırlamakla elde edilir.

$$\frac{\partial E_w}{\partial g_i} = -2 \sum_{n=0}^{L-1} [\tilde{s}(n) - g_i h_w(n - m_i)] \times h_w(n - m_i) = 0 \quad (6.39)$$

Buradan da g_i :

$$g_i = \frac{\sum_{n=0}^{L-1} \tilde{s}(n) h_w(n - m_i)}{\sum_{n=0}^{L-1} h_w(n - m_i) h_w(n - m_i)} \quad (6.40)$$

şeklinde bulunur ve genellikle aşağıdaki gibi yazılabilir:

$$g_i = \frac{\psi(m_i)}{\phi_{m_i, m_i}} \quad (6.41)$$

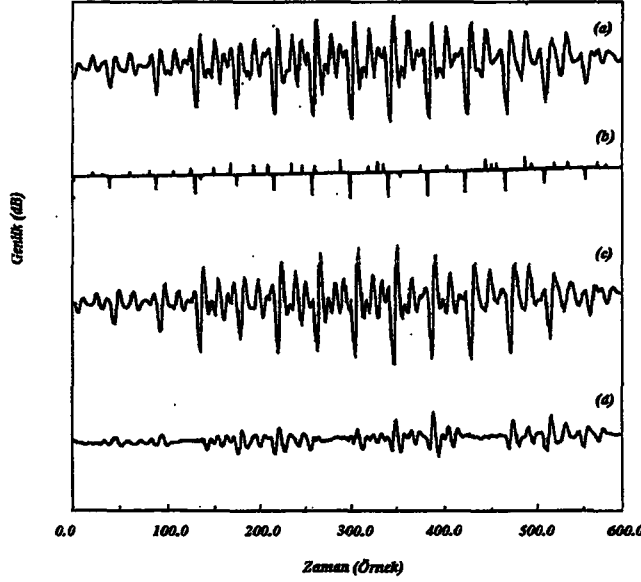
Burada $\psi(m_i)$, ağırlıklandırılmış hedef konuşma ile ağırlıklandırılmış STP dürtü tepkisi arasındaki çapraz ilişki ve ϕ_{m_i, m_i} ise, $0 \leq i \leq M-1$ ve m_i pozisyonları için ağırlıklandırılmış STP dürtü tepkisidir. (6.38) denklemin yerine (6.41) denklemi konularak, darbe konumları yönünden ağırlıklandırılmış karesel hatanın bir ifadesi bulunabilir.

$$E_w = \sum_{n=0}^{L-1} \left[\tilde{s}^2(n) - \frac{\psi^2(m_i)}{\phi_{m_i, m_i}} \right] \quad (6.42)$$

(6.42)'deki hatayı en aza indirmek için, tek bir darbe için en iyi pozisyonun, $\psi^2(m_i)/\phi_{m_i, m_i}$ terimini maksimize eden m_i değeri olduğu görülebilir. En uygun darbe arayışı sonuçlandırıldığında, bir sonraki darbe pozisyonunu saptamak için kullanılacak yeni bir referans işareti vermek için bu yeni bulunan darbe, giriş işaretinden çıkarılır. Böylece güncelleştirilmiş referans konuşma aşağıdaki gibi yazılabilir:

$$\tilde{s}_{i+1}(n) = \tilde{s}_i(n) - g_i h_w(n - m_i) \quad (6.43)$$

Geri kalan optimum darbe konum ve genliklerini bulmak için (6.40) denkleminde (6.43) denklemine kadar yürütülen adımlar tekrarlanır. Şekil 6.16, tipik bir konuşma işareti ile yukarıda bahsedilen AbS şeklinde üretilmiş bir uyarma işaretini göstermektedir. Şekil 6.14'de açıkça görülmektedir ki çoklu darbe yapısı, gırtlak yapısı, ses telleri özellikleri ve özellikle perde darbelerinin modellenmesi için esnek bir uyarma işaretinin üretilmesinde çok etkindir.



Şekil 6.16 MPLPC kodlayıcısı dalga şekilleri (a) orijinal konuşma, (b) çoklu darbe uyarması, (c) sentezlenmiş konuşma ve (d) hata işareti.

6.3.1 En Uygun Genlik Uyarmalı MPLPC

Yukarıda anlatılan AbS metodu basit ve hızlıdır, fakat bazı kusurları vardır. Çerçeve başına darbe sayısı arttıkça, her darbelerin ardışıl optimizasyonu duyarsızlaşır. Performansın iyileştirilmesi için tüm darbeler arasındaki karşılıklı ilişkilerin dikkate alınması gerekir. Darbeler arasındaki karşılıklı ilişkileri dikkate almak için, M darbelerini m_i pozisyonuna yerleştirdikten sonra ağırlıklandırılmış ortalama karesel hata aşağıdaki gibi verilebilir:

$$E_W = \sum_{n=0}^{L-1} \left[\tilde{s}(n) - \sum_{i=0}^{M-1} g_i h_w(n - m_i) \right]^2 \quad (6.44)$$

Yukarıdaki denklemin g_i 'ye göre türevi alınırsa:

$$\frac{\partial E_W}{\partial g_i} = -2 \sum_{n=0}^{L-1} \left[\tilde{s}(n) - \sum_{i=0}^{M-1} g_i h_w(n - m_i) \right] \times h_w(n - m_k) = 0 \quad k = 0, 1, \dots, M-1 \quad (6.45)$$

veya

$$\sum_{n=0}^{L-1} \tilde{s}(n) h_w(n - m_k) = \sum_{n=0}^{L-1} \left[\sum_{i=0}^{M-1} g_i h_w(n - m_i) \right] \times h_w(n - m_k) \quad k = 0, 1, \dots, M-1 \quad (6.46)$$

Sağ taraftaki toplamın tekrar düzenlemesi ile;

$$\sum_{n=0}^{L-1} \tilde{s}(n) h_w(n - m_k) = \sum_{i=0}^{M-1} g_i \left[\sum_{n=0}^{L-1} h_w(n - m_i) h_w(n - m_k) \right] \quad k = 0, 1, \dots, M-1 \quad (6.47)$$

elde edilir. Burada;

$$\phi(k, i) = \sum_{n=0}^{L-1} h_w(n - k) h_w(n - i) \quad (6.48)$$

ve;

$$\psi(k) = \sum_{n=0}^{L-1} \tilde{s}(n) h_w(n - k) \quad (6.49)$$

tanımlanırsa (6.47) deklemini basitleşerek aşağıdaki biçime dönüşür:

$$\psi(m_k) = \sum_{i=0}^{M-1} g_i \phi(m_i, m_k) \quad k = 0, 1, \dots, M-1 \quad (6.50)$$

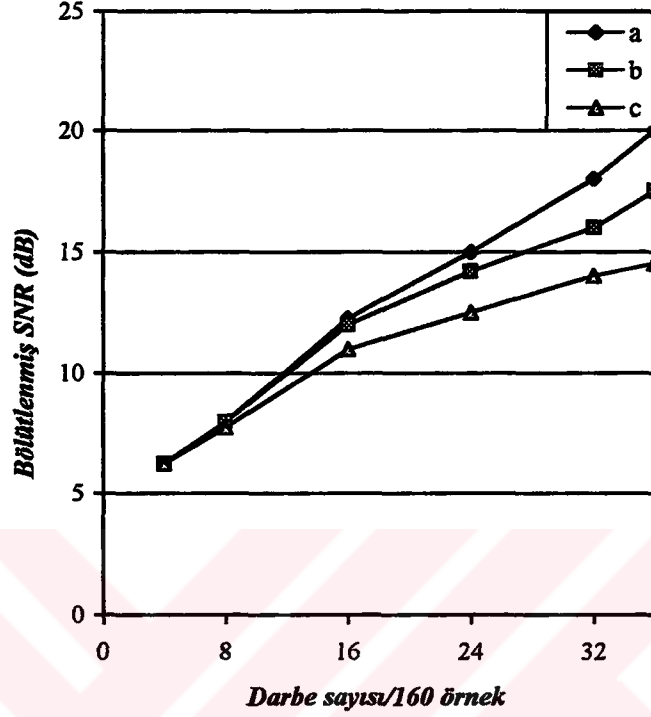
ki bu da aşağıdaki gibi bir ilişki matrisi biçiminde verilebilir:

$$\begin{bmatrix} \psi(m_0) \\ \psi(m_1) \\ \vdots \\ \psi(m_{M-1}) \end{bmatrix} = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{M-1} \end{bmatrix} \begin{bmatrix} \emptyset_{m_0 m_0} & \emptyset_{m_0 m_1} & \dots & \emptyset_{m_0 m_{M-1}} \\ \emptyset_{m_1 m_0} & \emptyset_{m_1 m_1} & \dots & \emptyset_{m_1 m_{M-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \emptyset_{m_{M-1} m_0} & \emptyset_{m_{M-1} m_1} & \dots & \emptyset_{m_{M-1} m_{M-1}} \end{bmatrix} \quad (6.51)$$

Şimdi g_i genlikleri, ilişki matrisinin *Cholesky* ayrışımı kullanılarak çözülebilir.

Yukarıdaki analiz kullanılarak darbe genliği için iki yeniden optimizasyon yordamı uygulanabilir [40]. Tüm M darbeleri bir alt çerçeve içine yerleştirildikten veya her yeni darbenin yeri tespit edildikten sonra, genlikler tekrar optimize edilebilir. Doğal olarak ikinci yöntem, matris tersi gibi daha büyük hesaplama yükünü beraberinde getirir, ancak önceki yöntem ile karşılaştırıldığında toplam kalite daha yüksektir. Önceki metodda, genliğin yeniden optimizasyonu her alt çerçeve sonunda gerçekleşir ve gereken matris tersi boyutu $M \times M$ 'dir. İkinci metodda, genliğin yeniden optimizasyonu, 1×1 'den $M \times M$ matris boyutlarına kadar M 'nin katlarında gerçekleşir. Şekil 6.17, üç değişik algoritma için *segSNR*'ye (bölütlenmiş SNR-*segmented SNR*) göre darbe sayısının değişimini göstermektedir. (c) Eğrisi, temel ardışıl MPLPC kodlayıcısı içindir. Darbe sayısı arttıkça *segSNR*'ler artmakta, ancak 30 darbe ve 160 örnekten sonra artık performansının sonuna geldiği de görülmektedir. (b) ve (a) eğrileri, daha iyileştirilmiş MPLPC algoritmaları içindir, şöyleki: sırasıyla tüm darbelerin konumları saptandıktan sonra genlik optimize edilir veya her yeni darbenin konumu bulunduğundan sonra genlik optimize edilir. Objektif sonuçlar, beklendiği üzere, (b) eğrisinin (a) eğrisinden daha düşük *segSNR* verdiğini göstermektedir. Subjektif sonuçlar ise, (a) ve (b) eğrileri, genliğin yeniden optimizasyonu sürecinde yalnız daha fazla sayıda darbe kullanıldığında daha iyi sonuç verir. Bu, her 20 msan ve üstünde takriben 8 darbe mertebesinde. Ortaya çıkan bu sonuç normaldir, zira yeniden optimizasyon süreci, yakın yerleştirilmiş darbelerin performansını artırır. Ortalama olarak 4-5 msan başına 5 darbe iyi bir konuşma kalitesi elde etmek için uygundur. Darbe araması esnasındaki ana sorunlardan, biri darbe çiftlemesidir (*doubling*). Darbe çiftlemesi genellikle, her 10

msan'de 8 darbeden fazla olan ses bölgelerinde oluşur ve zaten seçilmiş darbe konumlarının tekrar seçilmesi anlamını taşır.



Şekil 6.17 Bölütlenmiş işaret gürültü oranı ile darbe sayısının değişimi. (a) her darbeden sonra genlik yeniden optimize ediliyor, (b) tüm darbelerden sonra bir kere genlik yeniden optimize ediliyor ve (c) genliğin yeniden optimizasyonu olmaksızın ardışıl MPLPC

Bu etkiyi önlemek için yeni bulunan darbe genliği, mevcut genliğe eklenir veya zaten saptanmış darbe konumları, daha sonraki darbe konumu seçiminden ayrı tutulur. Darbe konumlandırıldıktan sonra her defasında ortak (*joint*) darbe genliğine ait yeniden optimizasyon uygulandığı takdirde, darbe katlanması otomatik olarak önlenir.

6.3.2 Uzun Dönemli Öngörülü MPLPC

Temel bir çoklu darbeleri kodlayıcı, orta bit hızlarında tatminkar konuşma kalitesi üretir. Ancak, bit hızı düşürüldükçe konuşma kalitesinde bozulmalar gözlenir. Bu, özellikle kadın konuşmasında rastlanan yüksek perdeden ses bölgeleri için geçerlidir. Karşılaşılan bu durum, darbelerin yoğunluğunun asıl tiz darbeleri modellemek için kullanılması ve dolayısıyla geriye kalan uyarma işaretinin modellenmesi için nispeten az sayıda darbe kalması nedeni ile oluşur. Şekil 6.18'de görülebileceği gibi AbS devresine uzun süreli bir öngörücünün yerleştirilmesiyle bu tür etkiler azaltılabilir.

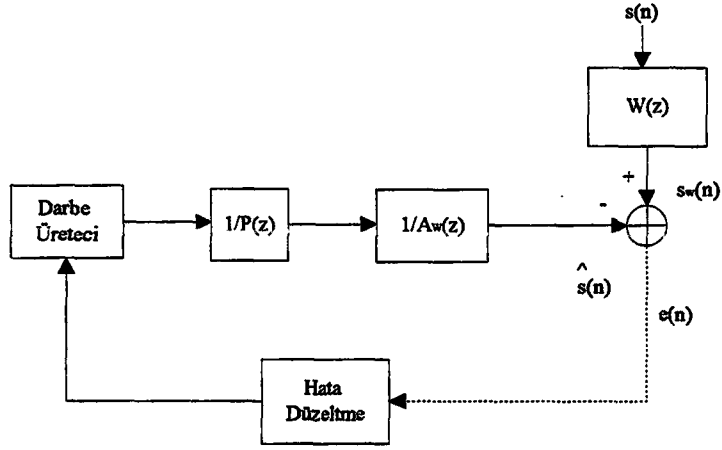
Singhal ve *Atal* [40], tiz parametrelere en uygun değerleri veren bir kapalı devre çözümü önermiştir. MPLPC'de kullanılan en popüler uzun süreli öngörücü, tek ara-uçlu (*single tap*) öngörücüdür.

Bu durumda STP sentez filtresinin girişine uygulanan $\hat{r}(n)$ aşağıdaki gibi ifade edilebilir:

$$\hat{r}(n) = Gx(n) + \beta \hat{r}(n - D) \quad 0 \leq n \leq L - 1 \quad (6.52)$$

Sıfır giriş uyarması ($x(n)=0$) varsayıldığında yukarıdaki denklem (6.53)'deki gibi olur:

$$\hat{r}(n) = \beta \hat{r}(n - D) \quad (6.53)$$



Şekil 6.18 LTP'li MPLPC AbS yordamı blok diyagramı

Eğer $h_w(n)$ STP sentez filtresinin ağırlıklandırılmış dürtü tepkisi olarak alınırsa, sentezlenmiş konuşma $\hat{s}(n)$ aşağıdaki gibi ifade edilebilir:

$$\hat{s}(n) = \beta \sum_{k=0}^n \hat{r}(n-k-D)h_w(k) \quad (6.54)$$

β ve D 'yi bulurken orijinal ağırlıklandırılmış konuşma $\tilde{s}(n)$ ile sentezlenmiş konuşma $\hat{s}(n)$ arasındaki karesel hata en aza indirilmelidir.

$$E_w(\beta, D) = \sum_{n=0}^{L-1} \left[\tilde{s}(n) - \beta \sum_{k=0}^n \hat{r}(n-k-D)h_w(k) \right]^2 \quad (6.55)$$

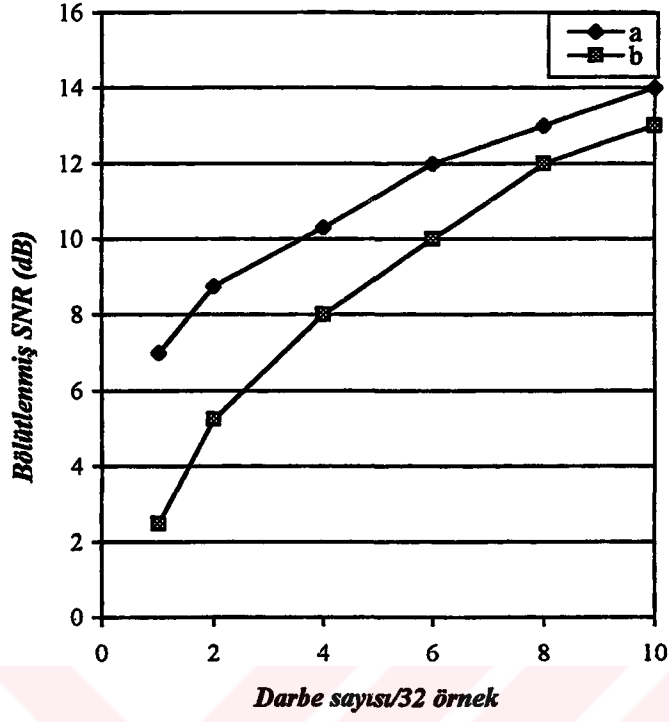
E_w , aşağıdaki durumda en aza indirilebilir:

$$\beta(D) = \frac{\sum_{n=0}^{L-1} \tilde{s}(n)z_D(n)}{\sum_{n=0}^{L-1} z_D^2(n)} \quad (6.56)$$

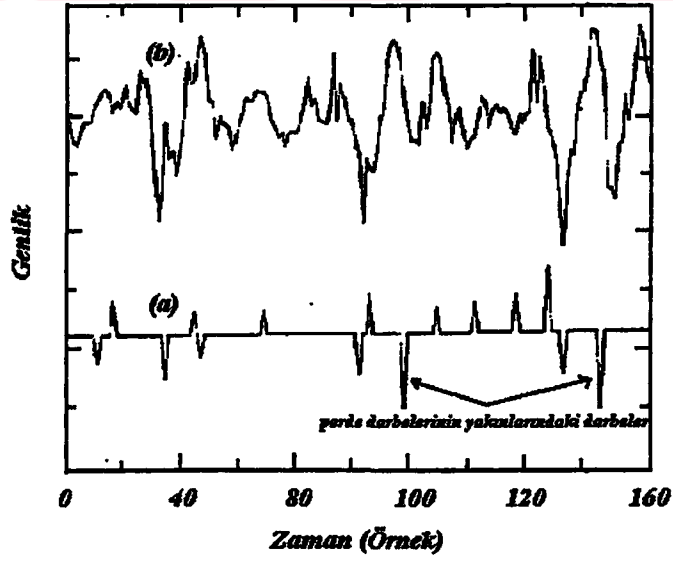
Burada $z_D(n) = \sum_{k=0}^n \hat{r}(n-k-D)h_w(k)$ 'dır ve dolayısıyla

$$E_w(D) = \sum_{n=0}^{L-1} \tilde{s}^2(n) - \frac{\left[\sum_{n=0}^{L-1} \tilde{s}(n)z_D(n) \right]^2}{\sum_{n=0}^{L-1} z_D^2(n)} \quad (6.57)$$

şeklinde ifade edilebilir. Yukarıdaki denklem, belirli bir dizi içinde D 'nin tüm değerleri için hesaplanır ve $E_w(D)$ 'yi en aza indiren değer öngörücü gecikmesi olarak alınır. Öngörücü kazancı β , daha sonra (6.56) denklemi kullanılarak bulunur. Bu parametreleri kullanarak LTP'nin katkısı çıkarıldıktan sonra orijinal ve sentezlenmiş ağırlıklandırılmış konuşma işaretleri arasındaki karesel hatayı en aza indirmek için çoklu darbe uyarması, daha önceki bölümde anlatıldığı üzere hesaplanır. LTP'yi kapalı devre analizi içine yerleştirmenin avantajı, LPC nicemleme etkileri ve STP bellekleri durumları dahil hataların en aza indirilebilmesi için, en uygun LTP parametrelerinin hesaplanabilmesidir. Bu yöntemin sakıncası, β ve D 'yi hesaplariken gereken hesaplama yüküdür. Karmaşıklığı azaltmak için uzun süreli öngörücüye açık devre çözümü uygulanabilir. Bu sistemde LTP parametreleri, AbS devresi dışında kalan LPC artık işareti kullanılarak saptanır. Bunun dışında uyarma analiz prosedürü aynıdır. Şekil 6.19, bir-ara uçlu kapalı devreli ve uzun süreli öngörücüsüz MPLPC'nin objektif performansını göstermektedir. Uzun süreli algılamanın tüm darbe hızlarında daha yüksek $segSNR$ sağladığı açıktır. Yüksek darbe hızlarında, MPE'nin darbeleri temel tizliği modelleyeceği için LTP'li ve LTP'siz MPLPC arasındaki subjektif fark doymaya gitmektedir.



Şekil 6.19 MPLPC'nin (a) LTP'li ve (b) LTP'siz performansı



Şekil 6.20 Uyarma dalga biçimleri (a) LTP uyarması ve (b) Çoklu darbeli uyarma

LTP'nin dahil edilmesi işaretin yaklaşık periyodikliğini ortadan kaldırmak için öngörülmüş olmasına rağmen, MPE yine de, gerçek perde darbelerinin üstüne veya yakınına darbe atamaya devam eder. Uyarma işareti ile LTP'den tekrar oluşturulmuş artık karşılaştırıldığında, durumların pek çoğunda, Şekil 6.20'den de görülebileceği gibi, MPE'deki darbelerin perde darbelerinin çok yakınına yerleştikleri gözlenir.

6.3.3 Darbe Yeri Kodlaması

Darbe pozisyonlarının kodlanması genellikle tek tek sayan kaynak kodlama teknikleri kullanılarak yapılır [35]. Aşağıdaki denklem, M darbelerini bir L örnekleri alt çerçevesine yerleştirmek için değişik olasılık sayısını vermektedir :

$$\Theta = \binom{L}{M} = \frac{L!}{M!(L-M)!} \quad (6.58)$$

Bu durumda pozisyonları kodlamak için gereken bit sayısı da,

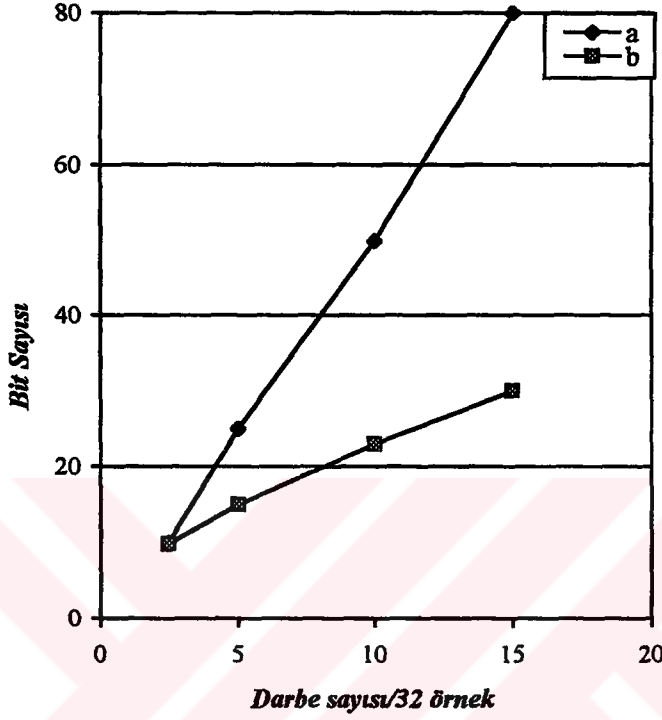
$$B_{\min} = \lceil \log_2 \Theta \rceil \quad (6.59)$$

olur. Burada $\lceil - \rceil$, Θ 'a eşit veya daha büyük en yakın tamsayıdır. Bu tür darbe pozisyon kodlama metodları, darbe pozisyonlarının vektör nicemlemesi olarak kabul edilebilir. Bu teknikler kötü kanal şartlarında iyi netice vermediğinden alternatif kodlama metodları kullanılır. Diğer bir metod, darbe pozisyonlarının bağımsız olarak kodlanmasıdır. Her ne kadar bu yöntem daha yüksek bit hızlarına yöneltirse de kanal hatalarına karşı daha dayanıklıdır. Bağımsız darbe pozisyon kodlaması için gereken bit sayısı aşağıdaki denklemle verilmektedir:

$$B_{\min} = M \times \lceil (\log_2 L) \rceil \quad (6.60)$$

Şekil 6.21, $L=32$ iken darbe pozisyonlarının bileşik ve bireysel kodlanması arasındaki karşılaştırmayı göstermektedir. Çok yüksek darbe hızlarında bileşik

kodlama açıkça en etkili olanıdır, ama çok az miktarda darbe durumunda aradaki fark 2 bitten azdır. Bu kodlama hızlarında bağımsız kodlama için analiz çerçevesi başına 3 bit fazlalığın doğurduğu sakınca, kodlayıcının karmaşıklık ve sağlamlığına yansır.



Şekil 6.21 (a) Bağımsız ve (b) bileşik darbe pozisyonu kodlamasının karşılaştırılması

6.3.4 Darbe Genlik Kodlaması

Darbe genliklerinin kodlanması için normalizasyon kaçınılmazdır, zira darbe genliklerinin dinamiği oldukça geniştir ve doğrudan nicemleme fazla sayıda bit gerektirir. Normalizasyon, genliklerin rms değerleri kullanılarak elde edilebilir. Bu nedenle genliğin rms değerinin de iletilmesi gereklidir. Bu da kaçınılmaz olarak daha yüksek bit hızlarına yol açar. MPLPC tasarımlarının çoğunda ilk darbenin nicemlenmesi, çok sayıda düzgün olmayan nicemleme seviyelerinin (genellikle 5 bit veya fazlası) eklenmesi ile gerçekleştirilir; alt çerçevede kalan darbeler, ilk darbe

genliği ile normalize edilir ve daha az nicemleme seviyesi kullanılarak kodlanır (tipik olarak 3 bit)[35].

Yan bilgi gerektirmeyen diğer dört genlik nicemleme yapısı aşağıda incelenmiştir:

- (a) Adaptif İlk Darbe Genliği
- (b) Adaptif Önceki Darbe Genliği
- (c) Adaptif Önceki Enerji Alt Çerçevesi
- (d) Adaptif LTP Filtre Belleği

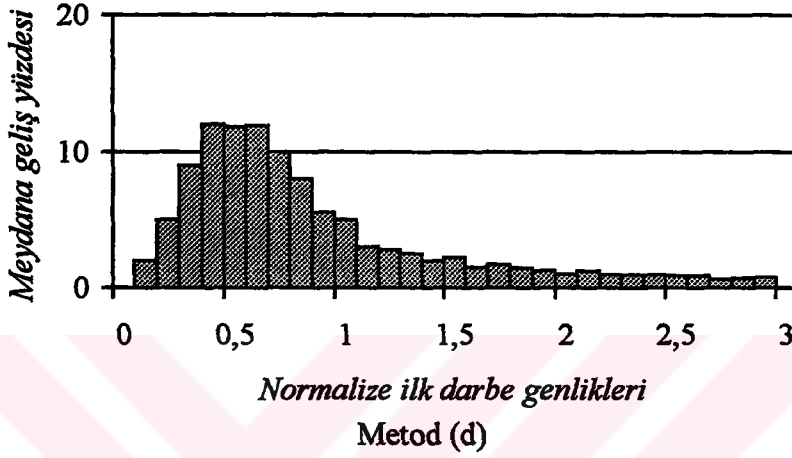
i^{nci} darbe büyüklüğünün g_i olduğu varsayıldığında, (a) yapısındaki $i > 0$ için g_i büyüklükleri, bloktaki ilk darbe g_0 'a göre normalize edilir. İlk darbenin kendisi (c) ya da (d) yaklaşımı kullanılarak kodlanır. İkinci yordam (b)'de g_i büyüklükleri ($I > 0$ için), o andaki bloğun içinde bulunan bir önceki en uygun darbe büyüklüğüne göre normalize edilir. Yine g_0 kodlamasında (c) ve (d) yapıları kullanılır. (c) metodunda, bir önceki çerçeveden uyarma bloğu enerjisi kullanılarak büyüklükler normalize edilir ve sonuç olarak (d)'de LTP filtre belleğindeki enerjiden, g_0 dahil tüm darbelerin normalize edilmesi için yararlanır. Kod ile uyarılan kodlayıcıların normalize edilmesi için buna benzer bir yaklaşım kullanılmaktadır [41].

Şekil 6.22 ve 6.23, yukarıda belirtilen uyarılama tekniklerini kullanarak, hem erkek hem kadın sesleri içeren 10 dakikalık konuşmanın normalize darbe büyüklükleri için dağılım grafiklerini göstermektedir. Birinci darbe hariç olmak üzere ilk iki yapı dikkate alındığında, tatmin edici konuşma kalitesi için her darbe başına 3 bit gerektiğinden, birbirine çok benzer. Bu metodların bellibaşlı sorunu, gürültülü ortam nedeniyle ilk büyüklük g_0 bozulabileceğinden alt çerçevede kalan darbe büyüklüklerinin hatalı olabileceğidir. Bozulan darbe ilk darbe değilse, (a) yapısı daha sağlamdır; zira (b)'deki g_i büyüklüklerinden herhangi birindeki hata takibeden darbelere taşınacaktır.

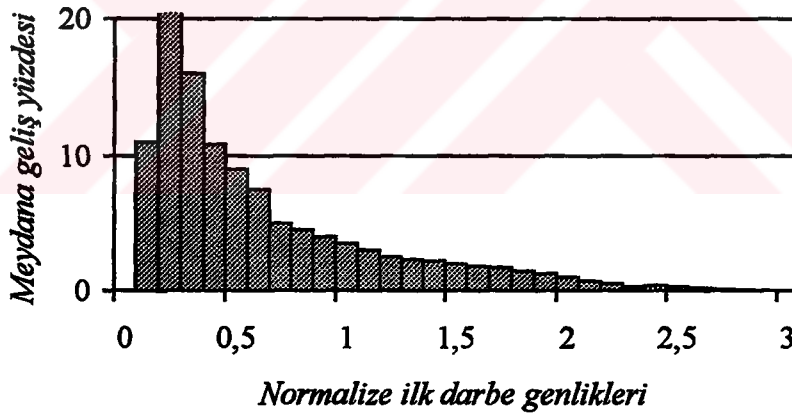
6.23'deki dağılım grafikleri arasından (c) en iyi gözükmemektedir. Alt çerçeve boyu daha küçükse ve daha fazla darbe içeriyorsa, bu teknik daha iyi bir normalizasyon değeri kümesi (*range*) (yani küçük dinamik değeri kümesi) verir. Daha büyük alt

çerçeve ve daha az darbe için ise, enerjideki değişkenlik artar. Dolayısıyla, belirli sınırlamalar konulmadığı takdirde bu teknik tatminkar şekilde uygulanamaz. Bu yapının sağlamlığı dikkate alındığında bundan önceki alt çerçevelerde bulunan herhangi bir hata, mevcut ve bundan sonraki alt çerçevelere de etki edecektir. Bu da, kaçınılmaz olarak daha büyük alt çerçevelerde daha büyük bozulmalara yol açar.

Metod (c)



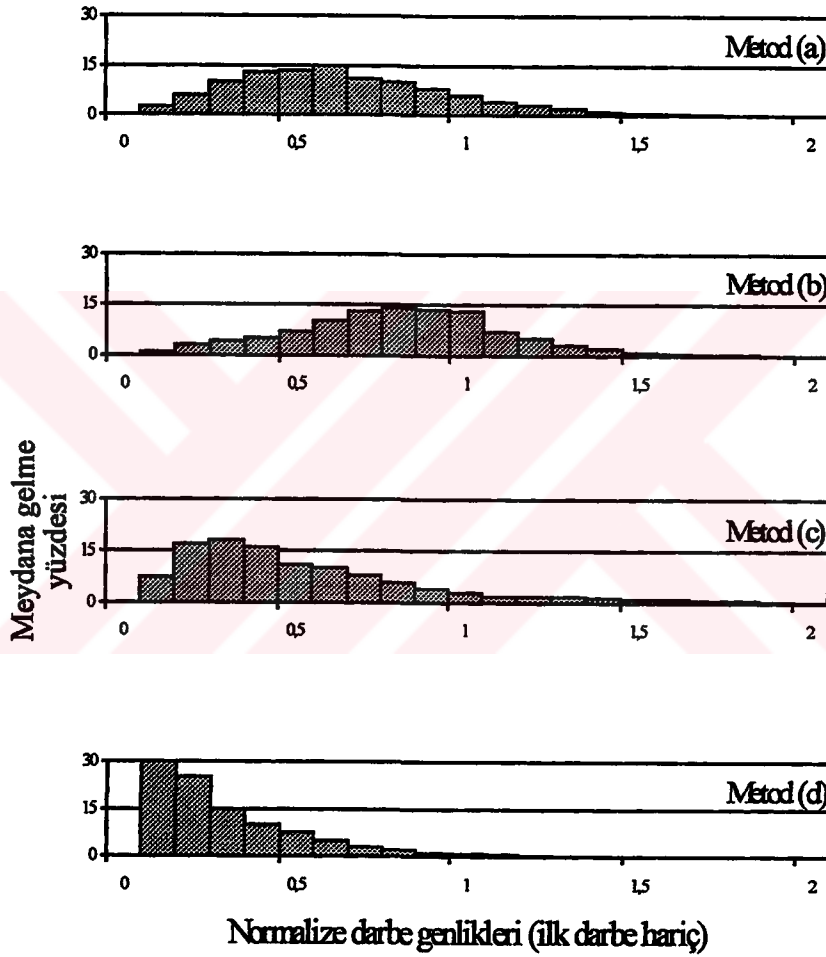
Metod (d)



Şekil 6.22 İlk genlik normalizasyonu için dağılım grafikleri

Aktif konuşma bölgeleri içinde LTP filtre belleğinin enerjisi, giriş işaretinin enerjisine çok benzer. Referans işaretinin biçimlendirilmesi sırasında giriş işareti enerjisinin yarıdan fazlası kaybolur. Kalan enerji daha sonra darbe genliklerine yansıtılır. Dolayısıyla, bunların LTP filtre belleği enerjisine oranla normalizasyonu,

normalize büyüklükleri 0 ile 1.5 civarı arasında bir değerde sınırlar (Şekil 6.20 ve 6.21). Bu yapı ile ilgili ana sorunlardan biri, muhtemelen kod çözücüdeki saat senkronizasyonu eksikliği veya ciddi kanal hatalarından dolayı kodlayıcı ve kod çözücüdeki uzun süreli öngörücü belleklerinin değişik olabileceğidir. Bu durum kod çözücünün kararlı olmamasına yol açabilir. Bu sorunun üstesinden gelmek için hem kodlayıcı hem kod çözücü içindeki uzun süreli öngörücülerin bellekleri periyodik olarak ilk duruma getirilmelidir (*reset*).



Şekil 6.23 İlk darbe dışında normalize edilmiş darbe genliklerinin dağılım grafikleri

6.3.5 Ortak Darbe Genliği Nicemlemesi (JPAQ)

Çoğunlukla ilk seçilen darbe en yüksek genlikli olandır. Bunu İkinci darbe takip eder, sonra üçüncü ve hesaplama sırasına göre (yani $|g_0| > |g_1| > |g_2| \dots > |g_M|$) devam eder. Bu davranış, darbe büyüklüklerinin hesaplama sırasına göre gösterildiği Şekil 6.24'de açıkça görülmektedir.

Bu tür bir davranış, ortak bir genlik nicemleme tekniğinde kullanılabilecek, darbe büyüklükleri arasındaki bir ilişkinin var olduğunu belirtir. $i > 0$ için $|g_i|$ büyüklüklerinin basit bir ifade yolu, $|g_0|$ bilgisi ile $i > 0$ için $|g_i|$ büyüklüklerini tekrar yaratacak bir katsayının bulunmasıdır. Yani [32]:

$$|g_i| = \lambda_i |g_{i-1}| ; i > 0 \quad (6.61)$$

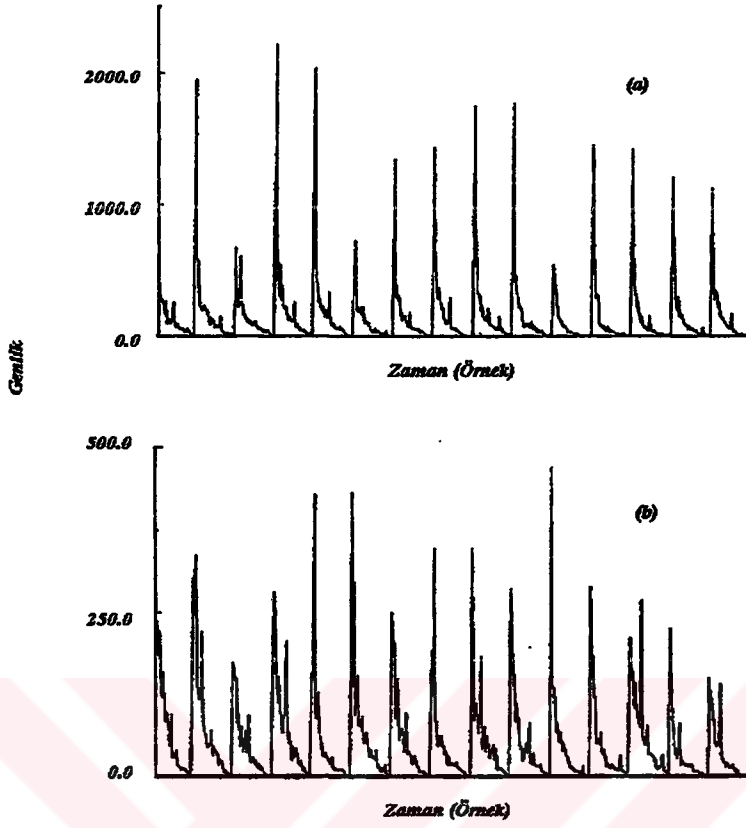
Burada λ_i , bitişik iki darbe büyüklüğü arasındaki orandır. Etkin bir nicemleme için $\lambda_i = \lambda$ ($i > 0$ için) varsayıldığında, (6.61) denklemi aşağıdaki gibi yazılabilir:

$$|g_i| = \lambda |g_{i-1}| \text{ for } i > 0 \quad (6.62)$$

λ için bir değer hesaplamının basit bir yaklaşımı, λ 'yı birbirini takip eden darbeler arasındaki oranların ortalaması olarak varsaymaktır. Dolayısıyla λ :

$$\lambda = \frac{\sum_{i=1}^{M-1} |g_i|}{\sum_{i=1}^{M-1} |g_{i-1}|} \quad (6.63)$$

şeklinde yazılabilir.



Şekil 6.24 (a) yüksek enerjili (b) alçak enerjili konuşma bölgeleri için darbe hesaplama sırasına göre darbe genlikleri

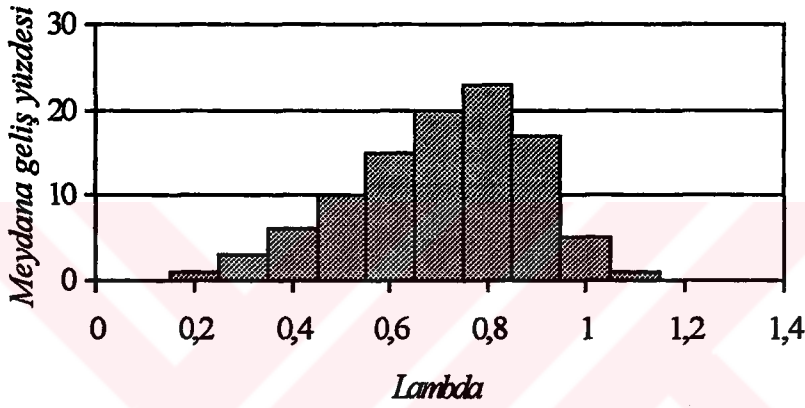
(6.63) denklemindeki ilişki tüm darbelere aynı ağırlığı verir. Bu, toplam darbe büyüklüğü nicemleme hatasını en aza indirmeyebilir. Bu nedenle, toplam darbe büyüklüğü nicemleme hatasını en aza indirmek için, aşağıdaki denklem kullanılarak optimum λ çözülür:

$$E_w = \sum_{i=1}^{M-1} \left[|g_i| - \lambda |g_{i-1}| \right]^2 \quad (6.64)$$

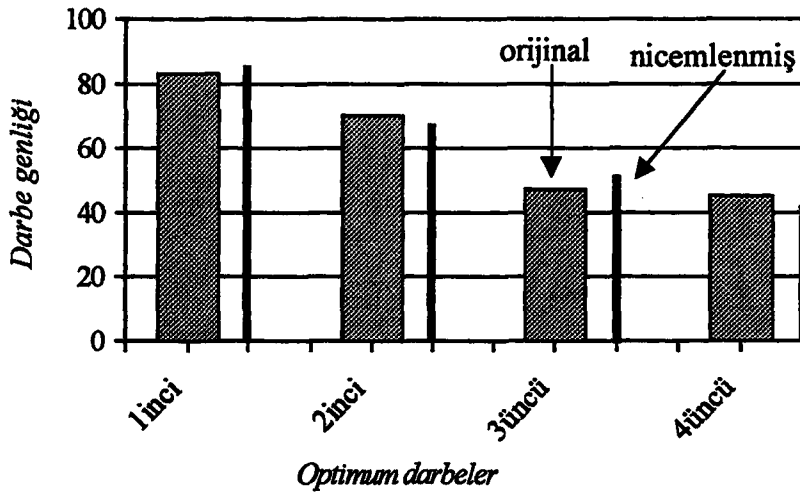
E_w 'nin λ 'ya göre olarak en aza indirilmesiyle:

$$\lambda = \frac{\sum_{i=1}^{M-1} |g_i| |g_{i-1}|}{\sum_{i=1}^{M-1} |g_{i-1}|^2} \quad (6.65)$$

denklemleri bulunur. Bu yaklaşımda, (d) yöntemini kullanan ilk darbenin büyüklüğü ile birlikte her darbenin işaret bitleri ilk olarak kodlanır. Daha sonra, denklem (6.65)'den elde edilen tek bir katsayı, 3 bitlik düzgün olmayan bir nicemleme kullanılarak kodlanır; ki bu da, darbe büyüklüklerini önceki darbe büyüklüklerine oranla ifade eder.



Şekil 6.25 λ değerinin histogramı



Şekil 6.26 Hesap sırasına göre orijinal ve ortak nicemlenmiş darbe genlikleri

Şekil 6.25, (6.65) denklemi ile hesaplanmış λ değerlerinin bir histogramını göstermektedir. λ 'nın 0 ile 1.0 arasındaki değer bölgesi yapısı, λ için 3 bitlik bir niceme ile oldukça etkin kılar.

Şekil 6.26, (6.62) denklemi kullanılarak elde edilen orijinal ve nicemlenmiş darbe genliklerini göstermektedir. Bu yaklaşımla, 1nci, 2nci ve 3üncü darbeler için sırasıyla 28, 21.5 ve 20 dB'lik işaret/nicemeleme gürültüsü oranları elde edilebilir. Alt çerçeve başına 3 darbeden yüksek darbe hızları için 500-1000 bit/san civarında kazanç sağlamak mümkündür. Ancak, ortak nicemeleme gürültüsü beş darbeden sonra artmaya başlar. Bunun sebebi, genliklerin, darbe büyüklükleri arasındaki ilişkiyi daha doğru olarak tanımlayacak katsayıya ihtiyaç duyarak üstel biçimde yok olmasıdır (Şekil 6.24).

6.4 KOD İLE UYARILAN DOĞRUSAL ÖNGÖRÜLÜ KODLAMA (CELP)

AbS-LPC yapıları arasında 8 kbit/san'nin altındaki çalışmalarda en çok uygulanan yaklaşım Kod ile Uyarılan Doğrusal Öngörülü Kodlama (CELP)'dir. CELP yapısı, *Atal* tarafından 1982'lerde ortaya atılmıştır [42], ancak temel CELP algoritmasının uygulamaları yarı iletken teknolojisi ve sayısal işaret işleme tekniklerinin gelişmesiyle birlikte son zamanlarda gerçekleşmiştir; bunların en önemlisi ABD Savunma Dairesi'nin (DoD) federal standardı olan 4.8 kbit/san'lik CELP kodlayıcısıdır [43]. Şekil 6.27'de CELP kodlayıcısının blok diyagramı gösterilmiştir ve algoritma aşağıdaki şekilde çalışır:

- (1) Orijinal konuşma işareti $s(n)$ önce 20-30 msan'lik analiz çerçevelerine bölünür. Konuşmanın spektral zarfının modellenmesi için, kısa dönemli öngörücüde (STP) kullanılacak bir LPC katsayılar kümesini elde etmek üzere, $s(n)$ 'e ait bir çerçeveye LPC analizi uygulanır.
- (2) Daha sonraki analizlerde belleksiz bir STP sağlamak için, STP belleği (ilk koşullar) referanstan çıkarılır.

- (3) LPC katsayıları bulunduğundan sonra uzun süreli öngörü (LTP) yapılabilir. LTP analizi genellikle LPC çerçevesinin astkatları (örn. 5-10 msan) üzerine uygulanır. LTP analizi, türetilmiş LPC katsayılı ters filtre ile üretilmiş artık üzerinde yapılır, yani açık devre metodunda (OLM) veya orijinal konuşma üzerinde (daha önceki bloklardan STP belleği çıkarılmış olarak); kapalı devre metodu (CLM). Her iki analiz de D gecikmesine ve buna bağlı β_i katsayılarının oluşmasına (ölçekleme faktörleri) neden olur; i endeksi burada filtre ara ucu sayısını temsil etmektedir. LTP, sentezlenmiş konuşmada orijinal işarete bulunan periyodikliğin oluşmasını sağlar.
- (4) İki sentez filtresinin parametreleri bulunduğunda uyarma tespit edilir. Uyarmanın güncelleştirilmesi yine LPC çerçevesinin ast katları üzerinde yapılır. Standart CELP'te uyarma işareti $x(n)$, rastgele beyaz Gauss dizilerinin bulunduğu bir kitaptan seçilir. En iyi $x(n)$ 'i bulmak için sürdürülen arama üç temel adımdan oluşur: (i) filtrelenenmiş Gauss dizilerinden bir takım üretmek (yapay konuşma); (ii) her dizinin hatası için objektif bir hata hesaplamak; (iii) en az hatalı diziyi seçmek. Böylece en az objektif hatanın karesi ve buna bağlı ölçekleme faktörünü üreten kod kitabı vektörü seçilir.
- (5) Sentezleyicide filtrelerin ilk durumları (yani hafıza) yeniden elde edilir ve yapay konuşma, herhangi bir ağırlıklandırma algılaması olmadan ölçekli optimum kod kitabı dizisinin filtreleneceği yoluyla üretilir.

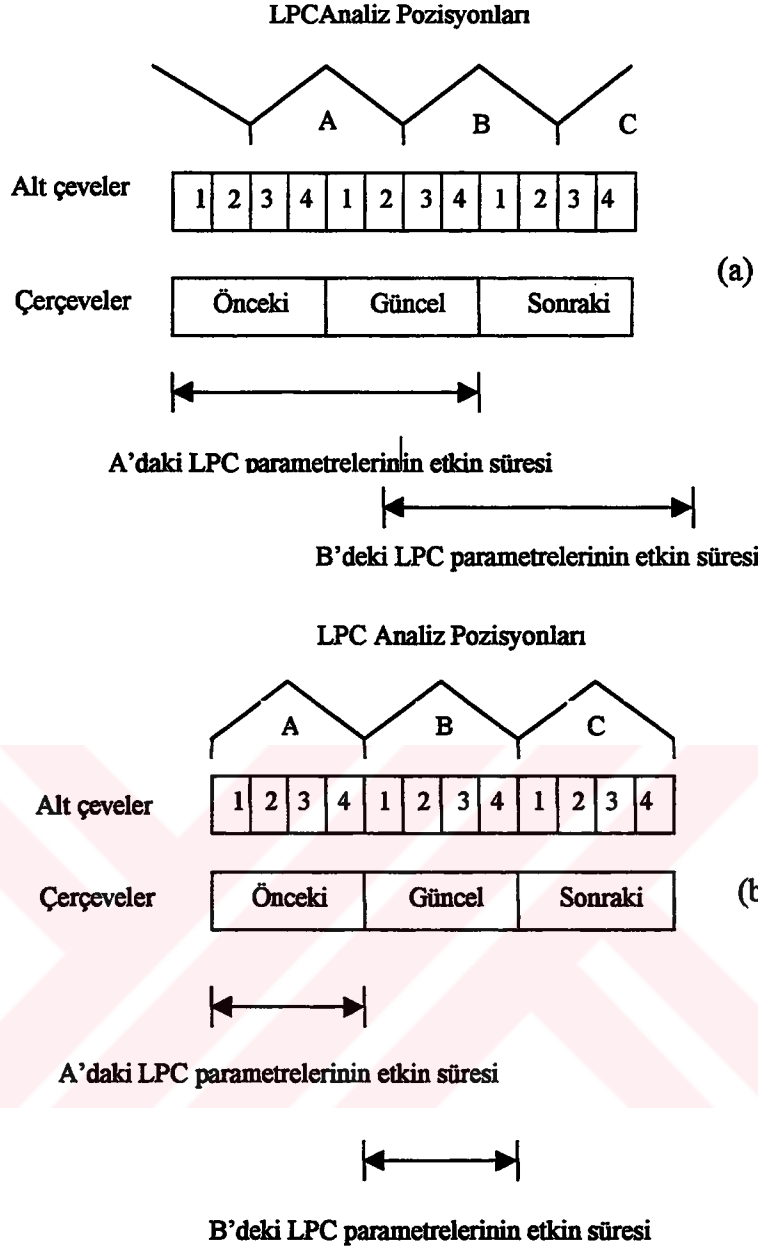
Yukarıdaki açıklamadan da anlaşılacağı üzere hesaplamanın üç bloğa bölünebileceği açıktır: (i) LPC analizi, (ii) LTP analizi, (iii) kod kitabını arama.

algoritmik gecikme sorunu, LPC analizinin geriye dönük biçimleri, yani yalnız nicemlenmiş (veya geçmiş) örnekleri, kullanarak çözümlenebilir [43]. Ancak, bu tür geriye doğru teknikler yalnız 10 kbit/san üzerinde başarılı olarak çalışabilir; çünkü LPC'nin doğruluğu, kodlanmış konuşmanın nicemleme gürültüsünün artışına paralel olarak süratle azalır.

Spektrum özellikleri yavaş değişen bölgelerde çerçeve çerçeve güncelleştirme uygun olabilir. Ancak, algılama için daha önemli olduğuna inanılan geçiş (*transition*) bölgelerinde çerçeve çerçeve güncelleştirme, geçişler çerçeveye dahil oldukça başarılı olmaz. Bu gibi durumlarda hesaplanan parametre kümesi, örneklerin yalnız o segmentteki spektral özelliklerinin ortalamasını temsil edecektir.

Yukarıda belirtilen sorunun üstesinden gelmek için popüler bir teknik, çerçeve aralarında aradeğerleme (*interpolation*) kullanmaktır. LPC çerçeveleri için aradeğerleme kullanımı, çerçeveler arasındaki parametre takımlarını değerlendirme yoluyla spektrumun daha iyi ifade edilmesini sağlar. Böylece kodlama kapasitesi yükseltilmeden ve yalnızca çok küçük bir gecikmeye katlanarak, çerçeve kenarındaki geçişler daha yumuşak olarak gerçekleştirilebilir. Şekil 6.28(a) bir aradeğerleme yapısı için göreceli zamansal yerleşimi yarım çerçeve gecikmesi eklenmiş olarak göstermektedir. Açıklama amacıyla, çerçeve dört alt çerçeveye bölünmüştür; bu, senteze göre analiz kullanan orta bit hızlı ile düşük bit hızlı LPC kodlama yapılarının büyük çoğunluğunda tipik özelliktir. Şekil 6.28(b), aynı yapıyı gecikmesiz olarak göstermektedir.

Yarım çerçevelik bir gecikmesi olan aradeğerlemeli LPC ile aradeğerlemesiz yapının toplam kodlama performansları objektif olarak çok benzerdir, ancak subjektif olarak aradeğerlemeli yapı ses olarak daha pürüzsüzdür.



Şekil 6.28 (a) yarım çerçeve aradeğerleme gecikmeli ve (b) gecikmesiz LPC analiz pencereleri

6.4.2 Uzun Dönemli Öngörü

STP ile karşılaştırıldığında LTP'nin daha az katsayısı bulunmaktadır ve genel biçimi (6.66) denklemi ile verilmektedir.

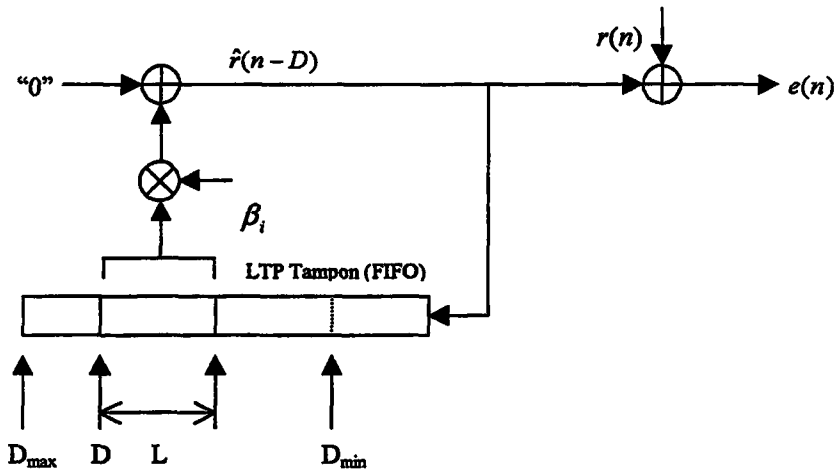
$$P(z) = 1 - \sum_{i=-I}^I \beta_i z^{(-D-i)} \quad (6.66)$$

CELP'teki LTP için kullanılan ifade, güncel perde uyarımasından olsun veya olmasın, uzun dönemli ortak ilişki oluşturacak biçimdedir. Bu nedenle, perde öngörücü yerine uzun süreli öngörücü (LTP) terimi tercih edilir, çünkü perde öngörücü ünsüz konuşmalar için, hatta bazı durumlarda ünlü konuşmalar için, bu filtrenin işlevini anlatmada yanlış anlaşılabilir. Ancak LTP, ünlü konuşmada en kullanışlı olanıdır, çünkü ünlü konuşma, bir perde süresiyle ayrılmış örnekler arasında önemli ölçüde ortak ilişkisi olan yaklaşık periyodik işaret olarak nitelenebilir.

CELP ve diğer AbS-LPC yapılarında LTP analizi, tek veya çok ara-uçlu gerek açık, gerek kapalı devre analizleri ile gerçekleştirilebilir [44]. Bu iki değişik analiz yöntemi aşağıda açıklanmıştır.

6.4.2.1 Açık Çevrim LTP

Açık devre yönteminde (*Open Loop Method-OLM*), orijinal konuşmanın LPC katsayıları ile ters filtrelenmesi yoluyla önce bir artık (*residual*) oluşturulur. Daha sonra bu artığın her alt çerçevesi üzerindeki gecikme ve kazanç terimleri saptanabilir.



Şekil 6.29 Değiştirilmiş açık devre (*Modified OLM*) LTP analizi blok diyagramı

OLM kullanılırken benimsenen genel uygulama, D gecikmesini LTP güncelleştirme boyu L 'den daha büyük tutmaktır. Bu sınırlama kullanıldığında hesaplanan gecikme ve kazanç değerleri daha optimumdur. Ancak, L büyük olduğu takdirde LTP'nin etkinliği azalır ki bu da sakıncalıdır, çünkü D , ünlü konuşmanın başlamasıyla aynı hızda uyarlanamaz.

L çok büyük değilse ve $D > L$ ise AbS-LPC yapılarında kullanılmak üzere daha iyi bir OLM tanımlanabilir. Şekil 6.29'da değiştirilmiş OLM (*Modified OLM-MOLM*) yapısı gösterilmektedir. Sarkma veya gecikme süresi, orijinal artık $r(n)$ 'ye en iyi uyan LTP sentezleyicisinin çıktısına, yani tek ara-uçlu LTP'ye göre belirlenir.

$$e(m) = r(n) - \hat{r}(n) \quad (6.67)$$

$$= r(n) - \beta \hat{r}(n - D) \quad (6.68)$$

Bu nedenle ortalama karesel hata E aşağıdaki gibi ifade edilebilir:

$$E = \sum_{n=0}^{L-1} e^2(n) = \sum_{n=0}^{L-1} [r(n) - \beta \hat{r}(n - D)]^2 \quad (6.69)$$

$$\frac{\partial E}{\partial \beta} = 2 \sum_{n=0}^{L-1} [r(n) - \beta \hat{r}(n - D)] \times -\hat{r}(n - D) = 0 \quad (6.70)$$

$$\beta = \frac{\sum_{n=0}^{L-1} r(n) \hat{r}(n - D)}{\sum_{n=0}^{L-1} \hat{r}^2(n - D)} \quad (6.71)$$

$$E = \sum_{n=0}^{L-1} r(n) - \frac{\left[\sum_{n=0}^{L-1} r(n) \hat{r}(n - D) \right]^2}{\sum_{n=0}^{L-1} \hat{r}^2(n - D)} \quad (6.72)$$

Bu nedenle, optimum D 'yi tespit etmek için gecikmenin değerleri D_{min} ve D_{max} arasında test edilir ve E hatasını en aza indiren gecikme optimum değeri olarak alınır. Çok ara-uçlu LTP için de benzer bir formül türetilir.

MOLM, normal OLM'dan daha iyi performans göstermektedir, çünkü gecikme ve kazanç katsayılarının saptanması için kullanılan değerler $r(n)$ 'nin nicemlenmemiş değerlerinden çok nicemlenmiş değerleridir. Bu şekilde, sentezleyicide önceki analiz alt çerçevelerindeki hatalardan kaynaklanan öngörü kazanç kaybı yoktur. Aslında MOLM, hatanın en aza indirilmesinin STP çıktısı yerine LTP çıktısında uygulandığı tam kapalı çevrim LTP'nin basit bir versiyonudur.

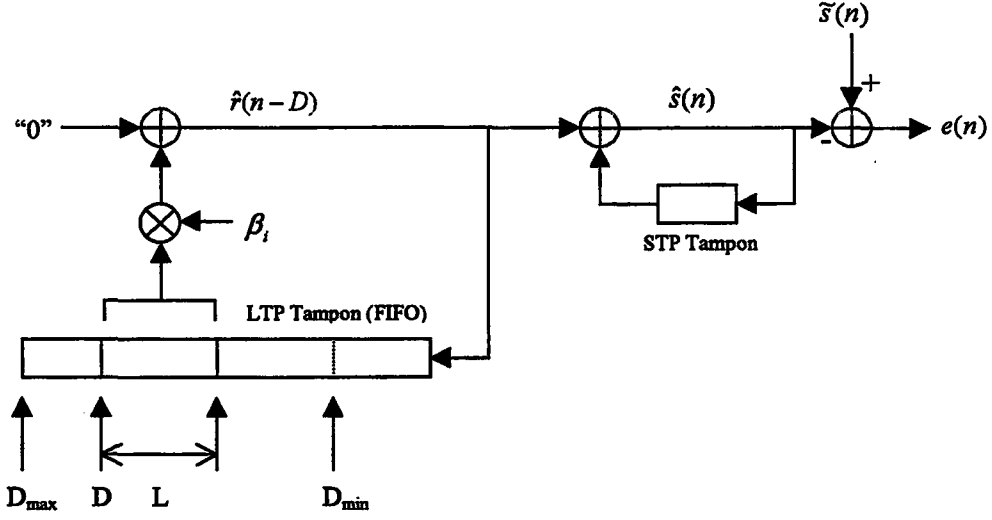
6.4.2.2 Kapalı Çevrim LTP

AbS kodlama yapılarında ağırlıklandırılmış orijinal konuşma ile sentezlenmiş çıktı arasındaki hatanın en aza indirilmesine çalışılır. Bu tanıma göre istenen LTP parametrelerini türetmek için yapılan işaret analizi, açık devre LTP'de olduğu gibi LTP öngörü hatasını (veya ikinci artığı) değil, ağırlıklandırılmış orijinal ile sentezlenmiş konuşma arasındaki hatayı asgariye indirmektedir. Esas amacın $e(n)$ 'in en aza indirilmesi olduğu Şekil 6.30'da gösterilmektedir. STP parametrelerinin hesaplanmış olduğu varsayımıyla kalan saptanmamış parametreler $Gx(n)$, D ve β_k 'dir. Her ne kadar bu parametreler, tüm $Gx(n)$ 'ler için olduğu gibi LTP parametrelerinin ayrıntılı aranması ile elde edilebilirse de yordam hesaplama yönünden çok yoğunlaşır ve dolayısıyla optimum olmayan çözümlerin kullanılması gereği doğar. Karmaşıklığı önlemenin bir yolu, LTP ve $Gx(n)$ 'nin ardışıl olarak iki aşamada bulunmasıdır. Önce $Gx(n)$ 'in sıfır olduğu varsayılır ve LTP parametreleri, $e(n)$ en az olacak şekilde hesaplanır. Daha sonra LTP sabit tutulur ve $Gx(n)$ hesaplanır.

$$x(n) = 0, \quad 0 \leq n \leq L-1 \quad (6.73)$$

$$\Rightarrow \hat{s}(n) = \sum_{i=-1}^I \beta_i \sum_{k=0}^n \hat{r}(n-k-D-i) h_w(k) \quad (6.74)$$

olsun, ki burada $h_w(k)$, STP filtresinin ağırlıklandırılmış dürtü cevabıdır.



Şekil 6.30 Kapalı Çevrim (Closed Loop-CL) LTP analizinin blok diyagramı

Ağırlıklandırılmış hata E_w ise (6.75)'teki gibi elde edilir:

$$E_w(D) = \sum_{n=0}^{L-1} e^2(n) = \sum_{n=0}^{L-1} (\hat{s}(n) - \hat{s}(n))^2 \quad (6.75)$$

Burada

$$\hat{s}(n) = s_w(n) - s_m(n) \quad (6.76)$$

olarak ifade edilebilir, ki ifadede $s_m(n)$, STP filtresinin bellek (veya sıfır girdi) katkısı ve $s_w(n)$, ağırlıklandırılmış orijinal konuşmadır. Bu nedenle:

$$\frac{\partial E_w}{\partial \beta_i} = 2 \left[\sum_{n=0}^{L-1} \hat{s}(n) - \sum_{i=-1}^I \beta_i \sum_{k=0}^n \hat{r}(n-k-D-i) h_w(k) \right] \quad (6.77)$$

$$x \left[- \sum_{k=0}^n \hat{r}(n-k-D-j)h_w(k) \right] = 0 \quad (6.78)$$

$$Z_i(n) = \sum_{k=0}^n \hat{r}(n-k-D-i)h_w(k) \quad (6.79)$$

$$\Rightarrow \sum_{n=0}^{L-1} \tilde{s}(n)Z_j(n) - \left[\sum_{i=-I}^I \beta_i \sum_{n=0}^{L-1} Z_i(n)Z_j(n) \right] = 0, \quad -I \leq j \leq I \quad (6.80)$$

olur ve dolayısıyla, 3 ara-uçlu filtre varsayılırsa matris olarak:

$$\begin{bmatrix} \beta_{-1} \\ \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \phi(-1,-1) & \phi(0,-1) & \phi(1,-1) \\ \phi(-1,0) & \phi(0,0) & \phi(1,0) \\ \phi(-1,1) & \phi(0,1) & \phi(1,1) \end{bmatrix} \quad (6.81)$$

ve tek ara uçlu filtre için, $I=0$ durumunda:

$$\beta_0 = \frac{B(0)}{\phi(0,0)} \quad (6.82)$$

yazılabilir, burada:

$$\phi(i,j) = \sum_{n=0}^{L-1} Z_i(n)Z_j(n) \quad (6.83)$$

$$B(i) = \sum_{n=0}^{L-1} \tilde{s}(n)Z_i(n) \quad (6.84)$$

LTP kazanç katsayıları bulunduğu (6.75) denkleminde yerine konulur ve $E_w(D)$ 'nin asgari olduğu gecikme D , optimum gecikme, D_{opt} , ve ilgili β_1 de optimum kazançlar olarak bulunur. Bu durumda $Gx(n)$ uyarması, D_{opt} ile β_{opt} sabit iken , bulunur. Ancak uygulamada D_{opt} , ikincil uyarma hesaplamasında olduğu gibi, tek

ara-uçlu filtrenin kazanç katsayısını hesaplamadan önce (kazanç nicemlemesinin analiz devresi dışında yapıldığı varsayılarak) bulunur. Böylece (6.75) denklemi şu şekilde yazılabilir:

$$E_w(D) = \sum_{n=0}^{L-1} \left[\tilde{s}(n) - \beta_0 \sum_{k=0}^n \hat{r}(n-k-D) h_w(k) \right]^2 \quad (6.85)$$

ve

$$\beta_0 = \frac{\sum_{n=0}^{L-1} \tilde{s}(n) Z_D(n)}{\sum_{n=0}^{L-1} Z_D^2(n)} \quad (6.86)$$

burada

$$Z_D(n) = \sum_{k=0}^n \hat{r}(n-k-D) h_w(k) \quad (6.87)$$

(6.86)'yı (6.85)'in yerine koyup tekrar düzenledikten sonra:

$$E_w(D) = \sum_{n=0}^{L-1} \tilde{s}^2(n) - \frac{\left[\sum_{n=0}^{L-1} \tilde{s}(n) Z_D(n) \right]^2}{\sum_{n=0}^{L-1} Z_D^2(n)} \quad (6.88)$$

Uygulamada, yukarıdaki ifadenin ikinci bölümü, denklem (6.86) kullanılarak ilişkili filtre kazanç katsayısı hesaplanmadan önce, optimum gecikme D_{opt} için maksimize edilir.

OLM'de olduğu gibi, D gecikmesi alt çerçeve L 'den küçük olduğu durumlarda CLM'de de sorunlar oluşur, yani LTP, aynı analiz alt çerçevesi içinde yinelenir [44]. Alt çerçeve boyundan küçük gecikmelerdeki kazanç ve gecikme katsayılarının

bulunmasındaki ana sorun, ortalama karesel hata ifadesinin $D < L$ için lineer olmamasıdır. Bir tek LTP katsayısının saptandığı durumu dikkate alınır ve LTP gecikmesinin de $L/2 \leq D \leq L-1$ aralığında bulunduğu varsayılırsa aşağıdaki ifadelerden biri geçerli olur:

$$\hat{r}(n) = \begin{cases} \beta \hat{r}(n-D) & 0 \leq n \leq D-1 \\ \beta^2 \hat{r}(n-2D) & D \leq n \leq L-1 \end{cases} \quad (6.89)$$

Ağırlıklandırılmış karesel hata E_w da aşağıdaki gibi ifade edilebilir:

$$E_w = \sum_{n=0}^{L-1} e^2(n) = \sum_{n=0}^{L-1} [\tilde{s}(n) - \hat{s}(n)]^2 \quad (6.90)$$

$$\frac{\partial E_w}{\partial \beta} = 2\beta^3 \sum_{n=D}^{L-1} Z_D^2(n) + \beta \left[\sum_{n=0}^{D-1} Z_D^2(n) - 2 \sum_{n=D}^{L-1} \tilde{s}(n) Z_D(n) \right] - \sum_{n=0}^{D-1} \tilde{s}(n) Z_D(n) = 0 \quad (6.91)$$

burada:

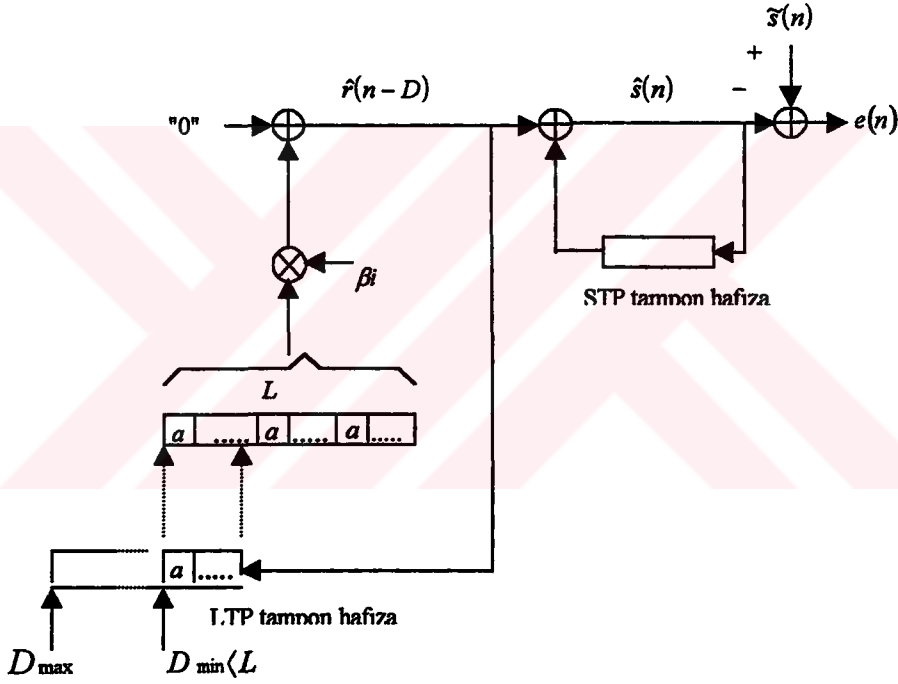
$$Z_D(n) = \sum_{k=0}^n \hat{r}(n-k-D) h_w(k) \quad (6.92)$$

Yukarıdan da görüleceği üzere E_w 'nin çözümü β 'nin kübünü içermektedir. Bu nedenle E_w 'yı çözmek, bu işlem L 'den düşük tüm D değerleri için yapılacağından oldukça fazla hesap yükü getirir. Bu sorunun çözümlerinden biri, β 'nin nicemlenmiş değerlerine dayanan bir deneme-yanılma yöntemidir. Bu yöntemde toplam terimleri önceden hesaplanır ve daha sonra β 'nin nicemlenmiş olası değerlerinden her biri denkleme yerleştirilir. En küçük karesel hatayı veren β değeri de istenen değer olarak belirlenir. Bu yöntemde bile gereken hesap miktarı oldukça fazladır (örneğin $L=50$, $D_{min}=20$ ve $\beta=3$ bit ise, arama sayısı $= 2^3 \times (50-20) = 240$).

$D < L$ için ikinci bir yöntem, geçmiş LTP filtre çıktılarının periyodik olarak tekrarlandığı LTP'nin adaptif kod kitabı ifadesini kullanmaktır [45], yani:

$$\hat{r}(n) = \begin{cases} \beta \hat{r}(n-D) & 0 \leq n \leq D-1 \\ \beta \hat{r}(n-2D) & D \leq n \leq 2D-1 \\ . & . \\ . & . \\ \beta \hat{r}(n-aD) & aD \leq n \leq L-1 \end{cases} \quad (6.93)$$

Böylece bu yöntem kullanıldığında yalnız β kere işlem yapılır. Bu yapı, otomatik olarak perde çiftlemesi (*doubling*) içerir ve alt çerçevedeki perde darbelerinin bir darbeden diğerine genlik değiştirmesine izin vermez. Bu adaptif yöntemi kullanan CELP sentezleyici Şekil 6.31'de gösterilmiştir.



Şekil 6.31 Optimizasyon aralığından daha küçük gecikmeler için perde tekrarını gösteren blok diyagram

6.4.2.3 Kesirli Gecikmeli LTP

Daha önceki iki LTP hesaplama yönteminde referans işaretinin uyumlaması, çapraz bir ilişki yordamı ile gerçekleştirilmiştir. Bunun ana kısıtlamalarından biri, işaretin

doğasında var olan örnekleme çözünürlüğüdür, yani çapraz ilişkinin etkin olabilmesi için referans ve yapay işaret arasındaki en iyi benzerlik anının yakalanmasını sağlayacak sürekli bir işaret gereklidir. Ancak, D gecikmeleri örnekleme hızının tamsayı değerleri ile kısıtlı olduğu için LTP, rastgele gecikme aralıkları ile başa çıkmak için tamsayı olmayan optimum gecikmeleri tamsayı değerler ile değiştirmek durumundadır. Bu da LTP'nin performansını objektif uyumlama yönünden bozucu bir faktördür.

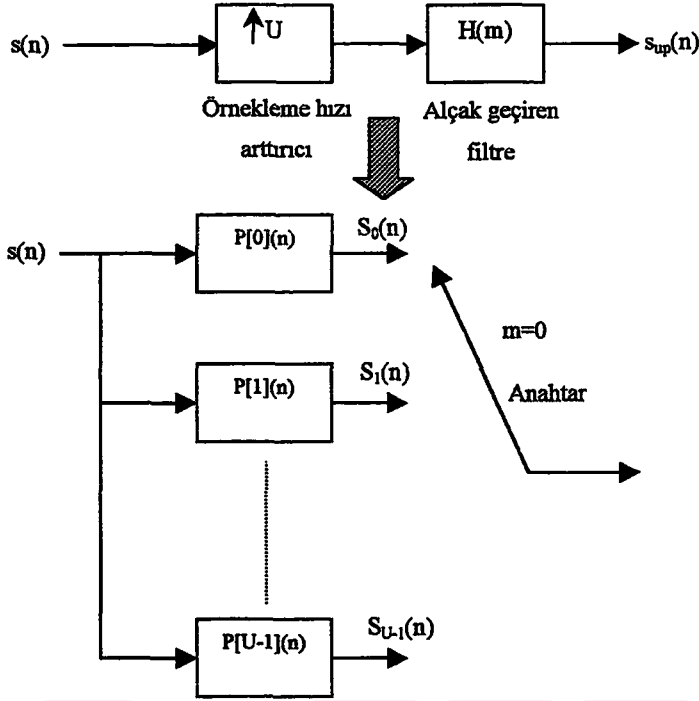
Önceki yöntemlerde daha yüksek dereceden LTP'ler incelenmişti. Gecikme değeri, tamsayı örneğe denk düşmediği durumlarda çoklu katsayılar örnekler arasında aradeğerlemeyi sağlayabilir. Ancak bunun sakıncası, kodlama kapasitesinin (hızının) artmasıdır. Buradan hareketle daha büyük bir LTP gecikme çözünürlüğü elde ederek kodlama kapasitesini de asgariye indirmek için çok-hızlı (*multi-rate*) bir yaklaşım kullanılabilir ve gerekenden daha hızlı örnekleme (*up-sampling*) kullanılır [46].

Hızlı Örnekleme ile Arttırılmış Çözünürlük:

LTP gecikmesi D , f_s örnekleme hızında alınan tamsayı miktarda örnek ile belirtilir. D 'nin yerine reel bir sayı koymaya çalışıldığında ayrık zaman işareti $s(n)$ 'i sürekli bir zaman işaretine çevirmek gerekir. $s(n)$ işareti, Nyquist hızına göre örneklenmiş olduğundan, herhangi bir zamanda nedensel olmayan bir alçak geçiren filtreden geçirilmek suretiyle sürekli $s_c(t)$ işareti elde edilebilir, yani:

$$s_c(t) = \sum_{k=-\infty}^{\infty} \frac{s(k) \sin(\pi(t-k))}{\pi(t-k)} \quad (6.94)$$

D 'nin ast katları ile ilgilenildiğinden, $s_c(t)$ işareti yerine daha yüksek hızla örneklenmiş bir işaret olan, $s_{up}(m)$ 'nin bulunması gerekir. İdeal alçak geçiren prototip filtre, sonlu uzunlukta bir filtre ile değiştirilir ve Şekil 6.32'deki gibi $s(n)$ işaretinden $s_{up}(m)$ 'nin elde edilir; burada U hızlı örnekleme faktörüdür ve çok fazlı (*polyphase*) bir yapı kullanılmıştır.



Şekil 6.32 Aradeğerlemenin gerçekleştirilmesi için uygulanan çok fazlı yapı

Tamsayı olmayan ve f_s örnekleme hızında alınmış bir gecikme değeri olan $(D+d/U)$ şimdi, tamsayı bir gecikmeye $(UD+d)$ denk düşmektedir (burada Uf_s hızında $d=0,1,...,U-1$). Dolayısıyla, tamsayı değerleri olmayan bir gecikme sağlamak için Şekil 6.30'da da gösterildiği gibi aradeğerleme filtresinin çok-fazlı yapısından uygun yol seçilir.

Aradeğerleme sürecinde, hem performans hem karmaşıklık yönünden alçak geçiren filtresinin seçimi çok önemlidir. *Kroon* [46] tarafından tavsiye edildiği şekilde basit fakat etkin bir seçim, Hamming penceresinin kullanılmasıdır. Hamming penceresinden geçirilmiş $\sin(x)/x$ fonksiyonunun kullanımı aşağıdaki üç sebepten dolayı faydalıdır:

- (1) Sonuç olarak ortaya çıkan FIR filtresi, lineer fazlı ve sabit gecikmelidir.
- (2) Filtre özellikleri, yalnız kısa filtre boyları için uygundur, yani örtüşen bileşenler küçüktür.

(3) Orijinal işaret, faz kayması olmadan elde edilebilir (Şekil 6.32'deki çoklu-faz yapısının en üst yolu). Yani, en üst yol sadece bir gecikme işlemi ise filtreleme işlemleri sayısı azaltılabilir.

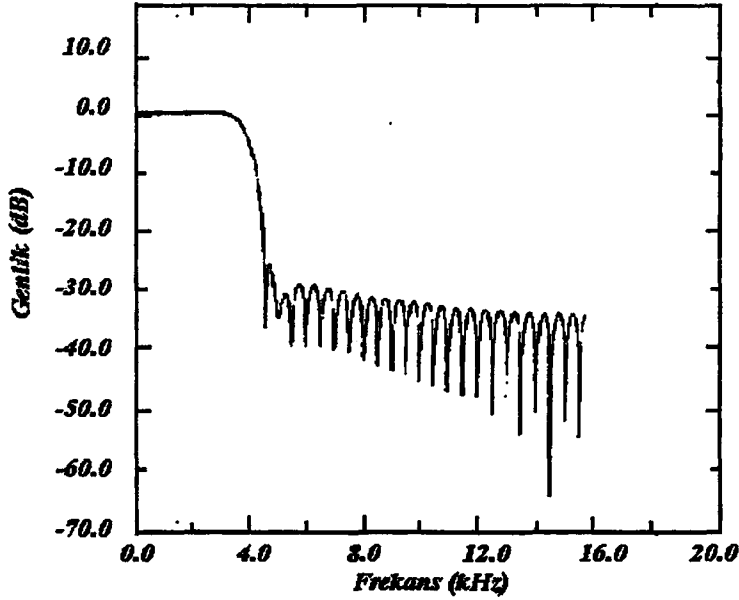
Yukarıda belirtilen faydaları elde etmek için filtre boyu N , Uf_s filtreleme hızında filtre gecikmesi, $(N-1)/2$, U 'nun tamsayı bir katı olacak şekilde seçilmelidir, yani:

$$N = 2IU + 1 \quad (6.95)$$

burada I, f_s örnekleme hızında alçak geçiren filtrenin gecikmesidir. $I = 8$ ve $U = 4$ için tipik bir filtre dürtü cevabı Tablo 6.1'de ve ilgili frekans cevabı Şekil 6.33'de verilmiştir.

Tablo 6.1 Aradeğerleme filtresinin dürtü cevabı

i	$H_f(i)$	i	$H_f(i)$	i	$H_f(i)$
0	1.000000000	11	0.062494423	22	-0.017205257
1	0.898382902	12	0.000000000	23	-0.010205095
2	0.631164074	13	-0.047242213	24	0.000000000
3	0.294341326	14	-0.058138002	25	0.007047138
4	0.000000000	15	-0.035739370	26	0.008219847
5	-0.170575693	16	0.000000000	27	0.004784813
6	-0.196243614	17	0.026833663	28	0.000000000
7	-0.115581349	18	0.032719128	29	-0.003293189
8	0.000000000	19	0.019861670	30	-0.003962609
9	0.083692148	20	0.000000000	31	-0.002463587
10	0.102025926	21	-0.014410716	32	0.000000000



Şekil 6.33 $N = 65$ iken alçak geçiren aradeğerleme filtresinin frekans cevabı

Hızlı Örnekleme Uygulaması:

Yukarıda anlatılan hızlı örnekleme gerek açık gerekse kapalı çevrim LTP'ye uygulanabilir. Ancak, LTP gecikmesi tamsayı ise, aday gecikme optimum aralıktan daha az olduğu takdirde filtre özyinelemeli olur. Burada tekrar "Perde Çiftleme" (*Pitch Doubling*) tekniği kullanılabilir. Daha iyi bir LTP gecikme tahmini, dolayısıyla daha iyi toplam SNR'in yanı sıra hızlı örnekleme başka bir fayda sağlar. Geçici çözünürlük daha yüksek olduğu için "perde çiftleme" bazı durumlarda azaltılabilir. Bunun sebebi, geçici çözünürlüğün daha yüksek olduğu durumlarda LTP'nin, birinci tizliği kaçıran tamsayı örnekleme etkisinden dolayı daha yüksek SNR verebilecek ikinci ve üçüncü perde sürelerini seçmesi gereksiniminin daha az olmasıdır. Ancak, bu etkinin güvenilir şekilde gözlenmesi zordur; çünkü hızlı örnekleme yalnız birinci tizliğe değil ikinci ve üçüncü tizliğe de daha iyi karar verir. Dolayısıyla, özellikle D gecikmesi ile karşılaştırıldığında optimizasyon aralığı küçükse, tizliklerin SNR yönünden de seçilmesi eşit olasılıktadır.

6.4.2.4 LTP Yöntemlerinin Performans Karşılaştırması:

Değişik LTP analiz metodlarının performansını belirlemek için nicemlenmemiş bir CELP ile değişik LTP'ler kullanılmıştır. Bu LTP'ler aşağıda verilmiştir:

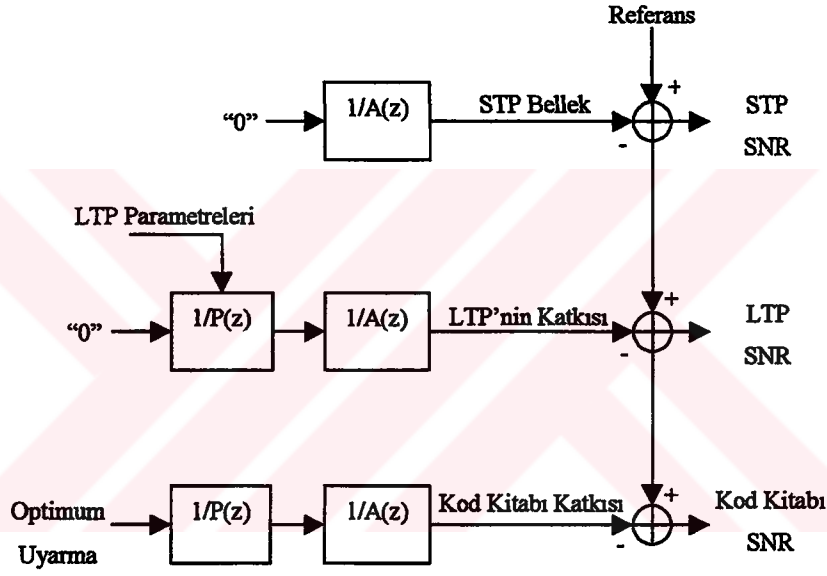
- (1) Değiştirilmiş Açık Çevrim (*Modified Open Loop-MOL*): 1 ara-uçlu (MOL1), 3 ara-uçlu (MOL3). (Bu, nicemlenmiş artığın nicemlenmemiş artık ile karşılaştırıldığı normal açık çevrim LTP analizinin değiştirilmiş biçimidir).
- (2) Hibrid Değiştirilmiş Açık Çevrim (*Hybrid MOL-HMOL*): 1 ara-uçlu (HMOL1), 3 ara-uçlu (HMOL3). (MOL1 ve MOL3'e benzer; yalnız gecikme MOL yöntemi ile fakat kazanç katsayıları β_k , kapalı çevrim yöntemi ile hesaplanmıştır).
- (3) Kapalı Çevrim (*Closed Loop-CL*) : 1 ara-uçlu (CL1), 3 ara-uçlu (CL3). (Bu D 'nin L 'den az olduğu adaptif kod kitabı bir kapalı çevrim analiz metodudur).
- (4) Değiştirilmiş CL3 (*Modified CL3-MCL3*): (Gecikme CL1 ile fakat kazanç katsayıları CL3 ile hesaplanmıştır).
- (5) Kesirli Gecikmeli Kapalı Çevrim: $U = 2$ (F2CL1), $U = 4$ (F4CL1). (Bu, temelde LTP'nin hızlı örneklediği CL1'dir).

CELP kodlayıcısının yapısı Tablo 6.2'de gösterilmiştir. LTP'nin performansını saptayabilmek için CELP kodlayıcısının toplam SNR'si üç bölüme ayrılır: (i) STP belleği katkısı, (ii) LTP katkısı ve (iii) Kod kitabı katkısı. Bunlar Şekil 6.34'de gösterilmiştir.

Tablo 6.3, Tablo 6.2'deki yapı kullanılarak değişik LTP yöntemleri arasındaki karşılaştırmanın sonucunu göstermektedir

Tablo 6.2 LTP karşılaştırma testi için CELP kodlayıcısı yapısı

<i>Parametre</i>	<i>Güncelleme</i>
Örnekleme sıklığı	8 kHz
STP: mertebesi=10	160 örnek
LTP: muhtelif	40 örnek
Kod kitabı: 10 bit	40 örnek
Ağırlıklandırma	$\gamma = 1$



Şekil 6.34 Toplam SNR'ye katkıda bulunan CELP alt sistemlerinin açılımı

Tablo 6.3 Değişik LTP'lerin SegSNR (bölütlenmiş SNR) değerlerinin açılımı

<i>Yapı</i>	<i>Gecikme</i>	<i>STP (dB)</i>	<i>LTP (dB)</i>	<i>KodKitabı (dB)</i>	<i>Toplam (dB)</i>
MOL1	41-168	1.60	3.39	4.95	9.95
HMOL1	41-168	1.69	5.23	3.53	10.45
MOL3	42-167	1.72	4.66	4.79	11.17
HMOL3	42-167	1.65	6.44	3.02	11.11
CL1	21-148	1.76	7.28	2.74	11.77
CL3	22-147	1.81	9.13	2.33	13.26
MCL3	22-147	1.77	8.98	2.42	13.16
F2CL1	21-148	1.79	7.37	2.79	11.95
F4CL1	21-148	1.80	7.68	2.76	12.24

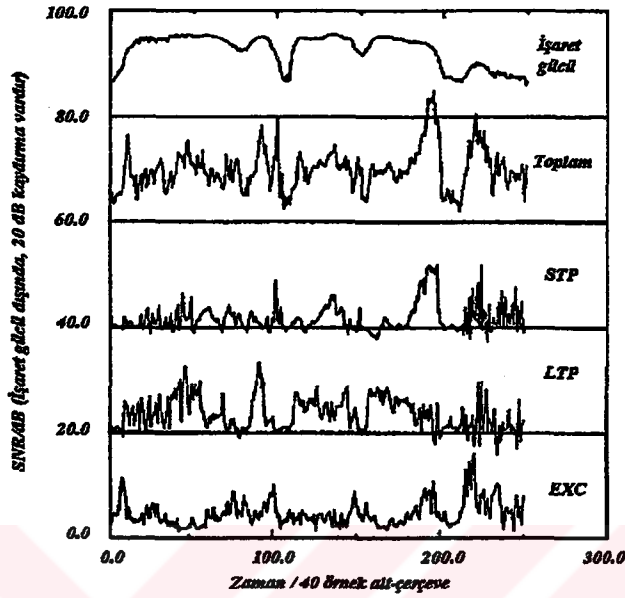
Karşılaştırma testi ve Tablo 6.3'de verilen bölütlenmiş SNR değerlerinden şu ilginç sonuçlara varılabilir:

(1) Beklendiği üzere STP belleğinin katkısı az çok sabittir ve 1.7 dB civarındadır.

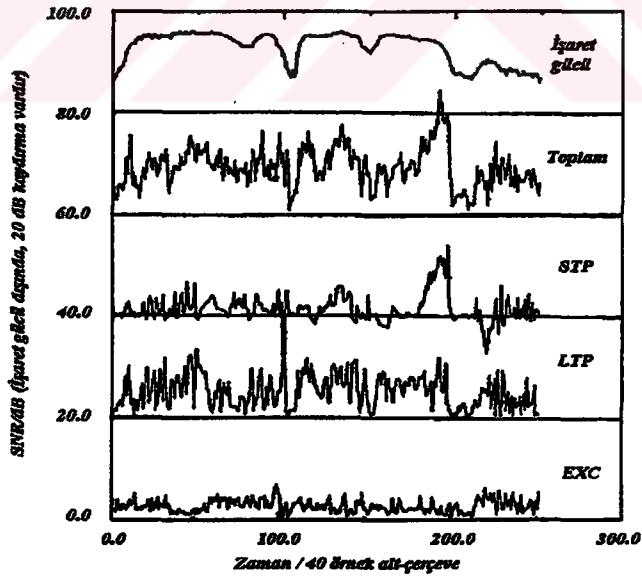
STP'nin SNR değerleri grafiği Şekil 6.35, 6.36 ve 6.37'de gösterilmiştir. STP katkısının toplam üzerinde her zaman pozitif etkisi olmadığı SNR'daki dalgalanmadan görülmektedir, yani bellek kodlama sürecinin kalanını aslında zorlaştırmaktadır. Bu, kodlama sürecinin geçmişini açıkça yansıtır, yani önceki alt çerçevede uyumlama zayıf ise, muhtemelen STP belleğinin içeriği de uygun olmayacaktır.

(2) Açık çevrim yapıları (MOL1 ve MOL3) için LTP ve kod kitabının katkıları kabaca eşittir. Ancak Şekil 6.35'e bakıldığında, kod kitabı katkısının her zaman pozitif olmasına karşın (hata uyumlama kriterinin sonucu olarak), zaman zaman görünen çok büyük negatif SNR'lerin de gösterdiği gibi, LTP katkısı her zaman arzu edilmemektedir.

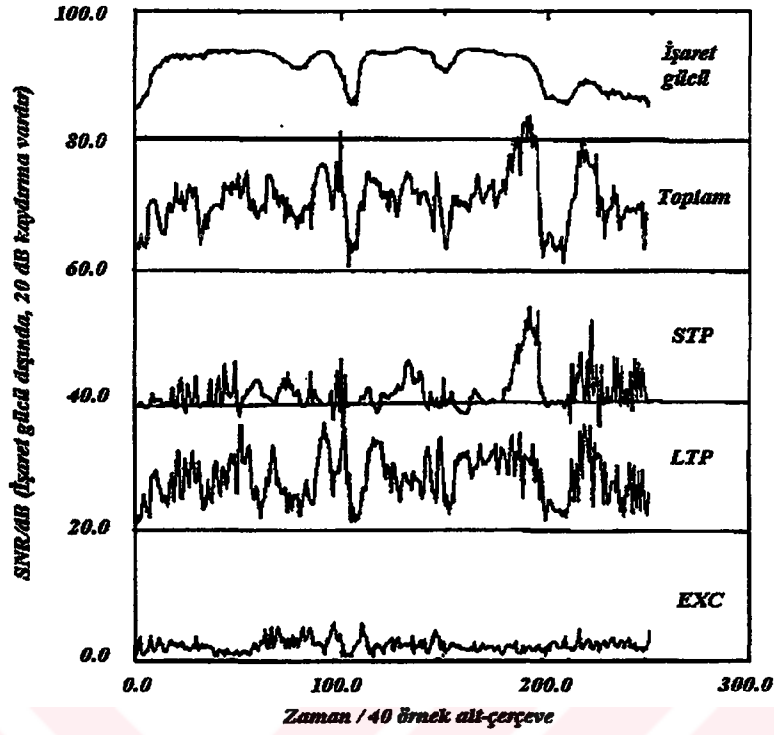
- (3) Hibrid MOL1 ve MOL3 için her ne kadar toplam SNR artmasa da değişik bir davranış gözlenmektedir. β 'yı hesaplamak için kapalı çevrim yöntemi kullanıldığında, Şekil 6.36'dan görüleceği gibi LTP katkısı her zaman pozitifdir.



Şekil 6.35 MOL3-LTP yapısı için SNR değerlerinin açılımı



Şekil 6.36 HMOL3-LTP yapısı için SNR değerlerinin açılımı



Şekil 6.37 CL3-LTP yapısı için SNR değerlerinin açılımı

- (4) Kesirli gecikmeli kapalı çevrim LTP'leri objektif olarak normal CL-LTP'lerden daha iyi performans gösterir. Ancak, subjektif olarak sağlanan iyileşme, SNR'nin gösterdiğinden daha somuttur. Örneğin özellikle konuşanın kadın olması halinde konuşma daha da berraklaşır. Burada yalnızca gecikmelerin düzgün dağıldığı durum incelenmiştir. Düzgün olmayan dağılım [46]'da belirtildiği gibi kullanılırsa performans daha da artırılabilir.
- (5) Tablo 6.3'den görülebileceği gibi kapalı çevrim LTP'ler, açık çevrim yöntemlerinden görülebilir ölçüde daha yüksek SNR sağlar. Ayrıca, MOL1 ve MOL3 durumlarında LTP ve kod kitabı katkıları kabaca eşit olmalarına karşın, CL-LTP durumlarında LTP kazancı önemli derecede artarken kod kitabı kazancı, Şekil 6.37'den de görülebileceği gibi azalmıştır. Bunun bir açıklaması CL-LTP'de LTP parametrelerinin optimum bir yordam ile hesaplandığı ve böylece bilginin çoğunun LTP tarafından zaten üretilmesidir. Bu nedenle geriye kalan eksik bilginin sağlanması için 10 bitlik kod kitabı, toplam SNR'de yalnız belirli bir

düzey sağlayabilir. Daha fazla katkı elde etmek için LTP ve kod kitabı beraberce optimize edilebilir [47].

Genel olarak öznel (*subjective*) dinleme testleri, CL-LTP yapılarının OL-LTP yapılarından üstün olduğunu göstermiştir. Sonuçta konuşma daha az kaba ve fondaki "tıslama" daha az rahatsız edici olmuştur. Her ne kadar CL-LTP'ler için farklar OL-LTP'lerinki bariz değilse de 3 ara-uçlu LTP'ler 1 ara-uçlulardan daha iyidir.

6.4.3 İkincil Uyarmanın Hesaplanması

Uyarma kod kitabında bulunan vektörler, CELP kodlama algoritmasının çok önemli bir parçasıdır. Bunların iki ana amacı bulunmaktadır:

- (1) LTP belleğine başlama bilgisi sağlar, ayrıca LTP tarafından gereğince takip edilemeyen konuşmanın ani değişimleri aynı işlemi yapar.
- (2) Özellikle ünsüz konuşma bölgelerinde LTP'nin atladığı noktalar için "doldurma" bilgisini sağlar.

LTP ile ilgili olarak daha önceki bölümde görüldüğü gibi, analiz için özellikle açık çevrim yöntemleri kullanılırsa kod kitabı uyarmasının kazanca oldukça önemli katkısı olabilir. Bir CELP kodlama algoritmasının kod kitabının nasıl dolduğu ve optimum vektörün nasıl hesaplandığı çok önemlidir. Başka bir önemli konu da, kod kitabı arama yordamının hesap yükü ve saklanmasıdır. Daha önce belirtildiği gibi tek ölçekleme faktörlü kod kitabı için CELP içindeki en iyi vektörün aranması, E_k 'nın $k = 1, \dots, C$ için (C , (6.36) denkleminde kod kitabındaki seviye sayısı) en aza indirilmesi yada (6.36) denklemindeki ikinci terimin maksimize edilmesini ifade etmektedir.

(6.36) denklemi incelendiğinde E_k 'yı değerlendirmek için aşağıdaki gibi 4 farklı hesaplamanın gerekli olduğu görülür:

- (1) $\hat{s}_k(n)$ çıktısını elde etmek için kod kitabı vektörünün sentezi;
- (2) Referans $\tilde{s}_k(n)$ (STP ve LTP etkileri çıkarılmış olarak) ile yapay öngörü $\hat{s}_k(n)$ arasındaki çapraz ilişkinin hesaplanması;
- (3) Yapay öngörü $\hat{s}_k(n)$ 'in özilişkisinin hesaplanması;
- (4) E_k hatasının minimum ve (6.36)'nın ikinci teriminin maksimum olup olmadığının test edilmesi.

Karmaşıklığı ve gereken bellek kapasitesini azaltma, ayrıca kalitenin iyileştirilmesi sebeplerinden dolayı ikincil uyarmanın pek çok tipi geliştirilmiş ve kullanılmıştır. Burada üç değişik ikincil uyarma tipi ele alınacaktır.

6.4.3.1 Standart (*Gauss* tipi) Uyarma:

CELP'in başlangıçta kullanılan biçimlerinin hemen tümünde, ikincil uyarma kaynağı için *Gauss* tipi kod kitabı yapısı kullanılmıştır. Bunun ana nedeni, STP ile ters filtreleme ve takip eden LTP'den sonra konuşmanın örnek dağılımının *Gauss* dağılımına benzemesidir. Bu yaklaşımın bir sorunu, *Gauss* kod kitabı vektörlerini depolamak için gereken bellek miktarıdır. Örneğin, L referans örneklerine uygun 10 bitlik kod kitabı kullanıldığı takdirde depolama yeri sayısı $2^{10} \times L$ olacaktır. $L = 40$ varsayılırsa, bu, 32 bitlik bir kayan nokta gerçeklemesi için çok büyük sayılabilecek 40.960 depolama yeri demektir. Bu sorunun üstesinden gelmek için, *Gauss* vektörleri, birbirini takip eden iki vektörün L örneklerinin ortak olduğu tek boyutlu bir dizi olarak ifade edilir. Üst üste binen bu tip kod kitaplarının en popüler biçimleri tek ve iki kaydırmalı (*shift*) olanlarıdır. Diğer bir deyişle, bir vektörden diğerine giderken vektörün sonundaki bir veya iki örnek atılır ve bir veya iki kaydırmalı vektörün başına, sırasıyla bir veya iki yeni örnek eklenir. Tek kaydırmalı, üst üste binen bir kod kitabı, en az ve en fazla gecikmeleri, bellek yerlerinin sıfırdan başladığı varsayımıyla, sırasıyla, $L-1$ ve $C+L-1$ olan bir LTP filtresi olarak temsil edilebilir.

Bazı kodlayıcılarda, birim varyans vektörü için 1.2'lik bir kırpma eşiği bulunan merkezden kırılmış (*centre clipped*) bir vektör kullanılır. Bu vektörün daha keskin konuşma çıktısı verdiği gözlenmiştir. Bunun sebeplerinden en önemlisi, kod kitabı vektörleri referans vektörü ile karşılaştırılırken, birkaç yüksek genlikli vektörün seçimde öne çıkması ve daha düşük büyüklükteki vektör elemanlarında hataya neden olmasıdır.

Kod kitabı boyu 8 bit veya üzeri ise *Gauss* kod kitaplarının eğitilmesi kaliteyi görülür derecede iyileştirmez. Bu nedenle bunlar, bir *Gauss* rastgele sayı üretici kullanılarak doldurulabilir. Bazı uygulamalarda, her *Gauss* sayı genliğinin pozitif ise 1 yapıldığı, sıfır ise sıfır bırakıldığı, negatif ise -1 yapıldığı üçlü kod kitapları kullanılır. Bu özellikle 16 bitlik sabit noktalı uygulamalar için kullanışlıdır. Yukarıdaki biçimlere ek olarak, standart kod kitabının popüler bir versiyonu, sıfır olmayan her vektör elemanının sabit sayıda sıfır tarafından takibedildiği "dağınık kod kitabı" adıyla anılır. Bu yaklaşım, düzenli darbe ile uyarılan LPC'ye (RPELPC) çok benzer.

Üstüste binen kod kitapları, kod kitabı arama hesap yükünü azaltırken aynı zamanda daha az bellek yerine ihtiyaç duydukları için kullanışlıdır. $h_w(n)$ 'nin ağırlıklandırılmış STP dürtü cevabını ifade ettiği hallerde, tek kaydırmalı bir kod kitabının k ıncı uyarma vektörü $x_k(n)$ ile sentezlenmiş vektörü $s_k(n)$ aşağıdaki gibi ifade edilebilir:

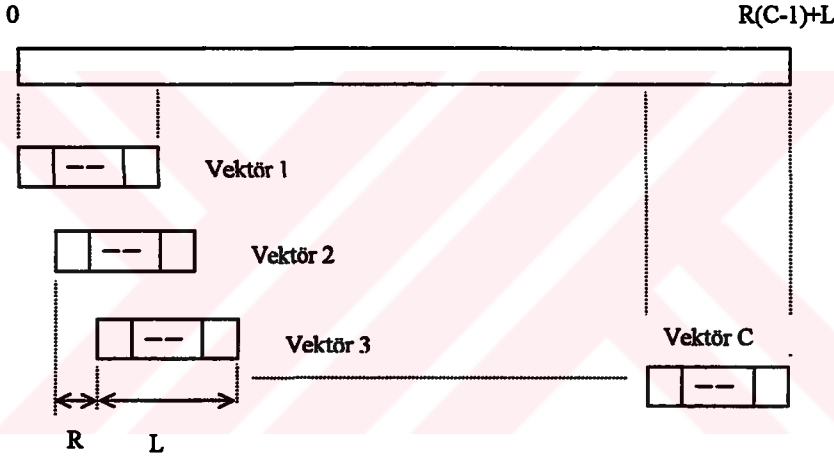
$$\hat{s}(n) = \sum_{i=0}^n x_k(n-i)h_w(i) \quad (6.96)$$

(Not: LTP, $D < L$ için adaptif bir kod kitabı olarak uygulandığında sabit kod kitabı ile paralel olduğu varsayılabilir ve dolayısıyla sabit kod kitabı vektörleri, STP filtrelerine doğrudan girdi olarak kullanılabilir.) Tek kaydırmalı bir kod kitabında, birbirini takip eden iki vektör arasındaki fark iki vektörün herhangi bir tarafında yalnız bir örnek olduğu için, sentezlenmiş vektör $\hat{s}_{k+1}$, $\hat{s}_k$ cinsinden aşağıdaki gibi yazılabilir:

$$\hat{s}_{k+1}(n) = x_{k+1}(0)h_w(n) + \hat{s}_k(n-1) \quad (6.97)$$

burada $s_k(-1)=0$. Yukarıdaki ifadeden görülebileceği üzere, önceki çıktıyı bir örnek boyu sağa kaydırıp, bunu yeni örnek tarafından ölçeklenmiş STP'nin dürtü cevabına ekleyerek konvolüsyon hesaplamalarının çoğu basitleştirilebilir. Ancak, kod kitabındaki kaydırmalar çoğaldıkça karmaşıklık çoğalır ve (6.98) genel ifadesinden de görülebileceği gibi kaydırma R vektör boyuna eşitlendiğinde üst üste binen kod kitabı, bağımsız vektörler içeren standart bir kod kitabına dönüşür.

$$\hat{s}_{k+1}(n) = \begin{cases} \sum_{i=0}^n x_{k+1}(n-i)h_w(i) & 0 \leq n \leq R \\ \sum_{i=0}^{R-1} x_{k+1}(i)h_w(n-i) + \hat{s}_k(n-R) & n \geq R \end{cases} \quad (6.98)$$



Şekil 6.38 Tipik bir üst üste binen kod kitabı yapısı

Tipik üst üste binen bir kod kitabı yapısı Şekil 6.36'da gösterilmiştir. Tek kaydırmalı ($R=1$) üst üste binen kod kitabının basitleştirilmiş arama metodu Şekil 6.37'de görülmektedir. Merkezi kırpma kullanılması durumunda (6.96)'daki $x_k(n)$ 'nin sıfır değerlerinin, $h_w(i)$ ile çarpımı gerekmeyecek ve böylece hesap karmaşıklığı azalacaktır. Ancak, birinci vektör tamamen sentezlenince (6.97) denklemini kullanmak daha çok yarar sağlayacaktır. Sıfır değerli bir uyarma örneğinin yeni bir vektör üretimine her katılışında (6.97) denkleminin ilk süresi sıfır olacaktır ve bu yeni yapay vektör, bir önceki vektörün ilk örneği sıfırlanmış halinin basitçe kaydırılmış biçimidir.

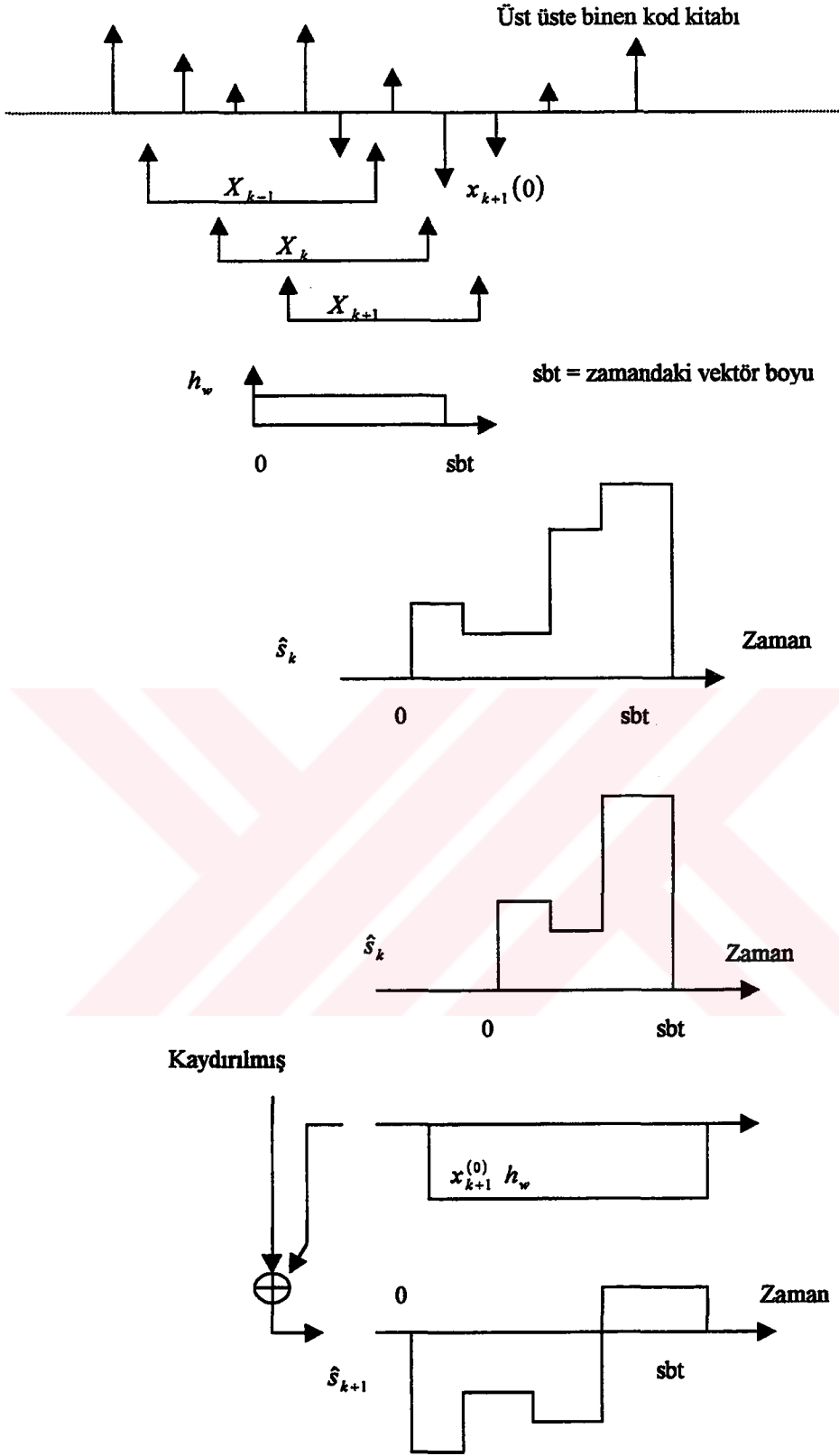
Tablo 6.4'te tanımlandığı biçimde nicemlenmemiş bir CELP kodlayıcısı kullanarak, standart *Gauss* kod kitabının dört değişik biçimi karşılaştırılmıştır. Sonuçlar Tablo 6.5'te gösterilmiştir.

Tablo 6.4 Karşılaştırma testi için CELP kodlayıcı parametreleri tanımı

<i>Parametre</i>	<i>Güncelleme</i>
Örnekleme	8 kHz
STP çerçevesi	160 örnek
LTP 1&3 ara uç	40 örnek
10 bit Kod kitabı	40 örnek
Ağırlıklandırma	$\gamma = 0.9$

Tablo 6.5 Dört-standard kod kitabının performansları

<i>Yapı</i>	<i>1 ara-uçlu</i>	<i>3 ara-uçlu</i>	<i>Bellek (Word)</i>
G	11.11	12.52	1024x40
CCG	11.20	12.53	1024x40
OG	11.16	12.49	(2x1023)+40
OCCG	11.18	12.55	(2x1023)+40



Şekil 6.39 Tek kaydırmalı üst üste binen kod kitabı araması sistematığı

Tablo 6.5'te G, CCG, OG ve OCCG, sırasıyla *Gauss* tipi, merkezi kırpmalı *Gauss* tipi, üst üste binen *Gauss* tipi ve üst üste binen merkezi kırpmalı *Gauss* tipidir ve üst üste binen kod kitapları için $R=2$ alınmıştır.

Subjektif dinleme testleri, merkezi kırpma ile konuşma kalitesinin normal *Gauss* kod kitaplarına göre daha iyi olduğunu göstermektedir. Ancak aynı tipteki üst üste binmiş kod kitapları ile üst üste binmemiş kod kitapları arasındaki fark gözardı edilebilir. Tablo 6.5'teki cazip sonuçlara göre OCCG, düşük bellek ve hesaplama gereksinimi ile oldukça başarılı bir yöntem olarak göze çarpmaktadır.

6.4.3.2 Vektör Toplamı Uyarma (VSELP):

CELP'in normal filtreleme yaklaşımında $\tilde{s}(n)$, sonlu sayıda $\hat{s}(n)$ dizisi ile detaylı bir işlem sonucunda eşleştirilir ve en iyi eşleşme $\hat{s}_{opt}(n)$ en az ortalama karesel hataya göre seçilir. $\tilde{s}(n)$ ile $\hat{s}(n)$ arasındaki eşleşmenin başarımı, $\hat{s}(n)$ 'in serbesti derecesine yani kod kitabının özellik ve boyuna göre saptanır. Daha önce açıklanan yöntemde $\hat{s}(n)$ 'in serbestisi, $\hat{s}(n)$ 'in pek çok biçiminin sentezlenmesi yoluya elde edilmiştir, yani $\hat{s}(n)$ 'in serbesti derecesi, analizin artık tarafındaki $x(n)$ ile sınırlıdır. Ancak, analizin yapay işaret tarafında aynı derecede serbesti sağlanırken tüm aday $\hat{s}(n)$ 'lerin LPC filtreleri tarafından spektrum biçimine sokulması halinde daha az karmaşıklık ve daha çok performans elde edilebileceği dikkate alınmalıdır. Bu nedenle amaç, sentez işlemlerinin sayısını sınırlamak ve analizin çıktı tarafında $\hat{s}(n)$ 'e gereken serbestiyi vermek için uygun vektör kombinasyonlarını gerçekleştirmektir. Bunun için kullanılan yöntemlerden biri de Vektör Toplamı Uyarımalı Doğrusal Öngörülü Kodlama (VSELP)'dir [21].

Konuşma kodlama analizi yöntemlerinin bir çoğunda olduğu gibi, VSELP'de de ortalama karesel hata tahmini kullanılmaktadır. Optimum uyarma $x_{opt}(n)$ ve dolayısıyla $\hat{s}_{opt}(n)$ 'i türetmek için VSELP uyarması aşağıdaki gibi elde edilir.

Her aday sentezlenmiş işaret $\hat{s}(n)$ aşağıdaki gibi ifade edilebilir:

$$\hat{s}(n) = \sum_{i=1}^M a_i \hat{s}_i(n), n = 0, 1, \dots, L-1 \quad (6.99)$$

burada $\hat{s}(n)$ dizileri, STP filtrelerini M değişik $x_i(n)$ uyarma dizisi ile uyararak türetilmiş, a_i 'ler değişken ölçekleme faktörleri, L uyarma alt çerçeve boyu ve $M < L$ seçilirse de $\tilde{s}(n)$ 'e daha düzgün bir yaklaşım mümkün olur. $\hat{s}(n)$ ile $\tilde{s}(n)$ arasındaki (6.100)'de verilen ağırlıklandırılmış ortalama karesel hatayı en aza indirgeyen optimum sentezlenmiş işaret $\hat{s}_{opt}(n)$, bir katsayı takımı a_i ($i=1, 2, \dots, M$) seçilerek türetilir.

$$E = \sum_{n=0}^{L-1} [\tilde{s}(n) - \hat{s}(n)]^2 = \sum_{n=0}^{L-1} \left[\tilde{s}(n) - \sum_{i=1}^M a_i \hat{s}_i(n) \right]^2 \quad (6.100)$$

(6.100) denklemi türetilerek, sıfıra eşitlenirse:

$$\frac{\partial}{\partial a_1} \left[\sum_{n=0}^{L-1} \left[\tilde{s}(n) - \sum_{i=1}^M a_i \hat{s}_i(n) \right]^2 \right] = 0 \quad (6.101)$$

elde edilir ve bu denklem de aşağıdaki şekilde basitleştirilebilir:

$$\sum_{i=1}^M a_i R(i, j) = \phi(j), j = 1, 2, \dots, M \quad (6.102)$$

Burada,

$$R(i, j) = \sum_{n=0}^{L-1} \hat{s}_i(n) \hat{s}_j(n) \quad (6.103)$$

ve

$$\mathcal{O}(j) = \sum_{n=0}^{L-1} \tilde{s}(n) \hat{s}_j(n) \quad (6.104)$$

olarak ifade edilebilir. $\hat{s}_j(n)$ vektörlerinin bağımsız ve ortogonal kılındığı varsayılırsa, böylece $\bar{s}_j(n)$ olarak yazılabilir ve:

$$R(i, j) = \sum_{n=0}^{L-1} \bar{s}_i(n) \bar{s}_j(n) \begin{cases} = 0 & i \neq j \\ \neq 0 & i = j \end{cases} \quad (6.105)$$

olur. (6.105), (6.102)'nin yerine koyularak ve tekrar düzenlenerek, $\{\hat{a}_i\}$ şu şekilde hesaplanabilir:

$$\hat{a}_i = \mathcal{O}(i) / R(i, i), \quad i = 1, 2, \dots, M \quad (6.106)$$

Böylece $\tilde{s}(n)$ için en iyi öngörü aşağıdaki şekilde verilebilir:

$$\hat{s}_{opt}(n) = \sum_{i=1}^M \hat{a}_i \bar{s}_i(n), \quad n = 0, 1, \dots, L-1 \quad (6.107)$$

Yukarıdaki yöntemde, karesel ortalama hatanın minimumu için en uygun çözümün elde edilebilmesi için, dizi kümesi $\hat{s}_i(n)$ 'lerin lineer olarak bağımsız (bir temel teşkil eden) ve birbirine ortogonal olmaları gerekir. Ortogonallığı sağlamak için elverişli bir yöntem, aşağıda özetlenen *Gram-Schmidt* yaklaşımıdır.

Her biri L boyunda $m+1$ tane $p_i(n)$ vektörlerden oluşan, temel teşkil edecek bir küme ele alınsın. Amaç, ortogonal $q_i(n)$ vektörlerini elde etmek olduğuna göre:

$$\sum_{n=0}^{L-1} q_i(n) q_j(n) \begin{cases} = 0 & i \neq j \\ \neq 0 & i = j \end{cases} \quad i, j = 0, 1, \dots, m \quad (6.108)$$

$q_0(n) = p_0(n)$ olsun ve $q_1(n)$, $q_0(n)$ ile $p_1(n)$ 'in bir birleşimi olarak tanımlansın; bu durumda:

$$q_1(n) = p_1(n) - \alpha_{01} q_0(n) \quad (6.109)$$

yazılabilir ve $q_1(n)$ 'in $q_0(n)$ 'e ortogonal olması için:

$$\sum_{n=0}^{L-1} q_1(n) q_0(n) = 0 \quad (6.110)$$

olması gerekir. Yani:

$$\sum_{n=0}^{L-1} p_1(n) q_0(n) - \sum_{n=0}^{L-1} \alpha_{01} q_0^2(n) = 0 \quad (6.111)$$

ve

$$\alpha_{01} = \frac{\sum_{n=0}^{L-1} p_1(n) q_0(n)}{\sum_{n=0}^{L-1} q_0^2(n)} \quad (6.112)$$

Bu q_{01} ile, her iki $q_0(n)$ ve $q_1(n)$ fonksiyonu ortogonal olur. Diğer $q_l(n)$ fonksiyonlarını elde etmek için lineer bağımsız olan $p_l(n)$, hepsi oluşturuluncaya kadar teker teker eklenir. Bu genel olarak aşağıdaki gibi ifade edilebilir:

$$q_k(n) = p_k(n) - \sum_{j=0}^{k-1} \alpha_{jk} q_j(n) \quad (6.113)$$

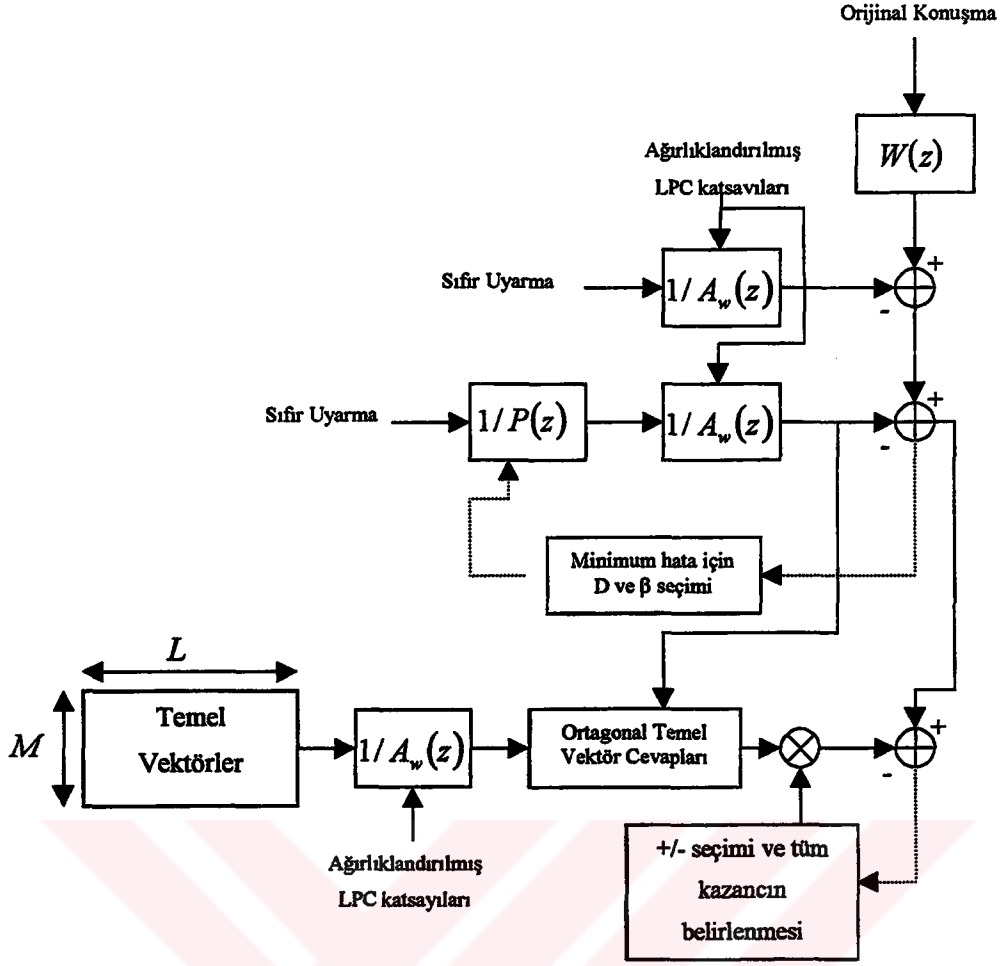
Burada,

$$\alpha_{jk} = \frac{\sum_{n=0}^{L-1} p_k(n) q_j(n)}{\sum_{n=0}^{L-1} q_j^2(n)}, \quad j = 0, 1, \dots, k-1 \quad (6.114)$$

Dolayısıyla, VSELP'in temel görevi, ortogonal vektörler $\bar{s}_i(n)$ ve bunların $\hat{a}_i$ optimum ölçekleme faktörlerini türeterek, $\hat{s}_{opt}(n)$ 'i elde etmektir. $x_{opt}(n)$, optimum uyarma daha sonra, $\hat{s}_{opt}(n)$ 'i ters STP filtresinden geçirerek bulunabilir.

VSELP'de temel vektörlerin algılanabilir olarak avantajlı biçimde elde edilmesi çok önemlidir. Bunun için yapının eğitilerek kod kitabının iyileştirilmesi gerekir. *Gauss* uyarmalı CELP'te olduğu gibi LTP, VSELP kodlayıcıda alt çerçeve boyundan küçük LTP sarkma değerleri için adaptif bir kod kitabı gibi çalışır ve dolayısıyla, her ne kadar LTP analizi ikincil uyarma hesaplamasından evvel yapılsa da ikinci uyarma analizi süresinde yalnız STP filtresinin etkisi dikkate alınır. Sonuç olarak toplam STP uyarması, kazanca göre ölçeklenmiş ikincil uyarma $Gx_{opt}(n)$ 'nin LTP uyarmasına eklenmesiyle bulunur.

VSELP kodlayıcının blok diyagramı Şekil 6.40'da gösterilmiştir. Burada STP ve LTP filtre parametreleri standart CELP'teki gibi hesaplanmaktadır. Uyarmanın bulunması için M adet temel vektör $[x_i(n)]_{i=1}^M$, M adet yapay temel vektörün $[\hat{s}_i(n)]_{i=1}^M$ oluşturulması için önce sentezlenir. Bunlar daha sonra *Gram-Schmidt* ortogonal bulma yordamı vasıtasıyla LTP katkı işaretine ortogonal hale getirilir ve $[\hat{s}_i(n)]_{i=1}^M$ elde edilir. Bu, ikincil uyarmanın LTP katkısı tarafından zaten kapsanan bölgenin yeniden ele alınmamasını sağlar. Ortogonalleştirmeden sonra her temel vektör ölçekleme faktörleri a_i , *Minci* element vektörü a 'yı oluşturmak üzere hesaplanır. a vektörü daha sonra $\pm$ 'ler ile nicemlenir. Uyarmanın nihai yapay işareti $\hat{s}_{opt}(n)$, önce uygun işaretlenmiş ($\pm$) ortogonal temel vektörlerinin toplanması, daha sonra toplanmış vektörlerin kazanç olarak ölçeklenmesi ile elde edilir. Nicemlenmiş ikincil uyarma $\hat{s}_{opt}(n)$ 'yi elde etmek için daha sonra ters filtrelenir. Toplam ölçekleme faktörü standart CELP'te olduğu gibi bulunur. Sonuçta ortaya çıkan yapay konuşma, ölçeklenmiş ikincil uyarma ile LTP katkılarının toplanması ve bunların ilk bellek değerleri geri yüklenmiş STP'den geçirerek sentezlenmesi ile elde edilir. Ortogonalleştirme yordamı ile ters filtreleme hem kodlayıcı hem kod çözücü için gereklidir.



Şekil 6.40 VSELP kodlayıcı blok diyagramı

6.4.3.3 Adaptif Perdeli Karışık Uyarma (PAME)

Standart bir kod kitabının kullanıldığı durumda kod kitabı boyunun hem ünsüz, hem de ünlü konuşma uyarımlarına hizmet edecek kadar büyük olduğu varsayılır. VSE durumunda LTP çıkışına göre ortogonalleştirme, ikincil uyarma vektörlerinin LTP tarafından kapsanmayan bölgeleri kapsamamasını ve böylece daha iyi bir sistem oluşmasını sağlar. Ancak düşük bit hızlarında, her iki yapı, kodlanacak ünlü konuşma durumunda beklenen başarıyı elde edemez. Bu nedenle, alt çerçeve (vektör) boyunun büyütülmesinden dolayı bit hızı azalırken konuşma kalitesi de önemli ölçüde bozulur. Dengeli durumdaki ünsüz konuşma süresinde ikincil uyarımdan konuşmaya eklenen gürültü miktarı da önemlidir. Zorlanmış bir kazanç yaklaşımı, dengeli durumdaki sesli konuşmalar süresince ikincil uyarmanın gücünü sınırlayarak daha temiz ünsüz konuşmalar üretir. Başka biçimlerde, kod kitabının bir kısmının

daha temiz ünlü konuşmalar üretilebilir. Başka uygulamalarda, kod kitabının bir kısmının çoklu darbe uyarması gibi olduğu, karışık kod kitapları kullanılır. Ancak, tüm bu yaklaşımların önceden seçilmiş kod kitabı yapıları vardır. Aşağıda, uyarma darbe pozisyonunun aynı alt çerçeve için hesaplanmış LTP sarkması ile adaptif olarak belirlendiği, dinamik bir ikincil uyarma kod kitabı anlatılmaktadır.

Adaptif perdeli karışık uyarda (*Pitch Adaptive Mixed Excitation-PAME*), sabit kod kitabı iki kısımdan oluşmaktadır [18, 19]. Birinci kısım, aşağıda açıklandığı gibi LTP sarkmasına (perde gecikmesi) bağlı olarak adaptif hale getirilir. Uyarma tampon hafızası, ilk pozisyondan başlamak üzere her D örnekte bir, birim örnek genlikleri ile doldurulur. Kalan vektör elemanları sıfırlanır. Kod kitabının aranması süresince bu vektör sentezlenir ve faz pozisyonu, yapay çıktısı her defasında bir örnek olmak üzere $D-1$ kere kaydırılarak saptanır. Daha sonra her faz pozisyonu yeni bir uyarma vektörü gibi işlem görür. LTP aramasında perde çiftleme hatalarına karşı, D sarkması $2D_{min}$ 'den büyükse, her $D/2$ örneğinde bir uyarma darbelerinin yerleştirilmesi yoluyla işlem tekrar uygulanır. Daha sonra aranan uyarma vektörlerinin sayısı, ilgilenilen faz pozisyonlarının toplanmasıyla bulunur. Bu, örnek seyreltme faktörü D ve $D/2$ olan düzenli darbe uyarmasına benzer. C_a faz pozisyonlarını kullanarak kod kitabının perde adaptif kısmından en iyi uyarma vektörünü seçtikten sonra, arama kod kitabının sabit olan ve merkezden katlamalı üst üste binen uyardan oluşan ikinci kısmında devam eder. Burada, $C_f = C - C_a$ adet vektör aranır ve toplam konuşma sürecinde en iyi performans gösteren vektör alıcıya iletilir. Alıcıda LTP sarkmasının kodu çözüldükten sonra, ilgili uyarma vektörünün kodu çözülür.

İkincil uyarmanın perde yapısı almasını zorlayarak, ünlü bölgeleri daha hassas olarak eşlemek mümkündür. Bunun sebebi, ilk olarak periyodikliği daha duyarlı olarak takip edebilmek için LTP belleğinin daha hızlı olması, ikinci olarak LTP'nin başarısız olduğu yerlerde ikincil uyarmanın gereken periodu sağlamasıdır. Bu doğal olarak, ilk önce LTP'nin periodu duyarlı olarak hesaplamasına bağlıdır. İkincil uyarma darbelerini her perde periyodunda hassas olarak yerleştirmek için başka birçok uyarılama yapısı kullanılabilir. LTP sarkma yöntemi, fazladan hesaplama veya bit ihtiyacı göstermediğinden kullanışlıdır. Bu algorithmada kod kitabı indisinin , kodlanması ve kodunun çözülmesi için üç olasılık bulunmaktadır:

$$(1) D = L$$

$$(2) L/2 = D < L$$

$$(3) D < L/2, D_{min} < L/2 \text{ olduğu varsayılırsa}$$

Her olasılık için dikkate alınacak toplam faz pozisyonları aşağıdaki gibi hesaplanabilir:

(1) $D = L$ durumunda, ikincil uyarma vektörünün ilk pozisyonunda tek bir uyarma darbesi bulunacak ve böylece olası L fazı pozisyonları dikkate alınacaktır. Ast katları da L 'den büyükse süreç durur. Ancak, eğer $D/2 < L$ ise, $D/2$ pozisyonunda fazladan bir darbe yerleştirilmiş uyarma vektörü bulunan $L - D/2$ kadar daha fazla faz pozisyonları dikkate alınır. Dolayısıyla toplam faz pozisyonları $2L - D/2$ olmaktadır.

(2) $L/2 = D < L$ durumunda, ikincil uyarma vektörünün ilk ve D 'inci pozisyonlarında iki darbe bulunacaktır. Dolayısıyla dikkate alınacak toplam faz pozisyonları D adettir. Eğer $D/2 \geq D_{min}$ ise, $D/2$ pozisyonları bir daha aranacak ve toplam faz pozisyonları $3D/2$ olacaktır.

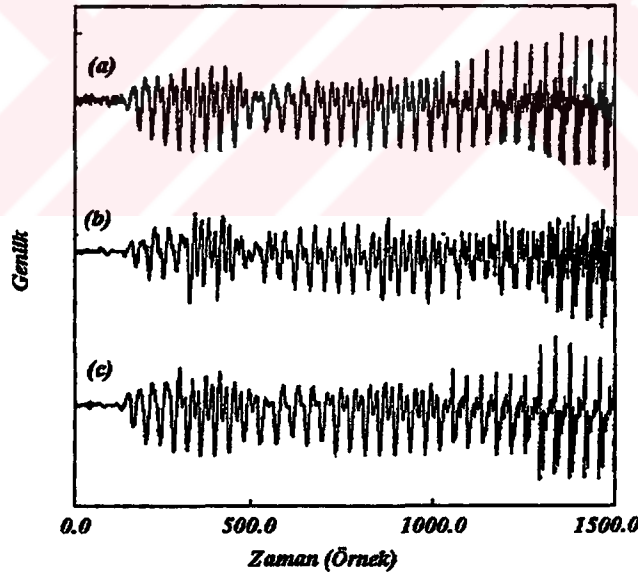
(3) Son olarak, $D < L/2$ ise birinci pozisyonla başlamak üzere ikincil uyarmanın her D örneğinde darbeler bulunacak ve muhtemelen sonuç D adet faz pozisyonu olacaktır. Ancak, eğer $D/2 \geq D_{min}$ ise, $D/2$ pozisyonları bir daha dikkate alınacak ve $3D/2$ neticesini verecektir.

Yukarıdaki olasılıklar, sabit alt çerçeve boyu ve kodlanmış LTP sarkması D olarak alıcıya gönderilir.

Üç uyarma yapısının öznel dinleme karşılaştırmaları sonucunda, adaptif perdeli uyarma, konuşmayı daha anlaşılır kılarak en iyi sonucu üretmiştir. Bu, ikincil uyarmanın periyodik kısmının, ünlü konuşmaya daha süratli ve dolayısıyla daha hassas uyması nedeniyle mümkün olmuştur. Adaptif perdeli uyarma yapısıyla perdesi daha iyi temsil edilen bir konuşma parçası Şekil 6.41'de gösterilmiştir.

Şekilde PAME'ye ait seslerin daha hızlı değiştiği de görülebilir. Burada, STP ve LTP ters filtrelenmiş konuşma spektrumunun düz olduğu varsayılmıştır. Ancak, pratikte bu varsayım tam olarak doğru değildir. Bu nedenle ikincil uyarma, modelin yanıtlarını ortadan kaldıracak şekilde yapılmalıdır. Her ne kadar bu tüm ikincil uyarma tiplerine uygulanabilirse de, zamanında bir tek darbe yerleştirmenin frekansta düz bir tepkiye denk düştüğü perde uyarlama durumunda daha önemlidir. Bu nedenle konuşma kalitesini daha da iyileştirmek için ikincil uyarıma LPC parametrelerini kullanan adaptif şekillendirme uygulanabilir. Bu şekillendirme, en etkili neticeyi almak için AbS devresinde gerçekleştirilmelidir. Değiştirilmiş LPC parametrelerini kullanan adaptif şekillendirme filtresinin transfer fonksiyonu aşağıdaki gibi gösterilebilir

$$H_s(z) = \frac{1 - \sum_{i=1}^p a_i \beta^i z^{-i}}{1 - \sum_{i=1}^p a_i \alpha^i z^{-i}} \quad (6.115)$$

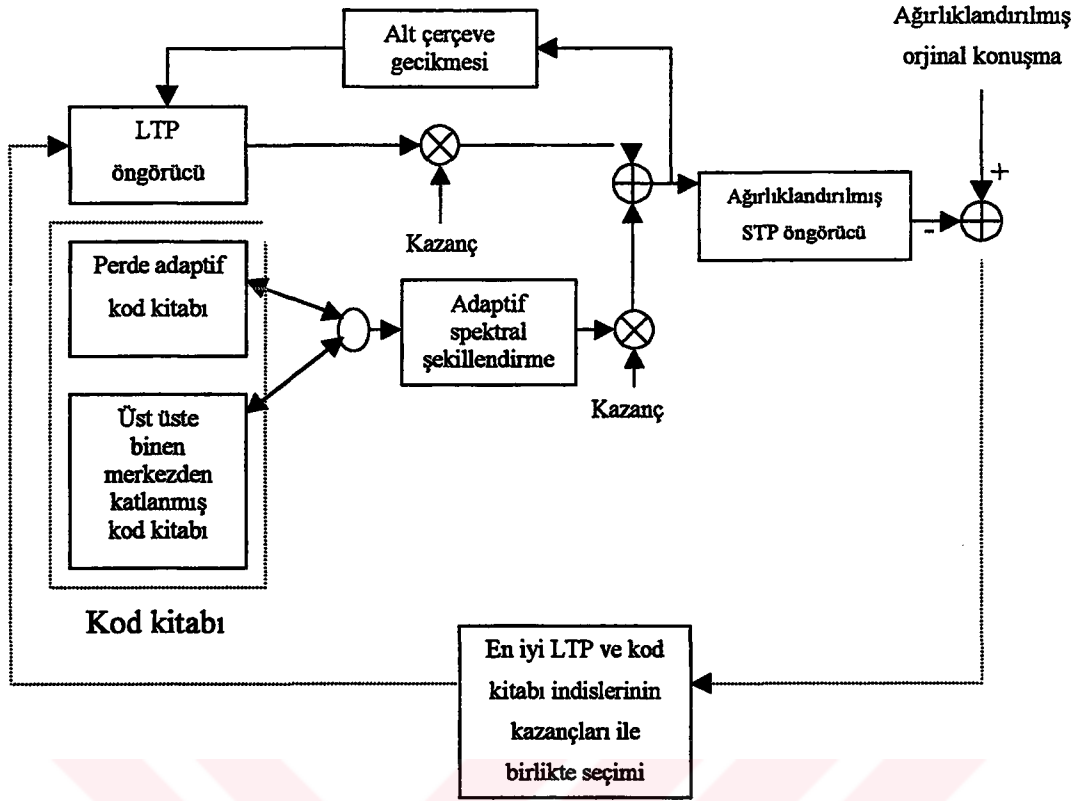


Şekil 6.41 (a) orijinal, (b) üst üste binen merkezden katlanmış Gauss uyarmalı CELP çıktısı ve (c) PAME uyarmalı CELP çıktılarının konuşma dalga biçimleri

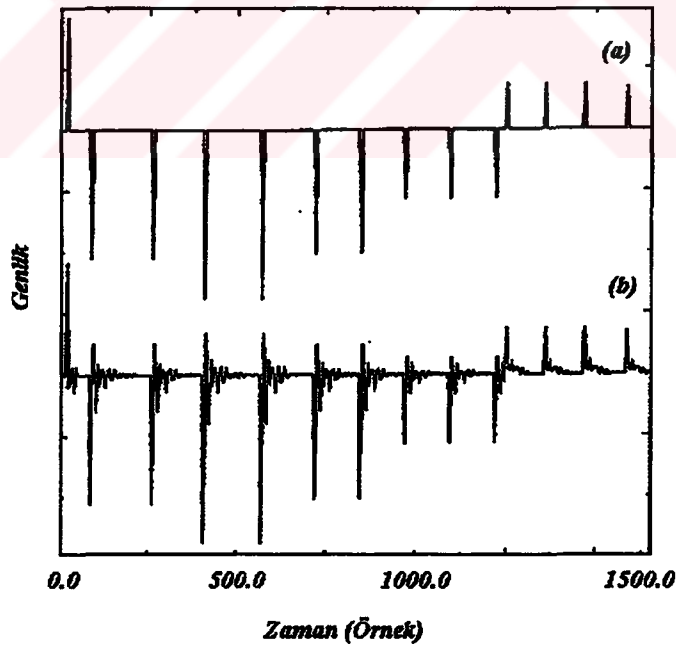
Tipik β ve α katsayı kombinasyonları sırasıyla 0.6 ve 0.7, 0.7 ve 0.8 veya 0.7 ve 0.9'dur. Adaptif şekillendirme dahil, adaptif perdeli ikincil uyarmalı bir CELP kodlayıcının blok diyagramı Şekil 6.42'de verilmiştir. Şekillendirme süreci fazladan hesaplama gerektirmekteyse de eğer dürtü cevabı, STP filtresi dürtü cevabı ile birlikte hesaplandığı takdirde arama süresince fazladan hesaplama gerekmez. Hem kodlayıcı, hem kod çözücüde gereken fazladan hesaplama, sonuç çıktıyı üretmek için (6.15) denkleminin her alt çerçevede bir kez uygulanmasından ibarettir. Şekil 6.43, spektrum şekillemesi olan ve olmayan PAME uygulaması çıktılarını göstermektedir.

6.4.4 Değişik İkincil Uyarmalı CELP'in Performansı

Çeşitli ikincil uyarmalı CELP'in performansını saptamak için Tablo 6.6 ve 6.10'da tanımlanan ve gürültü şekillendirme ağırlıklandırması 0.8 olan tam nicemlenmiş iki CELP kodlayıcısı ele alınsın. Her iki biçimde çerçeve enerjisi, uyarma vektör kazançları ve LTP kazançlarının tümü, her parametrenin dinamik dizisini kapsayacak şekilde optimum hale getirilmiş düzgün olmayan şekilde nicemlenmiştir. Tablo 6.6'da, verilen parametreler sonucunda elde edilen kodlayıcı yaklaşık 8 kbit/san hızında çalışmaktadır. Tablo 6.7'de, çeşitli kod kitabı boyları için 1 ara uçlu kapalı devre LTP içeren CELP kodlayıcısının performansı verilmektedir. Tablo 6.8'de, aynı kodlayıcının performansı, 3 ara uçlu bir LTP için gösterilmiştir. 1 ara uçlu LTP'de kazanç nicemleme yalnız pozitif değerleri kapsayacak şekilde sınırlandırılmıştır. 3 ara uçlu LTP durumunda ise kazanç değerleri için, 7 bitlik bir kod kitabı kullanılarak vektör nicemleme uygulanmıştır. Görüleceği üzere, 3 ara uçlu LTP genelde, daha yüksek bit hızı oluşturur, fakat performansı da yüksektir. Ancak, bu husus yüksek bit hızlarında öznel kaliteye genel anlamda yansımaz. Tablo 6.9'da, aynı kodlayıcının 3 ara uçlu LTP'si her 10 msan'de bir güncellenmiş halde gösterilmiştir. Sonuçta SNR'de görülen sadece küçük bir düşüş, LTP'nin uzun süreli ortak ilişkileri yeterince sağladığını belirtmektedir.



Şekil 6.42 PAME ve spektrum şekillendirmeli CELP'in blok diyagramı



Şekil 6.43 (a) orijinal ve (b) şekillenmiş PAME uyarma dalga biçimlerinin grafiği

Tablo 6.6 Yaklaşık 8 kbit/san'lik CELP yapılarının bit ataması

<i>Parametreler</i>	<i>Sayı</i>	<i>Bit</i>	<i>Güncelleme/msan</i>	<i>Bit Hızı (bit/san)</i>
STP(LSP-SQ)	10	37	20	1850
LTP	1 ara-uç	7+4=11	5	2200
LTP	3 ara-uç	7+7=14	5/10	2800/1400
CB indisi	1	9/10/11	5	1800/2000/2200
CB kazancı	1	5	5	1000
Çerçeve enerjisi	1	5	20	250

Tablo 6.7 1 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı

<i>CB Boyu</i>	<i>Hız</i>	<i>OCCG</i>	<i>VSE</i>	<i>PAME</i>
9	7100	12.41	11.89	12.63
10	7300	12.70	12.40	12.89
11	7500	12.93	12.45	13.07

Tablo 6.8 5 msan'de güncellenmiş 3 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı

<i>CB Boyu</i>	<i>Hız</i>	<i>OCCG</i>	<i>VSE</i>	<i>PAME</i>
9	7700	13.02	12.80	13.11
10	7900	13.33	13.02	13.43
11	8100	13.62	13.38	13.70

Tablo 6.9 10 msan'de güncelleştirilmiş 1 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı

<i>CB Boyu</i>	<i>Hız</i>	<i>OCCG</i>	<i>VSE</i>	<i>PAME</i>
9	6300	12.01	11.91	12.33
10	6500	12.48	12.09	12.78
11	6700	12.80	12.17	13.01

Tablo 6.10 Yaklaşık 5 kbit/san hızında CELP yapılarının bit ataması

Parametreler	Sayı	Bit	Güncelleme/msan	Bit Hızı (bit/san)
STP(LSP-SQ)	10	25	30	833.3
LTP	1 ara-uç	7+4=11	7.5	1466.6
LTP	3 ara-uç	7+7=14	7.5	1866.6
CB indisi	1	9/10/11	7.5	1200/1333.3/1466.7
CB kazancı	1	5	7.5	666.7
Çerçeve enerjisi	1	5	30	166.7

Tablo 6.11 1 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı

<i>CB Boyu</i>	<i>Hız</i>	<i>OCCG</i>	<i>VSE</i>	<i>PAME</i>
9	4333.4	9.22	8.76	9.48
10	4466.7	9.40	8.98	9.65
11	4600.0	9.78	9.35	9.95

Tablo 6.12 3 ara uçlu CL-LTP'li çeşitli CELP yapılarının SegSNR performansı

<i>CB Boyu</i>	<i>Hız</i>	<i>OCCG</i>	<i>VSE</i>	<i>PAME</i>
9	4733.4	9.81	9.52	9.96
10	4866.7	10.03	9.81	10.14
11	5000.0	10.15	9.90	10.21

Tablo 6.10'da konfigürasyonu verilen yaklaşık 5 kbit/san'lik bir CELP yapısı için benzer sonuçlar Tablo 6.11 ve 6.12'de gösterilmiştir. Objektif SNR değerlerinden anlaşıldığı gibi daha düşük hızlı kodlayıcıların performansı, 8 kbit/san'lik kodlayıcılar kadar iyi değildir. 8 kbit/san'lik kodlayıcılar arasında ise, 3 ara uçlu LTP kullanımı performansı biraz düzeltmektedir. Ancak genellikle standart *Gauss* uyarması ve VSE kullanan daha düşük hızlı CELP kodlayıcılar ile kodlanmış konuşma işaretleri, dikkat çekecek miktarda nicemleme gürültüsü içermektedir. Bu, basit olarak kodlayıcının büyük vektör boyları ile işlem yapmaya uygun olmaması nedeniyle ortaya çıkmaktadır. VSE, en düşük SNR'ye sahip olmasına rağmen, aynı zamanda öznel açıdan daha gürültülüdür. Bunun sebebi, testlerde kullanılan, üzerinde çalışılmamış temel vektörler olabilir. Eğitim yoluyla VSE performansının arttırabildiği bilinmektedir [12].

Tablolarda belirtildiği gibi PAME, tutarlı olarak en yüksek SNR'leri üretmiştir. Subjektif açıdan rastgele yapılan dinleme testleri sonucunda PAME gerçekte tüm durumlarda en iyi sonucu vermiştir ve yüksek hızlı ile alçak hızlı modeller arasındaki kalite farkı, diğer iki uyarma tipinde olduğu kadar değildir. LTP sarkmasının en başta doğru seçilmemesinin PAME performansını etkileyeceği ortadadır. Bu nedenle, LTP araması süresinde doğru LTP sarkması veya tamsayı katlarının seçilmesi önemlidir. Olası tüm LTP sarkmaları (bir alt çerçevedeki) ve faz pozisyonları etraflıca arandığı takdirde PAME performansının LTP'ye olan bağımlılığı kaldırılabilir. Ancak bu, kod kitabının uyarlama kısmında daha çok indis değerinin kodlanmasını gerektirecektir.

Bu durumda, ilk olarak uyarma tampon belleği x 'in başlangıcına ve daha sonra her P örnekte bir, birim genlik darbesi yerleştirilerek birincil uyarma vektörleri oluşturulur. Tüm birincil vektörleri içermek için P , D_{min} 'den (olabilecek en küçük perde) L 'ye (alt çerçeve boyu) değiştirilir. D_{min} en az perde ile ilişkili iken aslına sadakati arttırmak için değiştirilebilir. Bu nedenle her P için birincil aday uyarma aşağıdaki şekilde türetilir:

$$x_j(n) = \begin{cases} 1 & n = iP \leq L \quad i = 0, 1, 2, \dots \\ 0 & \text{aksi halde} \end{cases} \quad (6.116)$$

Olası tüm faz pozisyonlarını oluşturmak amacıyla, her birincil vektör x_j için, $P-1$ adet daha x_{j+k} vektörleri ($k=1, 2, \dots, P-1$), aşağıdaki şekilde kaydırılarak türetilir:

$$x_{j+k}(n) = \begin{cases} 0 & n = 0, 1, 2, \dots, k-1 \\ x_j(k-n) & n = k, k+1, \dots, L-1 \end{cases} \quad (6.117)$$

Aday uyarma vektörleri sayısı C_a 'nın, L ve D_{min} 'e, aşağıdaki gibi bağımlıdır:

$$C_a = L + \sum_{l=D_{min}}^{L/2} I + \sum_{I=1}^{L/2} I \quad (6.118)$$

Böylece, adaptif uyarma indisi için gerek duyulan bit sayısı $B = \lceil \log_2 C_a \rceil$ 'dir. Eğer C_a , 2'nin tamsayı üssü değilse $2^B - C_a$ adet vektör daha sabit kod kitabında tekrar aranır.

Bir CELP kodlayıcısının tablo 6.13'te tanımlanan çeşitli seviyelerdeki, tek ara uçlu kapalı devre LTP ve PAME ikincil uyarmalı tipik dalga biçimleri Şekil 6.44'de gösterilmiştir. Bu konfigürasyonda çerçeve enerjisinin karşı tarafa aktarılmamaktadır. Bu nedenle, Tablo 6.14'te verildiği gibi, ikincil uyarma kazançları, giriş işaretinin dinamik aralığını kapsamaya uygun düzgün olmayan 5 bit'lik nicemleyici ile nicemlenmiştir.

Tablo 6.13 4.7 kbit/san'de örnek bir CELP konfigürasyonu

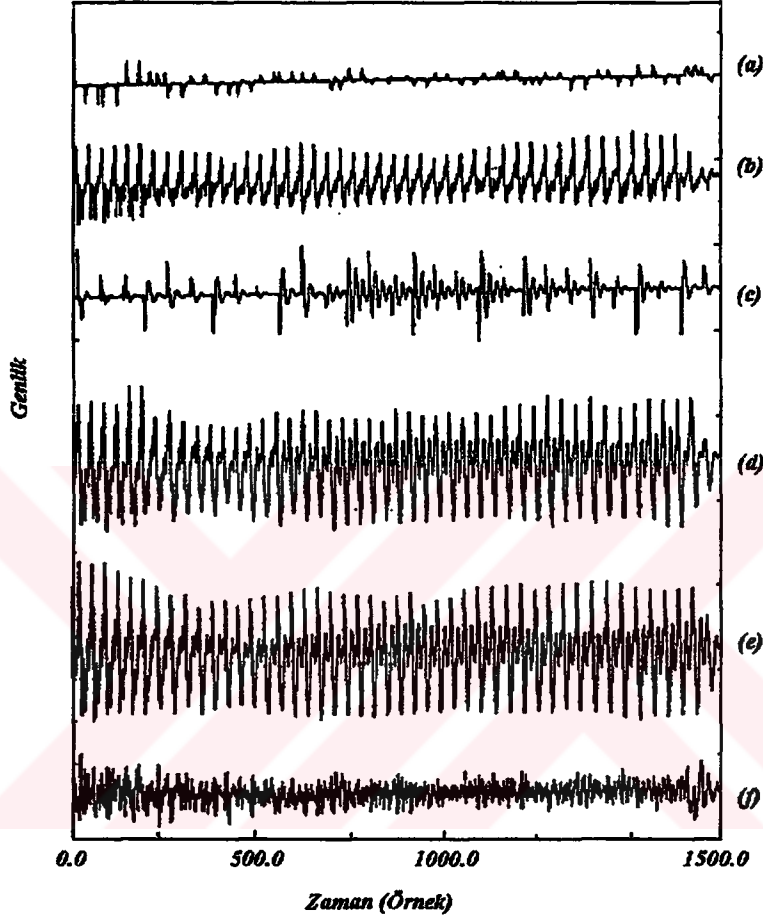
Parametreler	Sayı	Bit	Güncelleme/msan	Bit Hızı (bit/san)
STP(LSP-SQ)	10	37	30	1233.3
LTP	1 ara-uç	7+5=12	7.5	1600
CB indisi	1	9	7.5	1200
CB kazancı	1	5	7.5	666.6
Toplam Bit Hızı	-	-	-	4700

Tablo 6.14 Uyarma vektör kazançları için düzgün olmayan 5 bit nicemleyici

Seviye	Kazanç	Seviye	Kazanç	Seviye	Kazanç	Seviye	Kazanç
0	-1100	8	-175	16	1	24	220
1	-850	9	-135	17	3	25	275
2	-650	10	-98	18	12	26	335
3	-510	11	-65	19	35	27	415
4	-415	12	-35	20	65	28	510
5	-335	13	-12	21	98	29	650
6	-275	14	-3	22	135	30	850
7	-220	15	-1	23	175	31	1100

Tablo 6.15'te belirtildiği gibi LTP kazançlarının negatif olmasına izin verilmiştir [8]. Bu konfigürasyonun 6 kbit/san'den düşük bit hızlarında en iyi sonuçları ürettiği bulunmuştur. Bu şekilde ünsüz işaretler ve geçiş bölgeleri için kalite iyileştirilebilir. Vektör kazanç nicemleyicisinde en az kazanç 1,0 olarak belirlenmiştir. Eğer herhangi bir seviye 0'a seçilirse temiz ve hafif gürültülü çıkış arasındaki fon gürültüsü, arzu edilmeyen şekilde, modülasyon etkisi oluşturmaktadır. Sıfır olmayan seviyelerin diğer bir önemli etkisi, LTP belleğinin sürekli değiştirilmesi gereğidir. Bu, sessizlik bölgelerinde, LTP kod çözücünden kanal hatalarının temizlenmesi için çok uygundur. Şekil 6.44'de gösterilen dalga biçimlerinden, LTP katkısının çok önemli olduğu görülmektedir. Bunu doğrulamak için Şekil 6.45'de referans işareti ile her LTP ve

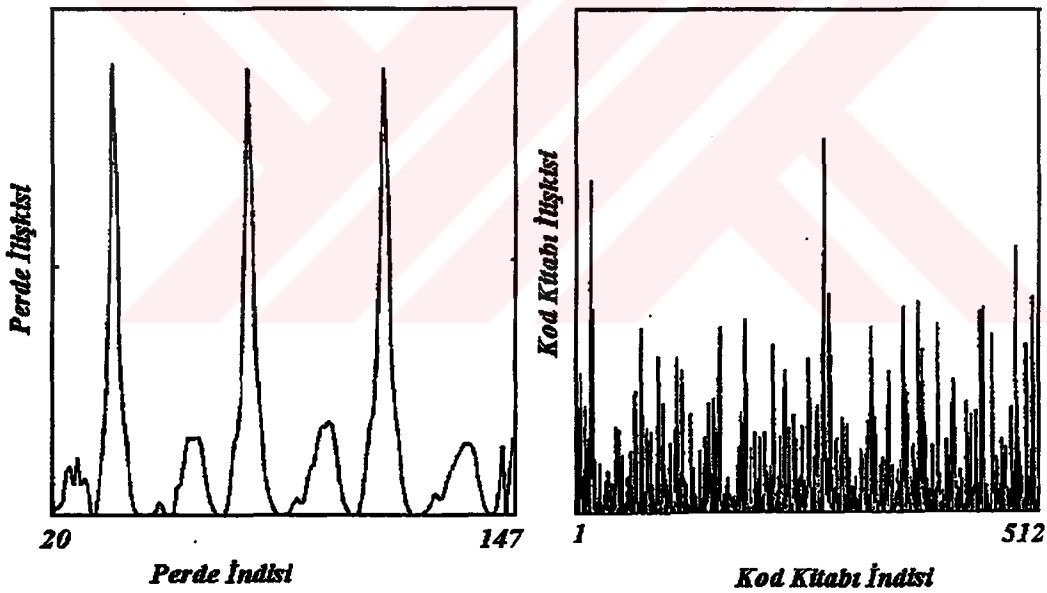
kod kitabı indislerinden doğan katkılar arasında üretilen ilişki gösterilmektedir. Görülebileceği üzere LTP indisi, her perde katında ayırdedici tepe şeklinde ilişkileri içermektedir.



Şekil 6.44 Tipik CELP dalga biçimleri. (a) Kod kitabı katkısı, (b) LTP (perde) katkısı, (c) STP belleği katkısı, (d) toplam çıktı, (e) orijinal konuşma ve (f) kalan hata

Tablo 6.15 LTP kazançları için düzenli olmayan 5 bit nicemleyici

Seviye	Kazanç	Seviye	Kazanç	Seviye	Kazanç	Seviye	Kazanç
0	-0.993	8	0.255	16	0.780	24	1.062
1	-0.831	9	0.368	17	0.816	25	1.117
2	-0.693	10	0.457	18	0.850	26	1.193
3	-0.555	11	0.531	19	0.881	27	1.289
4	-0.414	12	0.601	20	0.915	28	1.394
5	-0.229	13	0.653	21	0.948	29	1.540
6	0.000	14	0.702	22	0.983	30	1.765
7	0.193	15	0.745	23	1.020	31	1.991



Şekil 6.45 Bir konuşma alt çerçevesi için LTP (perde) ve kod kitabı ilişkisi

Diğer yandan, kod kitabı indis değişimi, yeterince rastgele bir ilişki sergiler. Bu da demektir ki LTP, uzun süreli ilişkiyi giriş konuşmasından yok etmiş ve kod kitabı uyarması ile eşleşecek oldukça rasgele görünümlü bir işaret bırakmıştır.

6.4.5 Ortak LTP ve İkincil Uyarma Hesabı

LTP ve ikincil uyarmanın toplanması yoluyla, LTP ve ikincil uyarmanın ardışıl olarak hesaplandığı yukarıdaki analiz optimum değildir. LTP optimum yapıldığında ikincil uyarmanın etkisinin 0 olduğu varsayılmaktadır. Böylece, ikincil uyarma hesaplandığında LTP uyarması optimum olmaz. AbS'in ana fikri optimum bir uyarma hesaplamak olduğundan hem LTP, hem ikincil uyarmlar, ideal olarak, beraberce hesaplanmalıdır. Tek ara uçlu LTP ve tek kazançlı ikincil uyarma kabul edilerek, bir alt çerçeve için ağırlıklı ortalama karesel hata aşağıdaki gibi hesaplanabilir:

$$E_w(k.D) = \sum_{n=0}^{L-1} \left[s_w(n) - s_m(n) - \sum_{i=0}^n g_k x_k(n-i)h_w(i) - \sum_{i=0}^n \beta_D \hat{r}(n-i-D)h_w(i) \right] \quad (6.119)$$

STP'nin sıfır giriş bellek cevabı, $s_m(n)$, değiştirilemeyeceği ve k ve D 'nin seçimini etkilemeyeceği için $s_w(n) - s_m(n)$ 'in yerine $s(n)$ konulabilir. Böylece:

$$E_w(k.D) = \sum_{n=0}^{L-1} \left[\bar{s}(n) - g_k \sum_{i=0}^n x_k(n-i)h_w(i) - \beta_D \sum_{i=0}^n \hat{r}(n-i-D)h_w(i) \right]^2 \quad (6.120)$$

denklemi yazılabilir. Bu denklem, $D_{min} \leq D \leq D_{max}$ ve $0 \leq k \leq C-1$ için araştırılır. Denklemi daha da basitleştirmek için aşağıdaki eşitlikler yerlerine konulur.

$$Z_D(n) = \sum_{i=0}^n \hat{r}(n-i-D)h_w(i) \quad (6.121)$$

ve

$$V_k(n) = \sum_{i=0}^n x_k(n-i)h_w(i) \quad (6.122)$$

Böylece:

$$E_w(k, D) = \sum_{n=0}^{L-1} [\tilde{s}(n) - g_k V_k(n) - \beta_D Z_D(n)]^2 \quad (6.123)$$

denklemini elde edilir. (6.123)'teki değişkenler k ve D , optimum kazançlar g_k ve β_D ile toplam hatalar $E_w(k, D)$ en az olacak şekilde beraber seçilmelidir. $E_w(k, D)$ 'yi en aza indirmek için k ve D 'nin fonksiyonu olan (6.123) g_k ve β_D 'ye göre türetilir.

Böylece:

$$\frac{\partial E_w(k, D)}{\partial g_k} = \sum_{n=0}^{L-1} [\tilde{s}(n) - g_k V_k(n) - \beta_D Z_D(n)] \times (-2) V_k(n) = 0 \quad (6.124)$$

oluşur ve benzer şekilde:

$$\frac{\partial E_w(k, D)}{\partial \beta_D} = \sum_{n=0}^{L-1} [\tilde{s}(n) - g_k V_k(n) - \beta_D Z_D(n)] \times (-2) Z_D(n) = 0 \quad (6.125)$$

Yukarıdaki denklemleri tekrar düzenleyerek aşağıdaki denklemler elde edilir:

$$\sum_{n=0}^{L-1} \tilde{s}(n) V_k(n) = g_k \sum_{n=0}^{L-1} V_k^2(n) + \beta_D \sum_{n=0}^{L-1} Z_D(n) V_k(n) \quad (6.126)$$

$$\sum_{n=0}^{L-1} \tilde{s}(n) Z_D(n) = g_k \sum_{n=0}^{L-1} V_k(n) Z_D(n) + \beta_D \sum_{n=0}^{L-1} Z_D^2(n) \quad (6.127)$$

Matris biçiminde ifade edilirse:

$$\begin{bmatrix} g_k \\ \beta_D \end{bmatrix} = \begin{bmatrix} G_1 & G_2 \\ G_2 & G_3 \end{bmatrix}^{-1} \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \quad (6.128)$$

bulunur. Burada:

$$G_2 = \sum_{n=0}^{L-1} Z_D(n) V_k(n) \quad (6.130)$$

$$G_3 = \sum_{n=0}^{L-1} Z_D^2(n) \quad (6.131)$$

$$K_1 = \sum_{n=0}^{L-1} \tilde{s}(n) V_k(n) \quad (6.132)$$

$$K_2 = \sum_{n=0}^{L-1} \tilde{s}(n) Z_D(n) \quad (6.133)$$

g_k ve β_D kazançları, yukarıdaki denklemin aşağıdaki gibi çözülmesiyle bulunabilir.

$$g_k = \frac{K_1 G_3 - K_2 G_2}{G_1 G_3 - G_2^2} \quad (6.134)$$

$$\beta_D = \frac{K_2 G_1 - K_1 G_2}{G_1 G_3 - G_2^2} \quad (6.135)$$

Yukarıdaki denklem kullanılarak tüm kombinasyonlar içinde k ve D değişkenleri aranır ve en az hata veren kombinasyon seçilir. Her ne kadar LTP ve ikincil uyarmanın ortak hesaplanmasıyla konuşma kalitesi önemli derecede yükselirse de gereken arama sayısı çok yüksektir. Ortak optimizasyon, özellikle PAME tipi ikincil uyarma için kullanışlıdır. PAMEi ortak optimizasyonda kullanılarak ünsüz konuşma bölgelerinin ikincil uyarma tarafından daha doğru olarak ifade edilmesi sağlanır. Ayrıca, toplam uyarmayı LTP ile daha doğru paylaşarak kalıcı durumdaki ünlü bölgeler iyileştirilebilir.

Sıralı optimizasyonun sebebi, böylelikle gereken hesaplama sayısının indirilmesidir.

Örneğin, LTP'nin 128 ($20 \leq D \leq 147$) olası sarkması ve ikincil uyarmanın maksimum

$C-1$ değeri bir indis değeri varsa, sıralı optimizasyon metodunda toplam olasılık sayısı $128+C$ 'dir. Ancak ortak optimizasyon durumunda bu $128 \times C = 128C$ kombinasyona yükselir. Bu nedenle, zaman zaman aşağıdaki açıklanan optimum benzeri bir ortak optimizasyon yöntemi kullanılır. LTP'nin D sarkmasının perde süresi veya katlarından birinde olduğu varsayılarak, sarkma D_{opt} 'u tanımlamak için kapalı devre LTP metodu kullanılarak ilk önce LTP aranır. Daha sonra, seçilmiş LTP uyarması ile birlikte ortak olarak karesel ortalama hatayı en aza indirecek vektörü bulmak için ikincil uyarma aranır. Burada yalnız LTP uyarma şeklinin sabit olduğuna dikkat edilmelidir. Her ikincil uyarma adayı için kazançların ortak optimumu bulunur ve en az hata kombinasyonu seçilir. Bu, matris hesaplama sayısını ikincil uyarma indis boyutuna indirir. Bu tür basitleştirilmiş ortak optimizasyonlar, ilk önce LTP uyarma boyutunun doğru seçimine bağlıdır, ki bu da ünlü konuşma bölgelerinde çok zor değildir. Ancak konuşmanın geçiş ve ünsüz bölgelerinde performans, sıralı optimizasyon yaklaşımı düzeyine inmektedir.

BÖLÜM 7

CELP KODLAYICILARIN GERÇEK ZAMAN UYGULAMALARI

7.1 GERÇEK ZAMANLI SÜREÇLER

Gerçek-zamanlı süreçler, belirli bir zaman içerisinde yapılması gerekli görevlerdir. Burada gerçek zamanlı terimi, 200 ms'den 20 veya 30 ms'ye kadar değişebilen bir süre için kullanılmıştır. Konuşmanın sayısal olarak kodlanmasında, örnekleme hızı 8 kHz varsayıldığında, gerçek zamanlı süreç için, kodlanmanın 125 µs içerisinde gerçekleştirilmesi gerekmektedir. Bununla birlikte CELP benzeri kodlayıcı tiplerinde kodlama blok blok işlenerek, yani konuşma işareti 20-30 ms'lik çerçevelere ve bu çerçeveler de 4-8 ms'lik alt çerçevelere ayrılarak kodlama için tanınan zaman sınırı artırılmak suretiyle bloklar halinde yerine getirilir.

Bir konuşma kodlayıcı algoritmasının gerçek zaman uygulamasının başarımı kodlayıcının toplam maliyeti açısından çok önemlidir. Herhangi bir konuşma kodlama algoritması, mevcut bir Sayısal İşaret İşlemci (*Digital Signal Processor-DSP*) tümdevresi kullanılarak gerçekleştirilebilir, ancak bu uygulamanın maliyeti, kullanılan DSP tümdevresi sayısından büyük oranda etkilenecektir. Gerçek zamanlı uygulamada gözönünde bulundurulacak diğer önemli bir nokta da, özellikle mobil sistemlerde sistemin toplam güç tüketimidir.

Konuşma kodlama algoritmalarının gelişimine paralel olarak, DSP teknolojisi son beş yılda önemli gelişmeler kaydetmiştir. Güçlü sabit-noktalı (*fixed-point*) ve kayan-noktalı (*floating-point*) DSP'lerin ortaya çıkışı, karmaşık konuşma kodlama algoritmalarının gerçek zamanlı uygulamasını mümkün kılmıştır. Sabit-noktalı DSP'ler genellikle daha hızlı ve daha ucuzdur, ancak bunların konuşma algoritmalarını gerçekleyecek şekilde programlanması ve yüksek doğruluk elde

edilmesi daha güçtür. Maliyet açısından sabit-noktalı DSP'lerin kullanılması arzu edilen bir durum olsa da yüksek derecede kayan-noktalı işlem içeren konuşma kodlama algoritmalarının tasarlanması çok kolay değildir. Öte taraftan kayan-noktalı DSP'lerin kullanımı, ölçekleme, normalizasyon ve taşma gibi sabit-noktalı DSP'lerin kullanımında geliştirme işini zorlaştıran sorunları içermediğinden uygulamanın geliştirilmesini kolaylaştırabilir.

Özel bir konuşma kodlama algoritmasının gerçekleştirilmesi için DSP'nin seçimi, diğer başka faktörlerden de etkilenir. Üzerinde iyice düşünülmesi gerekli en önemli iki faktör, DSP'nin komut çevrim (*instruction cycle*) süresi (hesaplama hızı) ve işlemcinin komut kümesinin konuşma kodlama algoritmasındaki ana işlem bloklarını gerçekleştirme için uygunluğudur. Örneğin, CELP tipi algoritmalarda DSP işlem zamanının çoğunluğu çarpma/toplama komutlarına harcanır. Bu tür algoritmalarda, yoğun kullanılan işlemleri, tek komut çevrimiyle yapan DSP'ler tercih edilmelidir. DSP ile ilgili dikkat edilmesi gereken diğer özellikler şunlardır :

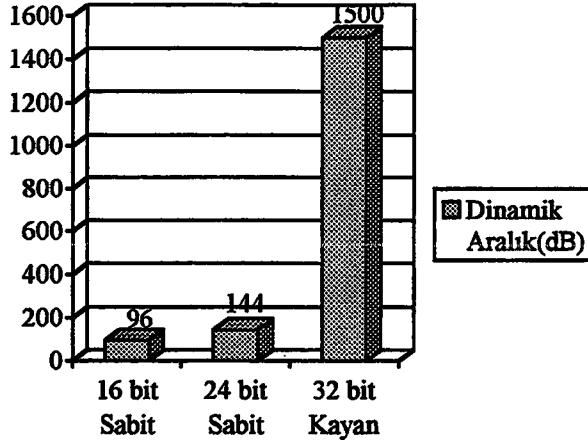
- Tümdevre üzerinde bulunan RAM/ROM miktarı,
- Program ve veri için kullanılan veriyolu yapısı,
- Adresleme kapasitesi,
- Tümdevre üzerinde bulunan ön bellek (*cache memory*),
- Tümdevre üzerinde bulunan çevrebirim (*peripherie*) donanımı,
- Sabit ve kayan-noktalı aritmetik işlem kapasitesi,
- Adresleme modelleri,
- Tümdevre üzerinde bulunan doğrudan bellek erişimi (*Direct Memory Access-DMA*) imkanı,
- Harici ve dahili kesme (*interrupt*) imkanı,
- Donanım/yazılım ile programlanabilen bekleme durumları (*wait state*),
- Tümdevre üzerinde bulunan emülasyon portları,
- Güç harcamayı kullanılmayan durumlarda en aza indirme (*power-down*) imkanı,
- Düşük güç harcaması.

7.2 SAYISAL İŞARET İŞLEMCİLER

Sayısal İşaret İşlemciler hassasiyet ve aritmetik işlem tiplerine göre sınıflanır: Sabit-noktalı DSP'ler daha hızlı ve daha ucuzdur, ancak bunların programlanması daha zor olup daha düşük hassasiyet sağlar. Bir DSP sistemindeki toplam kapı sayısı, kullanılan nümerik gösterimdeki bit sayısının bir fonksiyonudur. Kapı sayısı, kullanılan nümerik gösterimin mevcut aralığı ve hassasiyeti ile doğru orantılı olarak artar. Bu nedenle, matematiksel performans ile, devre ve sistem karmaşıklığı arasında bir seçim yapılmalıdır. Sistem işlevselliği için gerekli olmayan bir performans düzeyi sağlayan bir nümerik gösterim kullanmak, kaynakların verimsiz olarak kullanılması demektir.

Kayan-nokta uygulamasının, sabit-nokta uygulamasına karşı üstünlükleri oldukça fazladır. Bu üstünlükler, genellikle bir ürün veya program için yapılan geliştirme çalışmalarında büyük tasarruf sağlar. Mantis, işlemci tarafından normalize edildiği için kayan-noktalı bir sayının hassasiyeti bir program boyunca sabit kalır, oysa sabit-noktalı verinin doğruluğu, saklanan verinin veya bir işlem sonrasında elde edilen değerin büyüklüğüne göre değişiklik gösterir. Kayan-noktalı DSP'lerde mantisin otomatik olarak normalize edilmesi nedeniyle, sayıların yuvarlanması, sabit noktalı uygulamalara göre çok daha küçük hatalara yol açar. Çok büyük veya çok küçük sayıların gösterilmesi yeteneğiyle de birleşen bu sabit hassasiyet, filtrelerin gerçekleştirilmesi sorunlarını ortadan kaldırarak kutup veya sıfırların çok daha doğru yerleştirilmesini sağlar.

Şekil 7.1'de gösterildiği gibi, kayan-noktalı DSP'lerin çok geniş olan dinamik aralıkları, çok küçük veya çok büyük sayıların yüksek hassasiyetle gösterilmesini mümkün kılar. Bu, FFT ve yüksek dereceden özyinelemeli filtrelerin ara hesaplamalarında özellikle gereklidir. Geniş dinamik aralık olması, sabit-noktalı yapılarda verideki aşağıdan veya yukarıdan taşmayı engellemek için bir ara ölçekleme gereksinimini ortadan kaldırır. Bu yüzden, kayan-noktalı bir DSP'nin kullanımı genellikle yazılımın hacmini ve karmaşıklığını azaltır.



Şekil 7.1 Sabit ve Kayan-Noktalı DSP'lerin Dinamik Aralığı

7.2.1 Sabit-Noktalı Uygulama

Çoğu programcı, kullanım olarak kayan-noktalı sayılara daha yatkındır ve sabit-noktalı sayıların kullanımını sadece, dizi indeksleri, sayaçlar veya diğer değişkenler olmak üzere sınırlı tutar. Bununla birlikte, kullanıcının üstten taşma (*overflow*) veya alttan taşma (*underflow*) gibi iç konulara yakın bir ilgi göstermesinin gerektiği sabit-noktalı DSP'ler kullanılırken bu durum değişir. Uygulama boyunca düzgün dağılan doğruluk, sayıların ve uygun ölçeklemenin devamlı kontrolüyle sağlanmalıdır.

Doğruluk kaybı (veya nicemleme hataları), genellikle iki sayının çarpımında ortaya çıkar; çünkü çarpım sonucunu tam doğrulukla tanımlamak için gerekli bit sayısı işlem öğelerindeki (*operand*) bitlerin toplamına eşittir. Bu bitlerin herhangi birisinin atılması bilgi eksikliğine yol açar. Çarpım sonucunun tamamını çifte doğruluklu (*double precision*) bir sayı olarak saklamak mümkündür, ama bu hem zaman hem de gerekli bellek açısından çok yük getirir. Bazı DSP'lerde taşma, sonuç değerinin uygun şekilde en yüksek değerli pozitif veya negatif sayıya eşitlenmesiyle kontrol edilir. Bunun çalışması için, aritmetik lojik birimde (*Arithmetic Logic Unit-ALU*) ve ayrıca akümülatör ile veri yolu (*data bus*) arasında doyma (*saturation*) donanımı bulunmalıdır. Taşmayı önlemek için kullanılan bir diğer yöntem ise doğruluk

kaybına yol açsada, veriyi düşük anlamlı bitleri atarak akümülatöre eklemeyen önce işaret biti ile birlikte kaydırmaktır.

100 ns veya daha düşük komut çevrim zamanlı sabit-noktalı DSP'lerin karşılaştırması Tablo 7.1'de verilmiştir. Mevcut tipik sabit-noktalı DSP'lerin özelliklerini göstermek için aşağıda TMS 320C50 işlemcisi kısaca açıklanmıştır.

Tablo 7.1 Sabit Noktalı Sayısal İşaret İşlemci Örnekleri

(P: Program Belleği, V: Veri Belleği) (*: 48x48=96 bit çift duyarlıklı işlem)

Firma	Veri (bit)	MAC giriş hassasiyeti (bit)	MAC çıkış hassasiyeti (bit)	Komut Çevrim Süresi (ns)	Saat Frekansı (MHz)	İşlem Kapasitesi (MIPS)	Üzerindeki RAM (bit)	Üzerindeki ROM (bit)	Dış Bellek (bit)
A.Devices ADSP2101	16	16x16	40	40	10	25	2Kx24 P 1Kx16 V	Yok	32Kx24 P 16Kx16 V
A.Devices ADSP2105	16	16x16	40	80	12.5	25	2Kx24 P 512x16 V	Yok	32Kx24 P 16Kx16 V
A.Devices ADSP2172	16	16x16	40	30	16.67	33	2Kx24 P 2Kx16 V	8Kx24 P	14Kx24 P 14Kx16 V
Lucent Tec. DSP1615	16	16x16	32	33	40	20	2Kx16	24Kx16	64Kx16
Motorola DSP56002	24	24x24	56 (96*)	30.3	66	33	512x24 P 256x24 V	256x24 P	64Kx24 P 64Kx24 V
Motorola DSP56005	24	24x24	56 (96*)	40	50	25	4608x24 P 256x24 V	256x24	64Kx24 P 64Kx24 V
Texas TMS320C25	16	16x16	32	100	40	10	544x16	4Kx16 P	128Kx16 P 128Kx16 V
Texas TMS320C50	16	16x16	32	35	50	20	10Kx16	Yok	64Kx16 P 64Kx16 V
Texas TMS320C53	16	16x16	32	50	40	20	4Kx16	2Kx16	64Kx16 P 64Kx16 V

TMS320C50, 16 bitlik yüksek hızlı programlanabilir bir tümdevredir. Düşük güç gerektiren CMOS teknolojisiyle üretilmiş olup, 132 pinli plastik kırmık (*chip*) taşıyıcısı içinde paketlenmiştir. Paralel hatlı yapısı sayesinde programlama kısıtlamaları veya güçlükleri yaratmadan yüksek verimle çalışır. Temel özellikleri aşağıdaki gibidir :

- CMOS teknolojisi olduğu için düşük güç tüketir ve tek 5 V güç kaynağına ihtiyaç duyar.
- 35 nsn komut çevrim süresine sahip en hızlı sabit-noktalı DSP'lerden birisidir.
- Bir komut çevrimi içinde 16x16 bitlik çarpma ve 32 bitlik toplama yapabilir.

- 32 bitlik akümülatör ve paralel lojik birim (*Paralel Logic Unit-PLU*) içerir.
- Yüksek hızlı, tekrarlı işlemler için komut ön belleği mevcuttur.
- Harici olarak 224K (64K program, 64K veri, 64K giriş/çıkış, 32K global) kelimeye uzatılabilen 10K kelimelik ROM ve 2K kelimelik RAM'a sahiptir.
- Seri ve paralel giriş/çıkış kapılarını kullanarak çoklu-işlemci yapısını destekleyebilir.
- Maskelenebilir kesme olanağı sağlar.

7.2.2 Kayan-Noktalı Uygulama

1984 yılında, AT&T, ilk tek tümdevreden oluşan kayan-noktalı programlanabilir işlemci olan DSP32'yi geliştirdi. Hemen ardından NEC'in kayan-noktalı DSP'si olan μ PD77230 piyasaya sürüldü.

Kayan-noktalı bir sayı, X , bir mantis (*mantissa*), M_x , ve bir üsten (*exponent*), E_x , oluşur ve genel formülü aşağıdaki gibidir :

$$X = M_x \times 2^{E_x} \quad (7.1)$$

İki kayan noktalı sayı, X ve Y 'nin çarpımından çıkan Z sonucu şöyle ifade edilebilir:

$$Z = M_x M_y \cdot 2^{E_x + E_y} \quad (7.2)$$

Bu, donanım olarak kayan-noktalı bir çarpıcının hem mantis için bir çarpıcı ve hem de üst için bir ilave toplayıcı içermesi gerektiği anlamına gelir. İki mantisin saklanması için genellikle ekstra doğruluk gerekir. Kayan-nokta veri formatı için bir IEEE standardı mevcut olsa da DSP imalatçıları genellikle kendi özel formatlarını seçmektedir. 100 ns veya daha düşük talimat çevrim zamanına sahip kayan-noktalı DSP'lerin karşılaştırması Tablo 7.2'de verilmiştir.

Tablo 7.2 Kayan Noktalı Sayısal İşaret İşlemci Örnekleri

(P: Program Belleği, V: Veri Belleği, M: Mantis, E: Exponent)

Firma Model	Veri uzunluğu (bit)	MAC giriş hassasiyeti (bit)	MAC çıkış hassasiyeti (bit)	Komut Çevrimi Süresi (ns)	Saat Frekansı (MHz)	İşlem Kapasitesi (MIPS)	Üzerindeki RAM (bit)	Üzerindeki ROM (bit)	Dış Bellek (bit)
A.Devices ADSP-21060	24M 8E	40x40	40	25	40	40	128Kx32	Yok	4Gx32
A.Devices ADSP-21061	24M 8E	40x40	40	20	50	50	32Kx32	Yok	4Gx32
Lucent DSP32C	24M 8E	32x32	40	20	50	20	1Kx32	2Kx32	16Mx8 P 4Mx32 V
Motorola DSP96002	24M 8E	32x32	44	20	60	30	2Kx32	1088x32	4Gx32 P 8Gx32 V
Texas TMS320C30	24M 8E	32x32	40	40	50	25	2Kx32	4Kx32	16Mx32
Texas TMS320C40	24M 8E	40x40	40	33	60	30	2Kx32	Reserve	4Gx32

Mevcut tipik kayan-noktalı DSP'lerin özelliklerini göstermek için aşağıda TMS320C40 kısaca açıklanmıştır. TMS320C40, Texas Instruments firması tarafından üretilen ve zaten güçlü ve beğenilen bir ürün olan TMS320C30'a göre önemli gelişmeler içeren yeni bir kayan-nokta sayısal işaret işlemcisidir [48].

- Yüksek hızlı iç iletişim için altı adet 20 MBit/s asenkron transfer hızlı iletişim portu vardır
- Giriş/Çıkışın birlikte olması için altı kanallı DMA eşilemcisi içerir. Her bir kanalın kendi kendine programlama kapasitesi mevcuttur.
- Ana DSP'nin merkezi işlemcisi 275 MOPS ve 320 MBit/s kapasitesindedir. CPU, 40 ns veya 50 ns'lik komut çevrimi başına onbir işlem yapabilmektedir. Tek çevrimde 40/32 bit kayan-noktalı çarpma yapabilir. Bölme ve ters karekök fonksiyonları donanım mantığı ile desteklenmiştir.
- 100 MBit/s'de veri transferi yapabilen birbirine eş harici veri ve adres yolu bulunmakta ve 16 GBit'lik program/veri/çevrebirim adres alanını desteklemektedir.
- Tümdevre kendi üzerinde paralel işleme için hata ayıklama (*debug*) imkanı veren bir analiz modülü içerir. Bu; program, veri ve DMA erişimi için ayrı durma

- Tümdevre kendi üzerinde paralel işleme için hata ayıklama (*debug*) imkanı veren bir analiz modülü içerir. Bu; program, veri ve DMA erişimi için ayrı durma noktası (*breakpoint*) karşılaştırıcıları sağlamakta ve hızlı hata ayıklama ve geliştirme için tümdevre üzerinde donanım olarak durma noktası imkanı oluşturmaktadır.

7.3 GERÇEK ZAMANLI ALGORİTMALAR VE YAZILIM GELİŞTİRME

Teorik algoritma geliştirme ve bunun gerçek zamanlı uygulaması, yüksek konuşma kalitesi gerektirdiği kadar cihaz maliyeti ve enerji tüketiminin de çok düşük olması gereken bir çok iletişim sisteminde aynı derecede önemlidir. Bu nedenle, algoritmanın optimize olması çok önemlidir; bu ise yazılım ve donanımının optimizasyonunu sağlar. DSP tümdevresi üzerinde bir görevin gerçek zamanlı uygulaması aşağıdaki adımları içermektedir :

- teori geliştirme ve iyileştirme,
- algoritmanın, yüksek seviyeli diller (C gibi) kullanılarak bilgisayar simülasyonlarıyla doğrulanması,
- yüksek seviyeli kodun ilgili DSP koduna çevrilmesi,
- mevcut geliştirme araçları kullanılarak gerçek zamanlı kodun doğrulanması
- uygun donanımın tasarlanması,
- son olarak da yazılım ve donanımın birleştirilmesi.

Geliştirme çalışmasının her bir adımı bir sonraki kadar önemli olup, her aşama kendine has çeşitli zorluklar içermektedir.

İkinci aşamada, algoritmanın doğrulanması, teorinin iyileştirilmesine veya çalışmanın tamamen yinelenmesine yol açabilir. Üçüncü aşamada, yüksek seviyeli kodun ilgili DSP koduna çevrilmesinde:

- bir DSP çapraz derleyicinin kullanılması (yani C dilinden makine diline)
- DSP komut seti kullanılarak algoritmanın doğrudan elle kodlanması

- (iii) kısmen elle kodlama ve kısmen çapraz derlemenin kullanılması (ilk iki seçeneğin karışımı)

olmak üzere üç yaklaşım izlenebilir.

Çapraz derleyicilerin tek başına bellek ve hız bakımından iki veya üç kat daha zayıf olmaları nedeniyle, bir çok konuşma kodlama uygulamasında doğrudan elle kodlama veya kısmen elle kodlama kısmen de çapraz derleme kullanılır. Dördüncü safhada, bilgisayar simülasyon sonuçları eşdeğer DSP çıktıları ile karşılaştırılır. DSP simülatörü aracılığıyla yüksek miktarda verinin işleminden geçirilmesi çok zaman aldığı ve doğru bulunmadığı için, bu işlem genellikle sınırlı miktarda girdi verisi için yapılır. İki çıktı arasında uyuşma sağlandıktan sonra, DSP yazılımının esas olarak doğru olduğundan emin olunabilir. Bunun ardından yazılım, gerçek zaman testi için hedef donanıma yüklenebilir. Bu noktada, hata ayıklama işleminin hızlandırılması için devre-üstünde emülasyon (ICE: *in-circuit emulation*) kolaylığı çok faydalı ve gerekli olacaktır. Algoritma çalışırken algortimanın parametreleri, en uygun performansın sağlanması için gerçek-zamanlı bir ortamda ayarlanır.

Çoğu DSP'de veri işleme süresini azaltmak için ardışık düzen (*pipeline*) mantığı kullanılmaktadır. Ardışık düzende bir komutun sonucu, bir sonraki komutta elde bulunmayabilir, bu nedenle DSP programlanırken, bu fonksiyonun da mümkün olduğunca etkin şekilde kullanılması gerekir. Tipik bir DSP, bellekten iki işlem ögesi alınarak çarpılması, bunların akümülatöre eklenmesi, sonucun belleğe yazılması ve sözkonusu üç adres kütüğünün sonradan arttırılmasına yönelik komutlara sahiptir. Bütün bu işlemlerin bir komut çevriminde birbirinin ardısıra yapılması durumunda komut çevriminin çok uzun olacağı ortadadır. Ardışık düzenli bir yapıda, işlemci komut yürütme görevini komut alma, komut kodunun çözülmesi, işlem ögesinin alınması ve işlemlerin yerine getirilmesi evrelerine ayırmaktadır. Bu, hesaplamayı hızlandırır da programlanabilirlik üzerinde önemli etkileri bulunmaktadır. Aşağıdaki örnekte, ardışık düzen kavramı daha ayrıntılı olarak anlatılmaya çalışılmıştır.

DSP aşağıdaki çarpma-toplama işlemini:

$$aN = aM + Y * X \quad (7.3)$$

X ve Y 'nin alınması, X ve Y 'nin çarpılması ve $X * Y$ sonucunun akümülatör aM 'in içeriği ile toplanması olmak üzere üç aşamada (belleğe yazmanın gerçekleştiği dördüncü aşama gözönünde bulundurulmamıştır) yerine getirmektedir. Bu işlemin yerine getirilmesi şöyle gerçekleşmektedir:

1. Adım- X ve Y 'nin alınması
2. Adım- Y ve X 'in çarpılması
3. Adım- Sonucun aM ile toplanması

Bu biçimdeki çeşitli komutlar birbiri ardına yürütüldüğü takdirde, işlemci, her bir komut çevriminde bir komutun tamamlanabilmesi için, talimatları otomatik olarak ardışık düzene sokar. Aşağıdaki komut seti ele alınırsa:

- (1) $aN = aM + Y1 * X1$
- (2) $aN = aM + Y2 * X2$
- (3) $aN = aM + Y3 * X3$
- (4) $aN = aM + Y4 * X4$

Ardışık düzene sokulan komut akışı aşağıda verilmektedir:

- (1) $Al_1 XY$
- (2) $\text{Çarp } Y1 \text{ } Al_2 XY$
- (3) $\text{Topla_1 } \text{Çarp } Y2 \text{ } Al_3 XY$
- (4) $\text{Topla_2 } \text{Çarp } Y3 \text{ } Al_4 XY$
- (5) $\text{Topla_3 } \text{Çarp } Y4$
- (6) Topla_4

Görüldüğü gibi bu yaklaşım, birinci komutun yürütülmesinin iki komut sonra biteceği anlamına gelir. Bu duruma sonucun gecikmesi (*latency*) denir ve yazılım hazırlanırken bunun etkilerinin bilincinde olmak programcının sorumluluğundadır. Bazı DSP'lerde, ardışık düzen programcı için saydamdır, ayrıca gözetmek gerekmez ve bazılarında da toplam verimin azalması karşılığında aktif olmaktan çıkarılabilir.

7.4 CELP GERÇEKLEME YAKLAŞIMLARI

Bu bölümde, etkin programlama teknikleri kullanılarak TMS320C50 sayısal işaret işlemcisi esaslı bir CELP kodlayıcısının gerçekleştirilmesi ele alınmıştır [51]. Bazı uygulamalarda, genellikle etkin programlama teknikleri uygun miktarda hız ve bellek indirimini yaratmak için yeterli olmamakta, algoritmik sadeleştirmelerle tamamlanması gerekmektedir. Bu nedenle, CELP kodlayıcısı için, gerekli hız ve bellek kapasitesini azaltmak için çeşitli algoritmik sadeleştirmeler önerilmiştir. Ayrıca son bölümde, CELP kodlayıcıların AbS çevrimi için önerilen geliştirme önerilerine de değinilmiştir. Tüm bu çabalar sonucunda oldukça karmaşık bir yazılım gerektiren CELP kodlayıcısı TMS320C50 gibi sabit noktalı bir sayısal işaret işlemci üzerinde gerçekleştirilmiş ve uygulama DSK geliştirme ortamında denenmiştir. Yüksek seviyeli programlama dili C ile kodlanan yazılıma ait önemli rutinler Ek-B'de verilmiştir.

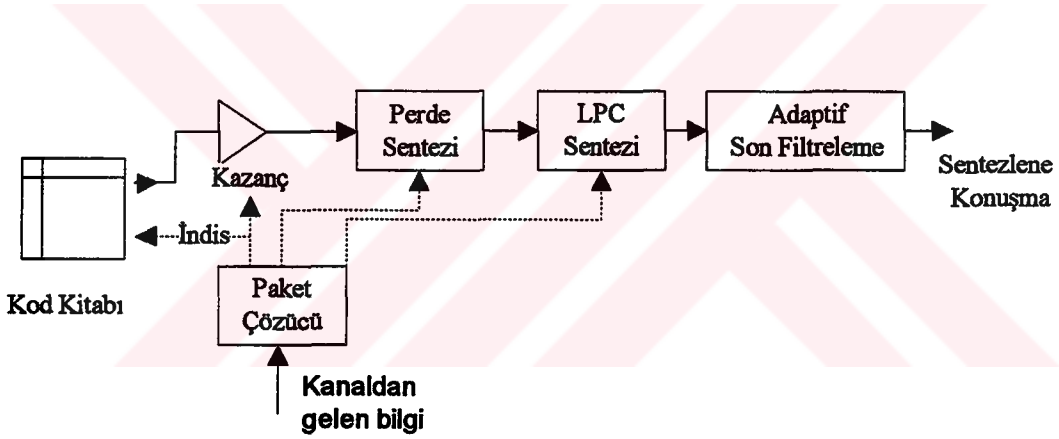
7.4.1 Gerçeklenen 4.8 kbit/san CELP Kodlayıcının Yapısı

Yukarıda söz konusu edildiği gibi CELP kodlayıcılar, düşük bit hızlarında yüksek kalite sağlarken, oldukça yüksek bir hesaplama gücüne gereksinim duyar. Standartlarda tanımlanan CELP sistemini olduğu gibi gerçeklemek için, saniyede birkaç yüz milyon işlem kapasitesi gerekmektedir. Bundan dolayı CELP teknikleri üzerindeki araştırmaların çoğu, bu hesap yükünü azaltacak ve gerçek zamanlı çalışmayı sağlayacak basitleştirme yöntemleri üzerine yoğunlaşmıştır.

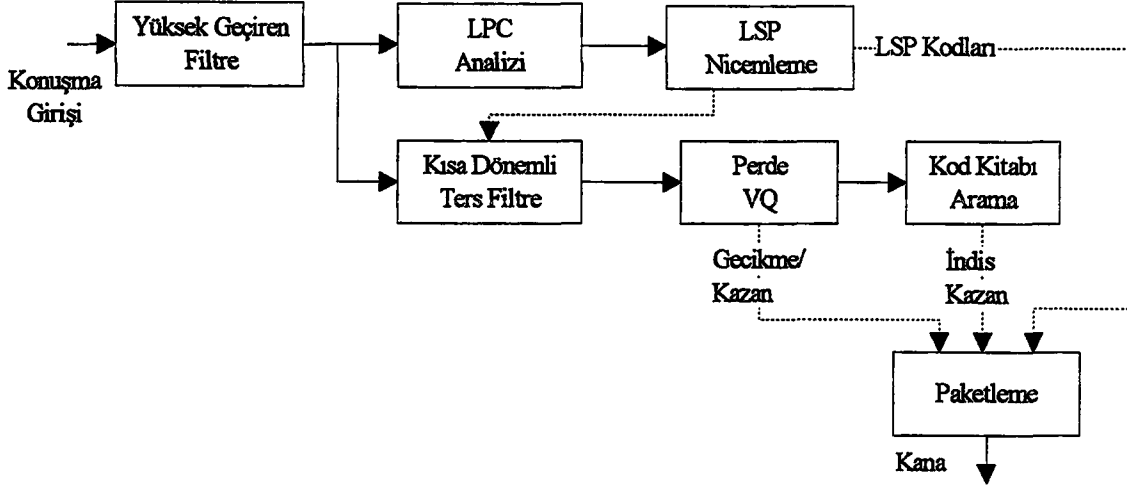
Burada gerçekleştirilen kodlayıcının temel karakteristikleri aşağıdaki gibidir:

- 8 kHz örnekleme hızında 240 örneklik çerçeve boyu
- 10. dereceden kısa dönemli öngörücü
 - Her çerçeve başına açık çevrim olarak bir kere hesaplanmakta
 - Hamming penceresi ile özilişki
 - LSP nicemleme
- Dört alt çerçeve (60 örnek)
 - Bir ara-uçlu perde öngörücü
 - 1) Kapalı çevrim analiz
 - 1024 elemanlı kod kitabı
 - 1) 2 elemanı örtüşen
 - 2) Elemanların %75'i sıfır

Kod çözücü ve kodlayıcıya ait blok şemalar aşağıda verilmektedir:



Şekil 7.2 CELP Kod Çözücü



Şekil 7.3 CELP Kodlayıcı

Tasarlanan CELP kodlayıcıya ait bit atamaları Tablo 7.3’de verilmektedir.

Tablo 7.3 4.8 kbit/san CELP parametreleri

	Spektrum		Perde		Kod Kitabı	
Güncelleme	30 msan (240 örnek)	(240 örnek)	7.5 msan (60 örnek)	(60 örnek)	7.5 msan (60 örnek)	(60 örnek)
Parametreler	10 LSP		1gecikme, 1 kazanç		1 indis, 1 kazanç	
Bit/san	1133.3		1466.7		2000	

Not: kalan 200 bps genişleme, hata koruma ve eşleme için ayrılmıştır

Algoritmanın en fazla hesap yükü gerektiren kısımları, her çerçevede dört kere tekrarlanan, perde ve kod kitabı arama kısımlarıdır.

7.4.1.1 Giriş/Çıkış

Konuşma örneklerinin elde edilmesi için iki yol kullanılmıştır, geliştirme sırasında sayısallaştırılmış konuşma örneklerinden oluşan diskte kayıtlı veri kullanılmış,

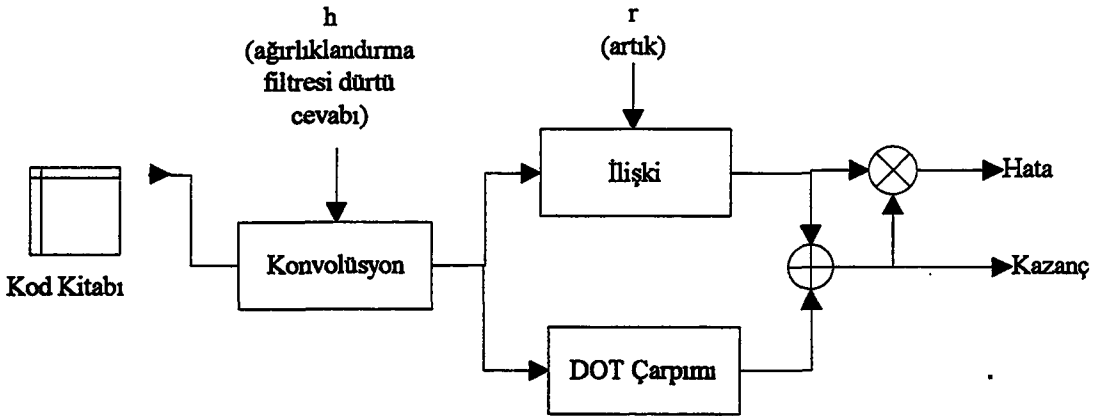
gerçek zamanlı uygulamada ise DSK modülü üzerinde bulunan analog/sayısal dönüştürücü ile çalışılmıştır.

7.4.1.2 Spektrum Analizi

İşlenecek işaret yüksek geçiren filtreden geçirildikten sonra, kısa dönemli öngörü filtresinin katsayıları bulunmaktadır. Vektör seçiminde ilgilenilen bölgeyi belirlemek için *Hamming* tipi bir pencere kullanılmıştır. Özilişki değerleri kısa dönemli filtrenin derecesi uzunluğunda bir vektörde saklanmaktadır. Ters filtre katsayılarının bulunması için *Levinson-Durbin* türü bir algoritma seçilmiştir. Daha etkin bir nicemleme yapılabilmesi için, öngörü katsayıları çizgi spektrum çiftlerine (LSP) dönüştürülmüş ve bu değerler nicemlenmiştir. Nicemlenmiş LSP değerleri daha sonra kısa dönemli öngörücüde kullanılmak için geri dönüştürülmektedir.

7.4.1.3 Perde ve Kod Kitabı Arama

Kısa dönemli öngörü bulunduktan ve buna ait artık üretildikten sonra, her dört alt çerçeve için perde öngörücü ve kod kitabı parametreleri bulunmaktadır. Perde ve kod kitabı arama işlemleri benzer şekilde gerçekleştirilmiştir; yani her iki aramada da bir hata terimi bir küme değeri için irdelenmekte ve minimize edilmeye çalışılmaktadır.

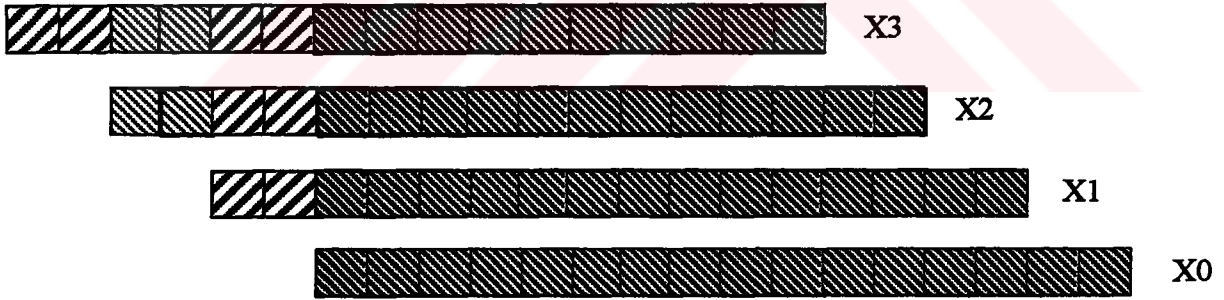


Şekil 7.4 Kod Kitabı Arama Blok Diyagramı

Şekil 7.4’de verilen kod kitabı aramaya ait blok şemada giriş vektörü ile üretilen birçok vektörden biri arasındaki uzaklık minimize edilmeye çalışılmaktadır. Minimize edilmeye çalışılan büyüklük *Euclid* normundadır ve aşağıdaki gibi ifade edilebilir:

$$\begin{aligned} e &= \|r - \hat{r}\|^2 \\ &= r'r - 2r'\hat{r} + \hat{r}'\hat{r} \end{aligned} \quad (7.4)$$

Burada r orijinal artık, $\hat{r}$ ise sentezlenmiş artıktır. Vektör tanımından hareketle yalnızca r ile $\hat{r}$ ’nin ilişkisi ve $\hat{r}$ ’nin enerjisinin hesaplanması gerektiği sonucuna varılabilir. Görülebilir ki her alt çerçeve için N adet konvolüsyon ve $2N$ adet dot çarpımı gerekmektedir. Buradan hareketle, eğer doğrudan gerçekleştirme kullanılsaydı $N=256$ ve alt çerçeve uzunluğu 60 için 66 MIPS’lik bir işlem kapasitesine ihtiyaç duyulmaktadır. Bunun yerine Şekil 7.5’de gösterilen üstüste binmiş kod kitabı yapısı kullanılarak hesap yükü azaltılmıştır. Şekildeki gibi bir kod kitabının kullanılması ile her alt çerçeve için yalnızca bir konvolüsyon işlemi gerekmektedir.



Şekil 7.5 Üstüste Binmiş Kod Kitabı Yapısı

Bu yapı özyinelemeli türden bir konvolüsyon hesabına izin vermektedir. İlk kod kitabı vektörü normal olarak ağırlıklandırma filtresi çıkışı ile çarpılmakta, sonraki hesaplamalar için ise aşağıdaki denklemler kullanılmaktadır:

$$\begin{aligned} V_{i+1}(z) &= z^{-1}\hat{R}_i(z) + x_{i+1}[1]H(z) \\ \hat{R}_{i+1}(z) &= z^{-1}V_{i+1}(z) + x_{i+1}[0]H(z) \end{aligned} \quad (7.5)$$

Burada $\hat{R}_i(z)$ üretilen artık işaretin z dönüşümüdür. Böylelikle bir önceki kod kitabı vektörünün ağırlıklandırma filtresi ile konvolüsyonu bilindiğinde, bir sonraki vektör için konvolüsyon yalnızca 120 (2 x 60) toplama-çarpma ile bulunabilir.

Bu sayı kod kitabının başka bir özelliği kullanılarak daha da azaltılabilmektedir. Kod kitabı vektörleri, *Gauss* tipi gürültü kaynağı kullanılarak merkezden katlanmış olarak üretilirse, vektörlerin yaklaşık %75'i sıfır olur. Bu şekilde güncellemelerin %75'i için toplama-çarpma işlemi gerekmez, yalnızca içeriğin kaydırılması yeterli olur. İşlemci üzerinde bulunan dairesel tampon bellekler kullanılarak bu işlem basitçe yapılabilmektedir. Konvolüsyon işlemi tamamlandıktan sonra, buna ilişkin hat ve kazanç bulunabilir. Her hesaplama için oluşan hata o andaki maksimum ile karşılaştırılır, eğer güncel hata daha büyükse, yeni en iyi vektör olarak güncel vektör seçilir. Bu işlem her N vektör için tekrarlanmaktadır. Arama tamamlandıktan sonra en iyi vektör için kazanç hesaplanmış ve nicemlenmiş olur. Sonuç olarak hesaplanan kazanç ve kod kitabı elemanının indis elde edilir.

7.4.2 TMS320C50 Sabit Noktalı Sayısal İşaret İşlemci

TMS320C50 yüksek performanslı, programlanabilir sayısal işaret işlemcisi; oldukça esnek komut kümesi, gelişkin paralel mimarisi ve fiyat avantajı ile önemli iletişim uygulamalarında rahatlıkla kullanılabilen bir tümdevredir [49]. 16-bitlik sabit noktalı işlemleri destekleyen işlemci, CMOS teknolojisi ile gerçekleştirildiği için düşük enerji tüketimiyle yüksek fonksiyonel yoğunluk sağlamaktadır. 35 nsan'lık tek çevrimli komut seti ile saniyede yaklaşık 28 milyon komutu yürütebilmektedir (28.6 MIPS). Temel özellikleri aşağıda verilmiştir:

- Tümdevre üzerinde komut ve verinin saklanması için 9K x 16 bit uzunluğunda RAM bellek bulunmaktadır. Ayrıca 2K x 16 bitlik önyükleme ROM belleği ve 1056 x 16 bitlik çift kapılı RAM belleği tümdevre üzerine yerleştirilmiştir.

Bellek, tümdevre dışında 64K program, 64K veri, 64K giriş/çıkış ve 32K global olmak üzere 224K x 16 bit genişletilebilmektedir.

- 32 bit aritmetik-mantık birimi (ALU), 32 bit toplayıcı (*accumulator-ACC*) ve 32 bit toplayıcı önbelleğinin (ACCB) yanısıra, 16bit paralel mantık birimi (PLU) ve 32 bit sonuç verebilen 16bit x 16 bit paralel çarpıcı bulunmaktadır. Tüm çarpma ve toplama işlemleri tek çevrimde yerine getirilmektedir.
- Aritmetik işlemler için kendi aritmetik birimi de bulunan 8 adet harici işlem kütüğü ve kesme hizmeti için 11 adet bağlam saklamada kullanılan gölge kütük bulunmaktadır.
- Sekiz düzeyli donanım olarak gerçekleşmiş yığın, 2 adet 0'dan 16'ya kadar kaydırma kapasiteli *barrel-kaydırıcı* ve 64 bit artırımı (*incremental*) veri kaydırıcı sağlanmaktadır.
- Diğer seri haberleşmeyi destekleyen birimlerle haberleşmek için, tam çift yönlü (*full duplex*) senkron zaman paylaşmalı çoklu erişim (*time division multiple access-TDM*) sağlayan seri iletişim arabirimi sağlanmıştır.
- TMS320C50 aynı zamanda şu fonksiyonları desteklemektedir :
 - (i) 64K paralel giriş/çıkış portu
 - (ii) saat üretimi
 - (iii) emülasyon ve test amaçlı JTAG sınır tarama (IEEE 1149.1)
 - (iv) Tümdevre üzerinde tarama bazlı emülasyon imkanı

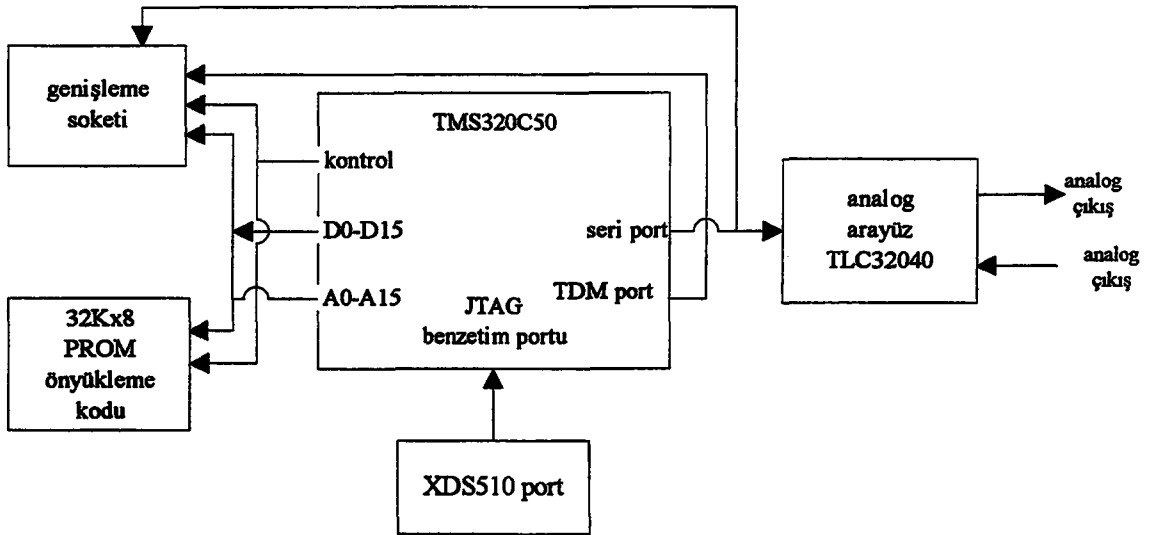
7.4.3 TMS320C50 Esaslı DSK Donanımı

Bu bölümde üzerinde CELP uygulaması geliştirilmiş olan TMS320C50 esaslı DSP başlangıç kiti (*DSP Starter Kit-DSK*) kısaca tanıtılacaktır. DSK donanımı, çevirici

(*assembler*) kodunu gerçek hızında test etmek için üzerinde bir TMS320C50 işlemcisi içermektedir [50]. DSK'nın dikkat çekici özellikleri olarak:

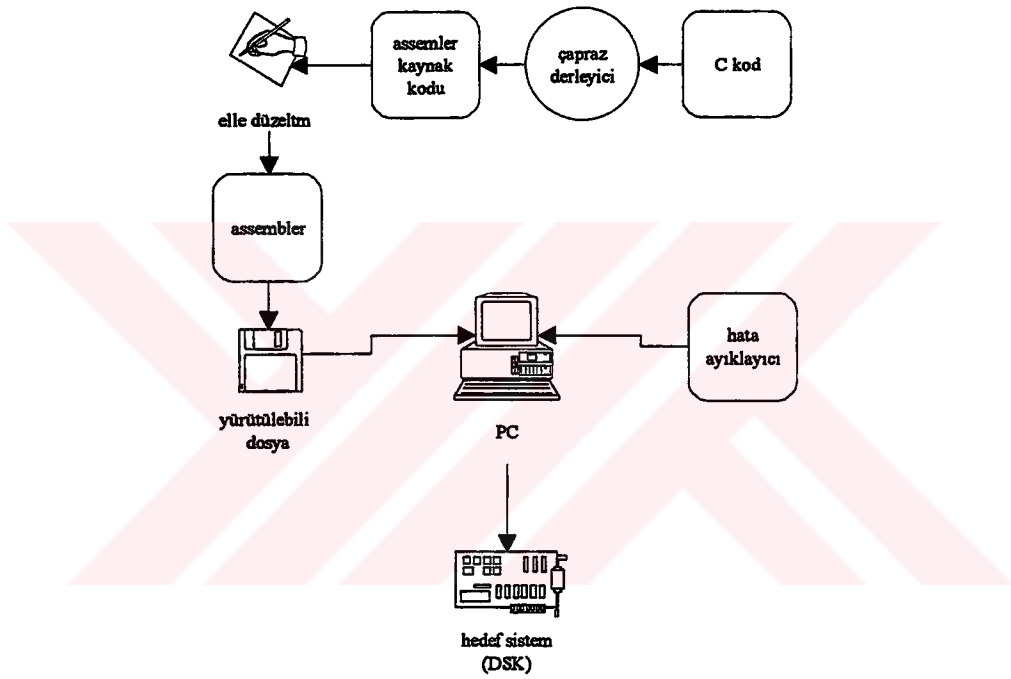
- endüstri standardı TMS320C50 sabit noktalı DSP
- 35 nsan komut çevrim süresi
- 32 Kbyte PROM
- ses kalitesinde analog veri girişi için arayüz (TLC320040)
- doğrudan mikrofon ve kolon bağlanabilme özelliği
- XDS510 emülatör bağlantısı
- I/O yolu genişletme imkanı

sayılabilir. Şekil 7.2'de DSK'nın işlevsel blok şeması verilmektedir. DSK TMS320C50 işlemcisinin en basit uygulama devrelerinden birisidir. Devrede bulunan 10 Kbayt'lık RAM bir çok DSP uygulaması için yeterli olmaktadır. Çekirdek program 32 Kbayt'lık PROM'a yazılmış durumdadır. Devre üzerinde bulunan TLC32040 tümdevresi ses kalitesinde tek kanal giriş/çıkış analog arayüz sağlamaktadır. Tek tümdevre ile A/D, D/A dönüştürme işlemlerini 14 bitlik bir dinamik ile sağlamak ve D/A, A/D için programlanabilir örnekleme hızına ve filtreleme imkanlarına sahiptir.



Şekil 7.6 TMS 320C50 DSK Blok Şema

DSK donanımı seri port üzerinden bir kişisel bilgisayara bağlanabilmekte ve pencere bazlı hata ayıklama yazılımı sayesinde, yazılım geliştirme işlemi oldukça kolaylaştırılmaktadır. Şekil 7.3’de yazılım geliştirmedeki akış verilmiştir. Yüksek seviyeli programlama dili C ile yazılan kod, çapraz derleyiciden geçirildikten sonra elde edilen kod yukarıda anlatıldığı gibi el ile düzeltilmiş ve böylelikle kaynakları daha iyi kullanan, etkin bir kaynak kodu elde edilmiştir. Bu kaynak kodu makina koduna dönüştürülerek hedef donanımda çalıştırılabilecek ve hata ayıklayıcı ile akışı izlenebilecek yürütülebilir bir dosya oluşturulmuştur.



Şekil 7.7 DSK Yazılım Geliştirme Akışı

7.4.4 Etkin DSP Programlama Teknikleri

C kodundan kaynak kodu elde edildikten sonra, el ile düzelterek etkin bir kod yaratmak için dikkat edilen noktalar aşağıda sıralanmıştır:

- Herhangi bir şarta bağlı komut içermeyen tekrarlı fonksiyonlar için mümkün olduğu kadar genel amaçlı döngüler kullanılmıştır.
- DSP'nin RAM'ında bulunan veri önbellekleri, mümkün olan en az sayıda "veri aktarma" komutu gerektirecek şekilde yeniden düzenlemiştir.
- TMS320C50 kodunda, beklemeyi en aza indirmek amacıyla mümkün olduğunca "NOP" komutu kullanılmamıştır.
- Tamsayı verilerin blok halinde bir yerden diğerine aktarılması için en etkin (en az komut gerektiren) yöntem kullanılmıştır.
- Hesaplama mümkün olduğunca program akışı dışında yapılmıştır. Diğer bir deyişle, işlem süresinden kazanabilmek için hesaplama yerine, bellek içerisinde bulunan referans tablolar kullanılmıştır.
- Program ve veri önbellekleri, birbiriyle uyuşmayan en az sayıda bekletme oluşturulacak şekilde farklı bellek bölgelerine yerleştirilmiştir. Böylelikle her bekletme durumu için yaklaşık 20 nsan'lık ilave bir işlem süresinin ortaya çıkmasına engellenmiştir.
- Bölme veya karekök alma gibi fonksiyonların çok kullanılması durumunda, bu fonksiyonlar alt-program olarak gerçekleştirilmemiştir; çünkü bu durumda alt-programın her çağrılışında 6-10 adet gereksiz komutun yürütülmesi zorunlu olacaktır. Bunun yerine, alt-program değiştirilmiş ve bu ana-programın içine yerleştirilmiştir.

7.4.5 CELP Karmaşıklık İndirgeme Stratejileri

Tipik bir CELP konuşma kodlama algoritmasında, işlem süresinin çoğunluğu, kapalı çevrim LTP ve kod kitabı arama süreçlerine harcanmaktadır. Bu nedenle aşağıda, etkin uygulama için, LTP ve kod kitabı arama karmaşıklığı azaltma stratejileri

araştırılmış ve bazı önerilerde bulunulmuştur. Bir CELP kolayıcıda kapalı çevrim LTP ve kod kitabı arama için oluşan genel karmaşıklık üç bölümde incelenebilir. İlk olarak, LTP ve kod kitabından alınacak her bir uyarma adayının filtreden geçirilmiş cevabının bulunması için konvolüsyon hesaplamalarına gerek duyulmaktadır. Bunun ardından her bir sentezlenmiş çıktının referans işaretiyle çapraz ilişkisi bulunur ve son olarak normalizasyon ve en uygun vektör kazanç hesaplaması için her bir sentezlenmiş cevabın özilişkisi bulunur. Basitleştirme önerileri de bu doğrultuda üç ana başlık altında incelenmektedir.

7.4.5.1 Konvolüsyonun Karmaşıklık İndirgemesi

LTP uyarma adaylarının aranması sırasında, her bir uyarma vektörü bir öncekine göre, örnek vektörlerin başında ve sonunda olmak üzere iki konumda farklılık gösterir. Bu nedenle, mevcut vektörün çıkış cevabı bir önceki vektörün kaydırılmış biçiminin ve o andaki vektöre yeni örneğin katkısının toplamı olarak tanımlanırsa, konvolüsyon işlemi büyük oranda sadeleştirilmiş olur.

Örneğin, $h_w(n)$ ağırlıklandırılmış STP dürtü cevabı ise, LTP belleğinden k . uyarma vektörü $\hat{r}(n)$ (k gecikmesine karşılık denk düşen vektör) nedeniyle sentezlenen $\hat{s}_k(n)$ vektörü:

$$\hat{s}_k(n) = \sum_{i=0}^n h_w(i) \hat{r}(n-i-k) \quad (7.4)$$

şeklinde ifade edilebilir. Ardışık iki vektör arasındaki fark, sadece iki vektörün her iki ucunda bir örnek olduğu için, sentezlenmiş vektör $\hat{s}_{k+1}(n)$, $\hat{s}_k(n)$ cinsinden aşağıdaki gibi yazılabilir:

$$\hat{s}_{k+1}(n) = \hat{r}(-k-1)h_w(n) + \hat{s}_k(n-1) \quad (7.5)$$

Bu formülde kullanılan $\hat{s}_k(-1) = 0$ 'dır. Yukarıdaki ifadeden de görülebileceği gibi, bir önceki çıktıyı sağa doğru bir örnek kaydırarak ve bunu, yeni örnekle ölçeklenmiş STP'nin dürtü cevabına ekleyerek konvolüsyon hesaplarının çoğunluğu sadeleştirilebilir.

Eğer kullanılan kod kitabı LTP ile aynı özelliklere sahipse kod kitabı araması sırasında konvolüsyonların sadeleştirilmesinde aynı yöntem kullanılabilir. Bununla birlikte, kod kitabındaki kaydırmaların sayısı arttıkça, karmaşıklık da artar ve 7.6 nolu genel ifadeden de anlaşılabileceği gibi, R . kaydırma vektör büyüklüğüne eşit olduğunda, kod kitabı da bağımsız vektörleri içeren standart bir vektör haline gelir :

$$\hat{s}_{k+1}(n) = \begin{cases} \sum_{i=0}^n h_w(i)x_{k+1}(n-i) & 0 \leq n < R \\ \sum_{i=0}^{R-1} h_w(n-i)x_{k+1}(i) + \hat{s}_k(n-R) & n \geq R \end{cases} \quad (7.6)$$

Merkez kırpma (*centre clipping*) kullanılıyorsa, ilk vektörün sentezi sırasında $x_k(n)$ 'in sıfır değerlerinin $h_w(i)$ ile çarpılmasına ve bu nedenle hesaplama karmaşıklığının azaltılmasına gerek kalmayacaktır. Bununla birlikte, ilk vektör tam olarak sentezlendikten sonra, (7.6) denklemi kullanılarak daha fazla tasarruf yapılabilecektir. Tek kaydırmalı bir kod kitabı kullanıldığı varsayılarak, yeni vektörü oluşturmak üzere sıfır değerli her başlatma örneği denkleme eklendiğinde, denklemin ilk terimi sıfır olacaktır.

7.4.5.2 Özilişkinin Karmaşıklık İndirgemesi

Sentezlenmiş vektör cevaplarının özilişkisi hesaplanırken, kapalı-çevrim LTP ve kod kitabı aramasının karmaşıklığını azaltmak için aşağıdaki sadeleştirme işlemi önerilmektedir:

- (i) Yeni vektör için özilişki hesaplanırken, yeni örnek sıfır (yani üstüste binen merkezi kırılmış kod kitabı), bir önceki özilişki değeri kullanılır ve bir önceki vektörün son örneğinin karesi bu değerden çıkartılır.
- (ii) İlk alt-çerçeve için bütün özilişki değerleri kaydedilir ve bunlar sonraki alt-çerçevelerde kullanılır. Burada, bütün alt-çerçeveler için STP parametrelerinin aynı olduğu varsayılmaktadır. STP parametreleri aradeğerleme yapılırsa, hesaplanan vektörün anlık kazancı önemli derecede etkilenebilir. Bununla birlikte, faz bilgisi etkilenmez. Bu nedenle, kod kitabı indeksinin seçiminden sonra, o vektörün en uygun kazancını hesaplamak için son sentez yapılır. Böylece, özilişki değerlerinin saklanması için kullanılan küçük bir ek bellek harcanarak işlem süresinde önemli azalma sağlanır.

LTP vektörleri, her bir alt-çerçeveden sonra değiştirildiği için, yukarıda tanımlanan karmaşıklık azaltma yaklaşımı doğrudan uygulanamamaktadır. Bu durumda, alt-çerçeve ölçüsünden büyük veya eşit indeks değerine sahip vektör cevaplarının, uyumlaştırma ile değişmeyen özilişki değerleri kullanılabilir. Bununla birlikte, LTP güncelleştirildiğinde, yeni numuneler, LTP'ye verilerek kullanılmayan özilişki değerlerinin sayısı, alt-çerçeve büyüklüğünün iki katına çıkarılır. Bu nedenle, vektör uzunluğu ve dolayısıyla alt-çerçeve ölçüsü artırılarak bit hızı azaltıldığında bu yöntemin etkinliği azalmaktadır. Örneğin, 30, 40 ve 60 büyüklüğündeki vektörler için özilişki kazanımı sırasıyla $147-60 = 87$, $147-80 = 67$ ve $147-120 = 27$ olmaktadır.

7.4.5.3 Çapraz-İlişkinin Karmaşıklık İndirgemesi

Konvolüsyon ve özilişki hesaplamalarının karmaşıklığı bir çok şekilde azaltılabilir de yapay ve referans vektörlerin çapraz-ilişki hesaplaması yukarıda verilen yöntemlerin hiç birisiyle sadeleştirilemez. Bunun asıl sebebi, referans vektörünün fazı sabit kalsa da, her bir yapay vektörün faz konumunun değişebilmesidir. Bu nedenle, bir vektör için hesaplanan çapraz-ilişki, bir sonraki vektörün çapraz-ilişkinin hesaplanmasında kolaylıkla kullanılamaz.

Aşağıdaki örnekte, çapraz-ilişkinin karmaşıklığını azaltan yeni bir kod kitabı arama algoritması tanımlanmıştır. Çapraz-ilişki karmaşıklığındaki azalma, herbir kod kitabı vektöründeki sıfır değerli örneklerin sayısına bağlıdır. Bu yöntemde, karmaşıklık STP dürtü cevabının uzunluğuna bakılarak belirlenir. Kod kitabındaki k 'inci vektörün yapay çıktısı:

$$\hat{s}_k(n) = \sum_{i=0}^n h_w(i) x_k(n-i) \quad 0 \leq n \leq L-1 \quad (7.7)$$

şeklinde ifade edilirse, burada kullanılan $x_k(n)$, k endeksi ile adreslenen L elemanlı kod kitabı uyarma vektörüdür. (7.7)'den görülebildiği gibi, çıktı yapay vektörü, ölçeklendirilen ve uyarma darbe genlikleri ve fazları ile en doğru şekilde konumlandırılan ağırlıklı STP impuls cevaplarının, $h_w(n)$, toplanması ile elde edilir.

Kod kitabındaki k 'inci vektörden kaynaklanan yapay ve referans vektörlerinin çapraz-ilişkisi θ_k , aşağıdaki gibi tanımlansın:

$$\theta_k = \sum_{n=0}^{L-1} \tilde{s}(n) \hat{s}_k(n) \quad (7.8)$$

(7.8) ifadesi, (7.7) denkleminde yerine konulursa,

$$\theta_k = \sum_{n=0}^{L-1} \left[\tilde{s}(n) \sum_{i=0}^n h_w(i) x_k(n-i) \right] \quad (7.9)$$

elde edilir. (7.9) da şu şekilde yazılabilir:

$$\theta_k = \sum_{n=0}^{L-1} \left[x_k(n) \sum_{i=0}^{L-1-n} h_w(i) \tilde{s}(i+n) \right] \quad (7.10)$$

Aşağıdaki formülde verilen bir $c(n)$ değişkeni tanımlanır:

$$c(n) = \sum_{i=0}^{L-1-n} h_w(i) \tilde{s}(i+n) \quad 0 \leq n \leq L-1 \quad (7.11)$$

ve (7.10)'da yerine konulursa:

$$\theta_k = \sum_{n=0}^{L-1} x_k(n)c(n) \quad (7.12)$$

elde edilir.

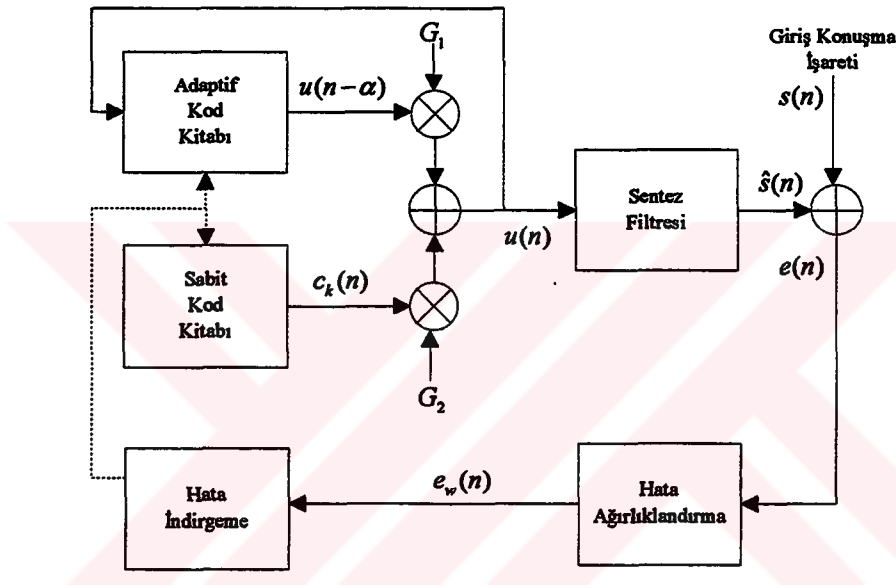
$x_k(n)$ uyarma vektörünün sıfır değerli elemanları varsa, yukarıda verilen çapraz-ilişkiler, önemli derecede sadeleşir. Tipik bir merkez kırpmalı kod kitabının kullanıldığı varsayılırsa, (7.11) denklemi kullanılarak bölünmüş, ağırlıklandırılmış STP dürtü cevabının, referans vektörü ile olan $c(n)$ başlangıç ilişkilerinin hesaplanmasından sonra elde edilen her bir θ_k çapraz-ilişkisi (7.12) denkleminde verilmiştir. Burada yalnızca sıfır değerli olmayan elemanların toplama yansıyacağına dikkat edilmelidir. Örneğin, 4 adet sıfır olmayan elemanlı, toplam 40 elemandan oluşan bir vektör kullanıldığında, her bir ilave kod kitabı için, ilgili θ_k değerini hesaplamak için sadece 4 adet çarpma-toplama işlemine gereksinim duyulacaktır. Kod kitabı araması sırasında, birinci alt-çerçevenin özilişki değerleri kullanıldığı takdirde, bu yöntem, standart bir kod kitabı araması için gerekli olan $\hat{s}(n)$ yapay çıktısının hesaplamasında gereksinim duyulan bütün konvolüsyon hesaplamalarını ortadan kaldırır.

Bu yöntemdeki temel hesaplama (7.11) denklemi kullanılarak $c(n)$ 'in bulunması olmasına rağmen, bu hesaplama alt çerçeve başına bir kere yapılmakta ve kod kitabının büyüklüğü ile orantılı olmamaktadır. Böylelikle kod kitabı büyüklüğü arttıkça bu yöntem ile sağlanan kazanç artmaktadır.

7.4.6 CELP Kodlayıcının AbS Çevrimi için Önerilen Geliştirmeler

Burada CELP kodlayıcıların AbS (sentezleyerek analiz) çevrimi için önerilen geliştirmeler tartışılacaktır. Bu geliştirmeler sonucunda özellikle kod kitabı arama performansında önemli iyileştirmeler gözlenmiştir. Bu da sistemin toplam etkinliğini olumlu yönde etkilemiş ve uygulama platformundan beklenen kaynak kapasitesinin azalmasına yol açmıştır.

Genel bir CELP kodlayıcıya ait blok şema Şekil 7.4’de verilmektedir. Uyarma işareti $u(n)$, özellikle ünlü konuşmadaki uzun dönemli periyodiklikleri ekleyen ölçeklenmiş adaptif bir kod kitabı ile, yine ölçeklenmiş büyük bir sabit kod kitabından elde edilen elemanların toplamından oluşmaktadır. Elde edilen bu uyarma, ağızdaki ses üretimine yarayan boşluğun etkilerini modelleyen bir sentez filtresinin girişine uygulanmaktadır. Kodlayıcıda tipik olarak öncelikle filtre parametreleri elde edilmekte, sonra kod kitabı endeksleri α ve k ile, G_1 ve G_2 kazançları bulunmaktadır.



Şekil 7.8 CELP Kodlayıcı Prensi Şeması

Kod kitabı parametreleri, orijinal ve yeniden elde edilen konuşma işareti arasındaki ağırlıklandırılmış hatayı en aza indireyecek şekilde seçilir. Aslında, orijinal konuşma işaretine en yakın çıktıyı bulmak için kod kitabındaki tüm değerler sentez filtresinden geçirilmelidir. Bu oldukça zahmetli kapalı devre yaklaşımı, orijinal konuşmaya mümkün olduğu kadar yakın bir değer elde etmek için kullanılmaktadır.

Fakat bir çok CELP kodlayıcısında bu zahmetli kapalı devre yaklaşımını uygulamak için iki önemli özel hal bulunmaktadır. Bunlardan birincisi, öngörü artık hatasının enerjisini en aza indiren kısa dönemli lineer öngörü hata filtresinin tersi olarak kabul edilen, sentez filtresinin belirlenmesidir. Bu demektir ki, $u(n)$ uyarma işareti

türetilirken sentez filtresinin biçimi dikkate alındığı halde, sentez filtresinin parametreleri belirlenirken uyarma işaretinin biçimine hiç dikkat edilmemektedir. Bu da yordamdaki bir açık olarak göze çarpmaktadır ve aslında sentez filtresinin konuşma işaretindeki uzun dönemli periyodiklikleri daha iyisi adaptif kod kitabına bırakacağına kendisinin dikkate alması sonucunu doğurur.

Sıkı bir kapalı devre yaklaşımdan ikinci farklılık ise kod kitabı parametrelerinin belirlenmesinde ortaya çıkmaktadır. Adaptif ve sabit kod kitabı parametreleri ağırlıklandırılmış hata işaretinin minimumunu elde etmek üzere birlikte belirleneceği yerde, öncelikle adaptif kod kitabına ait gecikme ve kazançlar sabit kod kitabı işareti sıfır varsayılarak belirlenmektedir. Daha sonra adaptif kod kitabı işareti varken sabit kod kitabı parametreleri bulunmaktadır. Uygulanan bu tam olarak optimum olmayan model, CELP kodlayıcıların karmaşıklıklarını indirgemek için yeterli olmasına rağmen, yeniden elde edilen konuşma işaretinde bir takım bozulmalara yol açmaktadır.

Çalışmada, yukarıda açıklanan iki yaklaşım sonucunda konuşma işaretinde ortaya çıkan bozulmalar incelenmiştir. Aynı zamanda belirli bir karmaşıklık düzeyinde kalarak yeniden elde edilen konuşmanın kalitesini arttıracak çeşitli yaklaşımlar önerilmiştir. Önerilen yaklaşımlar yukarıda söz konusu edilen donanım üzerinde denenmiş ve olumlu sonuçlar elde edilmiştir.

7.4.6.1 Sentez Filtresi Parametrelerinin Hesabı

Yukarıda bahsedildiği gibi, sentez filtresi genellikle $s(n)$ giriş konuşma işareti için öngörü artık enerjisini en aza indiren öngörü hata filtresi $A(z) = 1 - \alpha_1 z^{-1} - \alpha_2 z^{-2} \dots \alpha_p z^{-p}$ 'nin tersi olarak alınır. Burada p filtrenin derecesidir ve 10 olarak alınmaktadır. Uyarma işareti $u(n)$ belirlendikten sonra yeniden elde edilen konuşmaya ait SNR'ın iyileştirilmesi amacıyla sentez filtresi parametreleri yeniden hesaplaabilir. Aşağıda bu optimizasyon için çeşitli yöntemler verilmektedir.

Sentez filtresi parametrelerinin güncellenmesinin etkilerini bulmak için, önce filtre katsayıları nicemleme amacıyla Çizgi Spektrum Katsayıları'na (*Line Spectrum Frequencies-LSF's*) dönüştürülmüş ve her konuşma çerçevesinin en uygun nicemlenmiş LSF'lerinin bulunması için yöntemler araştırılmıştır.

Yeniden optimizasyon için denenen yöntemlerden birisi de en küçük karelerin güncellenmesidir (*least squares update*). Uyarma işareti $u(n)$ ve a_k $k=1,2,\dots,p$ filtre katsayıları verildiğinde, yeniden elde edilen konuşma işareti $\hat{s}(n)$ şöyle ifade edilebilir:

$$\hat{s}(n) = u(n) + \sum_{k=1}^p a_k \hat{s}(n-k) \quad (7.13)$$

Hata işareti olan $e(n) = s(n) - \hat{s}(n)$ için tanımlı E enerjisini, çerçeve uzunluğu L boyunca minimize etmek gerekmektedir. E aşağıdaki ifade ile verilebilir:

$$\begin{aligned} E &= \sum_{n=0}^{L-1} (s(n) - \hat{s}(n))^2 \\ &= \sum_{n=0}^{L-1} \left(s(n) - u(n) - \sum_{k=1}^p a_k \hat{s}(n-k) \right)^2 \end{aligned} \quad (7.14)$$

(7.14) denklemindeki sorun, E enerjisine ait ifadede yalnızca filtre katsayıları değil, aynı zamanda zaten filtre katsayılarına bağlı olan yeniden elde edilen konuşma işareti $\hat{s}(n)$ 'in de bulunmasıdır. Bu nedenle, basitçe kısmi türev $\delta E / \delta a_i$ bulunup sıfıra eşitlenerek optimal katsayılar kümesi p adet lineer denklem bulunamaz.

En küçük kareler yaklaşımında aşağıdaki yaklaşıklık kabul edilmektedir:

$$\hat{s}(n-k) \approx s(n-k) \quad (7.15)$$

Bu yaklaşıklık (7.14)'de yerine konulursa:

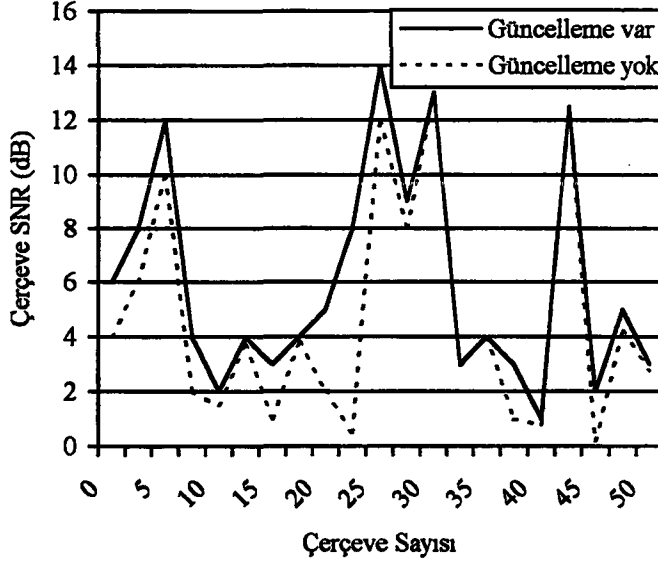
$$E \approx \sum_{n=0}^{L-1} \left(s(n) - u(n) - \sum_{k=1}^p a_k s(n-k) \right)^2 \quad (7.16)$$

bulunur.

Böylelikle $\delta E / \delta a_i$ kısmi türevleri bulunup sıfıra eşitlenerek, güncellenmiş filtre katsayılarını belirlemek için $i=1,2,...,p$ olmak üzere p adet lineer denklem elde edilebilir. Denemeler sonucunda güncellenen filtre katsayılarının, yeniden elde edilen konuşma işaretinin SNR'ı dikkate alındığında, orijinal katsayılara göre daha kötü sonuçlar verdiği ortaya çıkmıştır. Bunun nedeni ise (7.15) denklemindeki yaklaşıklık olarak görülmüştür. Araştırmalar sonucunda yeniden elde edilen konuşma işaretine ait bölütlenmiş SNR değerlerinde iyileşme elde edilmesinde, her çerçevede orijinal ve güncellenmiş filtre katsayıları için sentez filtresinin çıktısının bulunması ve bu değerlerden en iyi SNR'ı veren katsayıların o çerçeve için iletilmesi gerektiği sonucuna varılmıştır. Bu teknik kullanılarak güncellenen filtre katsayıları orijinal filtre katsayılarına göre çerçevelerin yalnızca %15'inde daha iyi olmuştur ve kodlayıcıya ait bölütlenmiş SNR, 0.25 dB iyileşmiştir.

Yukarıda bulunan sonuçlar umut kırıcı olmasına rağmen, sentez filtresinin parametrelerini güncellemek için daha gelişmiş bir yöntem bulunmasına çıkış noktası sağlamıştır. Nicemlenmiş LSF'lerin ilk değerleri ile başlanarak; ardışıl olarak p adet LSF değeri hata enerjisini azaltmak amacıyla nicemleme düzeyi olarak bir yukarı ve bir aşağı hareket ettirilmektedir. Bu uygulama p adet LSF için iteratif olarak tekrarlanmakta ve hata enerjisini düşüren her değişiklik kabul edilmektedir. Denenen bu yöntem ile üç iterasyon sonucunda bölütlenmiş SNR değerinin yaklaşık 1 dB iyileştiği görülmüş ve bu uygulamadan dolayı karmaşıklık yalnızca %80 oranında artmıştır.

Sentez filtresini güncellemek bölütlenmiş SNR değerlerinin iyileştirmenin yanı sıra, çerçeve çerçeve ölçülen SNR değerlerindeki minimumların çok düşük değerlere gitmesini de engellemiştir. Bunun etkisi Şekil 7.5'de gösterilmektedir.



Şekil 7.9 Güncellemenin SNR'ye Etkisi

Yukarıda elde edilen sonuçlar sentez filtresi parametrelerini güncellemenin bölütlenmiş SNR değerlerini yaklaşık 1 dB civarında iyileştirdiği ve göreceli olarak çerçeve çerçeve ölçülen SNR değerlerinin minimumlarının çok düşük değerlere gitmesini engellediğini ifade etmektedir. Benzer iyileşmeler LSF aradeğerleme gibi yaklaşımlarda da elde edilmesine rağmen, oluşan gecikme yukarıda açıklanan yöntemle göre daha fazladır.

7.4.6.2 Uyarma Parametrelerinin Hesabı

Kodlayıcının karmaşıklığını azaltmak için, orijinal konuşma işareti $s(n)$ ve yeniden elde edilmiş konuşma işareti $\hat{s}(n)$ ayrı ayrı ağırlıklandırılır, sonra fark bulunur.

$A(z) = 1 - \alpha_1 z^{-1} - \alpha_2 z^{-2} \dots \dots \alpha_p z^{-p}$ olan $H(z) = 1/A(z)$ şeklindeki tam kutuplu bir sentez filtresi ve γ 'nın sabit (benzetimlerde 0.9 değeri kullanılmıştır) olduğu $A(z)/A(z/\gamma)$ şeklinde bir hata ağırlıklandırma filtresi için, sentez filtresi ile ağırlıklandırma filtresinin arka arkaya bağlantısı $1/A(z/\gamma)$ şeklinde

ağırlıklandırılmış bir sentez filtresi kullanma ile eşdeğerdir. Buradan ağırlıklandırılmış hata $e_w(n)$ şu ifade ile verilebilir:

$$\begin{aligned} e_w(n) &= s_w(n) - \hat{s}_w(n) \\ &= s_w(n) - \hat{s}_0(n) - G_1[u(n-\alpha) * h(n)] - G_1[c_k(n) * h(n)] \end{aligned} \quad (7.17)$$

Burada $s_w(n)$ ağırlıklandırılmış giriş konuşma işareti, $\hat{s}_0(n)$ ağırlıklandırılmış sentez filtresinin önceki alt çerçevelerdeki girişlerden dolayı oluşan sıfır giriş cevabı ve $h(n)$ ise ağırlıklandırılmış sentez filtresinin dürtü cevabıdır.

Kod kitabı arama yordamı, alt çerçeve boyu N üzerinde bulunan ortalama karesel hata E_w 'yi minimize eden, adaptif kod kitabı için G_1 kazancını ve α gecikmesini ve sabit kod kitabı için k endeksini ve G_2 kazancını bulmayı amaçlamaktadır. E_w 'nin ifadesi (7.18) denklemi ile verilmiştir.

$$E_w = \frac{1}{N} \left(\sum x^2(n) - T_{ak} \right) \quad (7.18)$$

Burada:

$$T_{ak} = 2(G_1 C_\alpha + G_2 C_k - G_1 G_2 Y_{ak}) - G_1^2 \xi_\alpha - G_2^2 \xi_k \quad (7.19)$$

kod kitabı araması sonucunda maksimize edilmesi gereken terimdir. $x(n) = s_w(n) - \hat{s}_0(n)$ ise kod kitabı aramasının hedef işaretidir. Bu ifadede:

$$\xi_\alpha = \sum_{n=0}^{N-1} [u(n-\alpha) * h(n)]^2 \quad (7.20)$$

filtrelenmiş adaptif kod kitabı işaretinin enerjisidir ve

$$C_\alpha = \sum_{n=0}^{N-1} x(n) [u(n-\alpha) * h(n)] \quad (7.21)$$

ise filtrelenmiş adaptif kod kitabı işareti ile, kod kitabı hedefi $x(n)$ işareti arasındaki ilişkidir. Benzer şekilde, ξ_k filtrelenmiş sabit kod kitabı işareti $[c_k(n) * h(n)]$ 'in enerjisi ve c_k 'da bununla hedef işaret arasındaki ilişkidir. Son olarak da:

$$Y_{\alpha k} = \sum_{n=0}^{N-1} [u(n-\alpha) * h(n)] [c_k(n) * h(n)] \quad (7.22)$$

kod kitaplarına ait filtrelenmiş işaretlerin ilişkisidir.

Kod kitabı parametrelerini bulmak için genellikle kullanılan yaklaşım, başlangıçta (7.19) denkleminde $G_2=0$ alarak çözüme gitmektir. Böylelikle herhangi bir α değeri için en uygun G_1 kazancı $T_{\alpha k}$ 'nın G_1 'e göre kısmi türevi sıfıra eşitlenerek bulunabilir. Buna göre her α değeri için $T_{\alpha k}$ bulunabilir ve $T_{\alpha k}$ 'yı maksimize eden adaptif kod kitabı gecikmesi seçilebilir. Böylelikle adaptif kod kitabı parametreleri belirlenmiş olur ve benzer yöntemler sabit kod kitabı parametreleri k ve G_2 'nin bulunması için de uygulanabilir.

Burada kod kitabı araması için üç temel yöntem önerilmiş ve bu yöntemler ardışıl yaklaşım ve her iki kod kitabının da birlikte arandığı etraflı yaklaşım ile karşılaştırılmıştır. $T_{\alpha k}$ 'nın G_1 ve G_2 'ye göre kısmi türevleri sıfıra eşitlenerek, her kod kitabı endeksi α ve k çifti için en uygun kazanç değerlerini verecek şekilde çözülebilecek, iki adet eşitlik elde edilmektedir. Bunlar:

$$G_1 = \frac{C_{\alpha} \xi_k - C_k Y_{\alpha k}}{\xi_{\alpha} \xi_k - Y_{\alpha k}^2} \quad (7.23)$$

ve

$$G_2 = \frac{C_k \xi_{\alpha} - C_{\alpha} Y_{\alpha k}}{\xi_{\alpha} \xi_k - Y_{\alpha k}^2} \quad (7.24)$$

şeklindedir. Her iki kod kitabının da birlikte arandığı etraflı yaklaşımda $\xi_\alpha, \xi_k, C_\alpha, C_k$ ve $Y_{\alpha k}$ terimleri her kod kitabı endeks çifti α ve k için hesaplanmakta ve bu değerler G_1 ve G_2 'nin bulunması için kullanılmaktadır. Bulunan kazanç değerleri daha sonra (7.19) denkleminde yerine konularak, kodlayıcının α ve k değerlerini uygun seçerek maksimize etmesi gereken, $T_{\alpha k}$ bulunur. Etraflı yaklaşımdaki karmaşıklığın büyük kısmı her kod kitabı endeks çifti için çapraz ilişki terimi $Y_{\alpha k}$ 'nin bulunmasından dolayı ortaya çıkmaktadır.

Gerçeklenen CELP kodlayıcının önerilen yöntemlere ek olarak gerek ardışıl, gerekse etraflı arama için bölütlenmiş SNR (segSNR) cinsinden performansı ve karmaşıklığı Tablo 7.1'de verilmektedir. Önerilen alternatif arama yöntemleri aşağıda açıklanmıştır:

- Yöntem A: Burada α ve k alışılmış arama yordamı ile bulunmuş ve daha sonra (7.23) ve (7.24) denklemleri kod kitabı kazançlarının birlikte optimizasyonu için kullanılmıştır.
- Yöntem B: $G_2=0$ varsayılarak adaptif kod kitabı gecikmesi α bulunmuş ve sadece bu değer G_1 , G_2 ve k 'nin birlikte belirlendiği sabit kod kitabı araması sırasında kullanılmıştır.
- Yöntem C: G_1 , G_2 , α ve k yöntem B'deki gibi bulunmuş ve daha sonra bir kere k bulunduktan sonra α mümkün olan her değer için G_1 , G_2 ve $T_{\alpha k}$ 'yi bulmak suretiyle güncellenmiştir.

Tablo 7.4 Çeşitli Kod Kitabı Arama Yöntemlerinin Performansı ve Karmaşıklık

	<i>SegSNR(dB)</i>	<i>Karmaşıklık</i>
Ardışıl Arama	9.7	1
Yöntem A	10.1	1.02
Yöntem B	10.3	1.3
Yöntem C	10.6	1.4
Etraflı Arama	10.8	60

Tablo 7.1'den görülebildiği üzere etraflı kod kitabı arama, ardışıl aramaya göre yaklaşık 1 dB'lik bir iyileştirme getirmektedir. Bununla birlikte kodlayıcının karmaşıklığını ise altmış kat arttırmaktadır. Yöntem C, kodlayıcı karmaşıklığını yine ardışıl aramaya göre %40 oranında arttırırken, neredeyse etraflı arama kadar bir performans sağlamaktadır. Ayrıca yöntem A bile kodlayıcı karmaşıklığını yaklaşık %2 arttırırken önemli sayılabilecek performans artışı sağlamaktadır.

BÖLÜM 8

SONUÇLAR VE ÖNERİLER

8.1 GİRİŞ

Kanal veya disk kapasitesinden bahsedilirken gigabitlerin alışlagelmiş büyüklüklerden olageldiği bir çağda, düşük bit hızlarında ses kodlama veya sıkıştırmanın amacı herkes tarafından pek anlaşılamamaktadır. Bundan dolayı yeni gelişmeler doğrultusunda ortaya çıkan tekniklerin kapasite gereksinimlerini ve şu anda sağlanabilen kapasiteleri karşılaştırmak gereklidir. Bu tezin temel amacı, düşük bit hızlarında ses işaretlerinin kodlanmasının, günümüzde de halen çok önemli bir alan olduğunun altını çizmek ve teknolojinin bize sağlamış olduğu gelişmiş işaret işleme altyapılarını kullanarak, kaliteli kodlayıcılar tasarlanmasının mümkün olduğunu kanıtlamaktır.

8.2 KONUŞMA KODLAMA

Konuşma kodlama algoritmaları analog bir işaretin sayısal gösterimindeki bit hızını, işaret kalitesini mümkün olduğu kadar kabul edilebilir bir düzeyde tutarak, düşürmeye çalışmaktadır. Konuşma kodlamada istenilen başarımın elde edilebilmesi için ilgilenilen işaretin özelliklerinin iyi bilinmesi çok önemlidir. Ancak bu şekilde, konuşma işaretinin içerisindeki artık bilgi atılıp daha etkin kodlama yapılarak düşük bit hızlarına inilebilir. Konuşma kodlama sistemlerinin temelinde hemen hemen bütün algoritmalarından bağımsız olarak işaretin analizi, önceden tahmin edilen fazlalık bilginin atılması ve esas bilgiyi içeren kısmın verimli bir şekilde kodlanması bulunmaktadır.

8.2.1 Konuşma Sıkıştırma için Süregelen Gereksinim

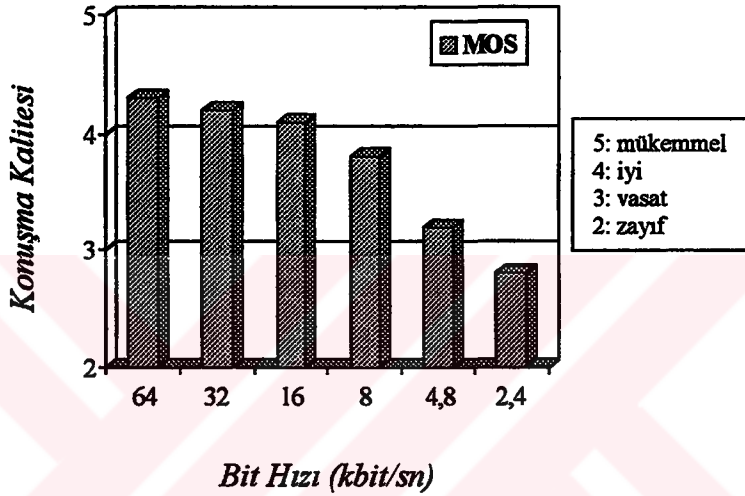
Konuşma sıkıştırma tekniği, güvenli uzak mesafe iletişimi, yüksek kaliteli konuşma kaydı (saklama) ve mesaj şifreleme alanlarında anahtar teknoloji konumundadır. Gelecekte de, optik iletim ortamları teorik olarak sınırsız bandgenişliği sağlasa da, konuşma sıkıştırma anahtar teknoloji olmaya devam edecektir. Bunun nedeni, özellikle uydu ve mobil haberleşme gibi band sınırlı uygulamalara ve CD-ROM ve yarıiletken bellekler gibi bit hızı sınırlı saklama ortamlarına giderek daha fazla duyulan gereksinimdir. Konuşma bilgisinin arşiv amaçlı büyük miktarlarda kaydı için, optik disk ve katıhal (*solid-state*) bellek kapasiteleri önemli miktarlarda artsa bile, konuşma sıkıştırma önemini koruyacaktır.

Sayısal mobil haberleşmeye gittikçe artan oranda gereksinim duyulması, düşük bit hızlı konuşma teknolojisinin önemini arttırmaktadır. Düşük bit hızlarında düşük maliyetler ile konuşma kodlamada son yıllarda sağlanan gelişme, oldukça etkileyicidir. Onbeş yıl kadar önce 24 kbit/san'nin altında kaliteli kodlanmış konuşma sağlanamazken, günümüzde 4.8 kbit/san'de kodlanmış yüksek kaliteli konuşma sağlamış, hatta bu hızlar standartlaştırılmıştır.

Şu anda önemli çalışmaların yapıldığı 2 kbit/san civarı ses kaydı için oldukça çekici bir hızdır. Bu hız kullanılarak yaklaşık iki saat uzunluğunda bir ses kaydı, tek bir 16 Mbit'lik hafıza tümdevresine kaydedilebilecektir. Böylelikle kişisel haberleşme terminallerindeki ses postası servislerinin imkanları ve telefon cevaplama makinelerindeki kayıt kapasiteleri önemli ölçüde artırılabilir. MELP gibi üzerinde hemen hemen yeni çalışılmaya başlanan teknikler, ayrıca konuşma üretimi ve algılanması konusundaki anlayışın temelindeki gelişmeler de kullanılarak, sözkonusu düşük hızlara inilmesi işten bile değildir. Böylelikle ses iletiminde, elektronik yayıncılıkta, uzaktan eğitim/mühendislikte, çoklu-ortam (*multimedia*) uygulamalarında ve benzer. alanlarda yeni imkanların ortaya çıkması sağlanacaktır.

8.2.2 Konuşma Kodlamanın Günümüzdeki Yeterliliği

Günümüzde darbandlı telefon konuşmasında (300-3400 Hz), 2.4-64 kbit/san aralığındaki çeşitli bit hızları için erişilebilen konuşma kalitesi beş noktalı MOS ölçeği cinsinden Şekil 8.1’de verilmektedir. Belirtilen bit hızlarında kodlanan konuşmaya ait anlaşılabilirlik, sözkonusu bütün yapılarda yeterlidir, ve bu bağlamda önemli bir husus da değildir [52].



Şekil 8.1 - Çeşitli bit hızları için konuşma kalitesi MOS değerleri

Yukarıdaki şekilde görüldüğü gibi PCM (64 kbit/san), ADPCM (32 kbit/san) gibi dalgaşekli kodlama teknikleri ile oldukça yüksek düzeyde konuşma kalitesi elde edilebilmektedir. 16 kbit/san’nin altındaki hızlar için yeterli konuşma kalitesi karmaşık doğrusal öngörülü kodlama teknikleri ile sağlanabilmektedir. Bu yaklaşıma ait önemli örnekler çoklu darbe ile uyarılan LPC (MPLPC), düzenli darbe ile uyarılan LPC (RPLPC) ve kod ile uyarılan LPC (CELP) teknikleri olarak verilebilir. CELP algoritması dalgaşekli kodlamasına ait yüksek kalite potansiyelini, model tabanlı ses kodlayıcıların sıkıştırma yeteneği ile birleştirmektedir. Günümüzde CELP tekniği 16 kbit/san ve altındaki konuşma kodlaması için seçilmiş bir teknolojidir. CELP kodlamasının düşük gecikmeli yapıları, 16 kbit/san civarında ses kalitesi ve performans olarak PCM’e oldukça yaklaşmıştır.

Günümüzdeki araştırma konuları 16 kbit/san'den çok daha düşük bit hızlarında yoğunlaşmaktadır. Bunun yanı sıra bu çalışmada açıklandığı gibi, eşdeğer hızlardaki kodlama algoritmalarının daha kaliteli konuşma sağlayacak ve altyapıyı daha verimli kullanacak şekilde alternatif tasarımları yapılmaktadır. Karmaşık olmalarına rağmen düşük bit hızlı kodlama algoritmaları ile tümdevre ve sayısal işaret işleme teknolojilerindeki gelişmeler çerçevesinde giderek daha mükemmel çözümlere ulaşılmaktadır.

8.2.3 Teknolojik Hedefler

Konuşma entropisini tam tamına ifade eden bir formül bulunmadığı için, konuşma kodlamanın doğal hedefi, şu anda kabul edilen yüksek kaliteyi sağlayan bit hızlarının altına inmektir [53]. Yani hedef, telefon konuşması için 4 kbit/san'nin, genişbandlı konuşma için 8 kbit/san'nin ve CD kalitesinde konuşma için 24 kbit/san'nin altıdır. Verilen bu örnekler için bit hızı örnek başına yaklaşık 0.5 bit civarındadır.

Diğer yandan gerçek hayatta karşılaşılan gürültü, iletim hataları ve paket kayıpları gibi sorunlara karşı sağlam algoritmaların gerçekleşmesi de bir başka önemli husustur. Bunun yanı sıra, sözkonusu geliştirmeleri kabul edilebilir bir gerçekleştirme karmaşıklığı içerisinde sağlamak da, üstesinden gelinmesi gereken bir sorundur.

Bütün bu uğraşlar, bir çok ayrı disiplin çerçevesinde verilmektedir ve gerek bu disiplinler, gerekse bunların etkileşimi konusunda şu andaki bilgi düzeyi ise sınırlıdır. Kodlama, haberleşme ve ağlar, konuşma üretimi ve algılanması ve sayısal işaret işleme disiplinlerinde sağlanacak gelişmeler bu kapsamda oldukça önem kazanmaktadır.

8.2.4 Gelecekteki Yönelim

Aşağıda konuşma kodlamanın iletişim, bilgisayar ağları, konuşma üretimi ve algılanması, sayısal işaret işleme alanlarında nerelere yöneleceği incelenmiştir.

8.2.4.1 Kodlama, İletişim ve Bilgisayar Ağları:

Yakın tarihlerde yapılan çalışmalar, kaynak kodlamanın temel yapısal blokları konusunda önemli gelişmeler sağlanmıştır. Bu gelişmeler:

- zaman-frekans analizi için esnek yöntemler
- adaptif vektör nicemleme
- gürültüsüz kodlama

olarak sıralanabilir. Kanal kodlamasında, mobil kanallar için kodlu modülasyon ve bilgisayar ağları için gömük (*embedded*) iletim protokolleri gibi tamamlayıcı gelişmeler de bulunmaktadır. Ancak bu tekniklerin konuşma kodlamasına uygulanması yönünde henüz zorlayıcı gereksinimler tam olarak oluşmamıştır. Fakat yakın gelecekte, mobil haberleşme alanında özellikle çoklu-ortamlı (*multimedia*) uygulamalar sözkonusu olduğunda, kaynak ve kanal kodlamanın yanısıra ağ yapılarının da gözetildiği tümleşik tasarımlar özel bir önem kazanacaktır.

8.2.4.2 Konuşma Üretimi ve Algılama:

Konuşma işaretlerindeki periyodikliğe ait basit modeller ve ağızdaki ses oluşumunu sağlayan boşluğun basit kaynak modelleri, konuşma üretim sürecini doğrudan ve derlitoplu ifade eden yaklaşımlar ile ya değiştirilmeli ya da desteklenmelidir. Benzer şekilde bozulma önleme modelleri de, bozulmayı spektral veya zaman bölgesinde engelleme kapasitesine sahip modeller ile değiştirilmelidir. Bu modeller, duymayı daha iyi modelleyen temellere oturmalı ve aynı zamanda tonlar ve gürültü gibi basit uyarma modelleri yerine, gerçek konuşma işaretleri ile yapılan deneysel uygulama sonuçlarına dayanmalıdır.

8.2.4.3 Sayısal İşaret İşleme:

Şu andaki teknolojinin erişilen düzeyi sözkonusu olduğunda, tek tümdevreden oluşan genel amaçlı bir sayısal işaret işlemci saniyede yaklaşık yüz milyon aritmetik işlemi yapabilmekte, ve bir santimetrekarelik bir bellek yongası üzerine yaklaşık yirmibeş megabit bilgi yazılabilmektedir. Önümüzdeki birkaç yıl içerisinde bir tümdevreden sağlanabilen işlem ve saklama kapasitesinin önemli şekilde artacağı beklenmektedir. İşlemci etkinliği, yani saniyede miliwatt güç başına milyon işlem olarak sağlanan kapasitenin (MIPS-*Million Instruction Per Second*) yakın bir gelecekte şu andakine göre bir kat artması mümkün olacaktır. Buna rağmen, bu işlemciler ile daha karmaşık konuşma işleme algoritmalarının gerçekleştirilmesini, özellikle konuşma kalitesinin işlemci maliyeti ve güç harcaması ile formüle edilebildiği etkin yapılar için işlemci mimarilerini karmaşık algoritmalara eşlemeyi kolaylaştıracak doğrultuda sürekli gelişmelerin oluşmasını sağlamak gerekmektedir.

8.3 SONSÖZ

Bu tezde düşük bit hızlarında konuşma kodlama teknikleri ve sayısal işaret işlemciler (DSP) ile gerçekleştirme yöntemleri araştırılmış; bu tekniklerden hareketle, 4.8 kbit/san civarındaki bit hızları için DSP tabanlı gerçeklemeler yapılmıştır. Çalışmalarda daha düşük hızlara inmek değil eşdeğer hızlarda kaynakları (DSP imkanlarını) daha verimli kullanıp, daha kaliteli kodlayıcılar tasarlama amacı güdülmüştür. Ayrıca iletişim uygulamalarında yaygın olarak kullanılan bir DSP olan TMS320C50 esaslı yazılım kullanılarak uygulanan bazı etkin programlama teknikleri ele alınmıştır. Bununla birlikte, bazı uygulamalarda, etkin programlama teknikleri uygun miktarda hız ve bellek indirimini yaratmak için yeterli olmadığından, algoritmik sadeleştirmelerle tamamlanması gerekmiştir. Bu nedenle, bir CELP kodlayıcısı gerçeklerken gereken hız ve bellek kapasitesini azaltmak için çeşitli algoritmik sadeleştirmeler önerilmiştir. Ayrıca kodlayıcının sentezleyerek analiz çevrimi incelenmiş ve uyarma parametreleri belirlendikten sonra kısa dönemli sentez filtresi parametrelerinin güncellenmesi için yöntemler araştırılmıştır. Gösterilmiştir ki sentez

filtresi parametrelerini bandgeniřlięi geniřletme (*bandwidth expansion*) ve aradeęerleme (*interpolation*) yaklařımlarında olduęu gibi gncellemek performansta nemli geliřmeler saęlamaktadır. Bundan bařka btn nerilen geliřtirmelerin kullanıldıęı DSP tabanlı bir CELP kodlayıcısı, yksek seviyeli yazılım dili C kullanılarak gereklenmiř ve sabit noktalı bir sayısal iřaret iřlemciye uygulanmıřtır. Ayrıca CELP kodlayıcıların yeterlilięini arttırmak iin AbS evriminde geliřtirmeler yapılmıřtır. Tm bu abalar sonucunda olduka karmařık bir yazılım gerektiren CELP kodlayıcısı TMS320C50 gibi sabit noktalı bir sayısal iřaret iřlemci zerinde yksek seviyeli programlama dili C ile gereklenmiř ve uygulama DSK geliřtirme ortamında denenmiřtir.

Yukarıda elde edilen sonular doęrudan uygulanabilir sonulardır. Eldeki program sisteminin bir zel santralin (*PBX-Private Branch Exchange*) sz postası iřlevini yerine getiren modlnde kullanılması iin alıřmalar bařlatılmıřtır. Konuřma sıkıřtırma, sz postası gibi sesin yarıiletken belleklere kaydedildięi uygulamalarda modl maliyetini nemli oranlarda ařaęılara ekebilmektedir. Ayrıca bu alıřmanın 4.8 kbit/san'nin altındaki hızlarda benzerlerinin gereklenmesi de planlanmaktadır.

KAYNAKLAR

- [1] DAS J., MULLICK S. K., CHATTERJEE P. K., Principles of Digital Communication, John Wiley&Sons, Inc., New York, ISBN:0-470-20240-8, (1986)
- [2] JAYANT N. S., NOLL, P., Digital Coding of Waveforms: Principles and Applications to Speech and Video, Prentice-Hall, Englewood Cliffs, New Jersey ISBN: 0-1321-1913-7, (1984)
- [3] KONDOZ A. M., Digital Speech, John Wiley&Sons, Inc., New York, ISBN:0-471-95064-5, (1995)
- [4] INCE N., Digital Speech Processing: Speech Coding, Synthesis and Recognition, Kluwer Academic Publishers ISBN: 0-7923-9220-5, (1992)
- [5] FELLBAUM K., Elektronische Sprachverarbeitung, Franzis Verlag GmbH-München, ISBN:3-7723-6532-9, (1992)
- [6] FELLBAUM K., Sprachverarbeitung und Sprachübertragung, Springer Verlag, (1984)
- [7] CCITT, Yellow Book, Recommendation G.711,G.712
- [8] GRAY M. R., Source Coding Theory, Kluwer Academic Publishers, ISBN: 0-7923-9048-2 (1990)
- [9] ANONİM, Speech Processing in the Pan-European Digital Mobile Radio System (GSM)-System Overview, Proceedings of GLOBECOM, (1989)
- [10] FEUERSTEIN M.J., RAPPAPORT T.S., Wireless Personal Communications, Kluwer Academic Publishers, ISBN: 0-7923-9280-9, (1992)
- [11] PHILIPS R., Review of INMARSAT System, IEE Colloquium on Satellite Communication for Mobile Users, (1985)
- [12] ANONİM, INMARSAT-M Voice Coding System Description, Digital Voice Systems Inc., (1991)

- [13] CULOT B., CACHEIRO R., COUET J., LIYOTKY H., Rural Communication: Problems and Solutions, Electrical Communications pp.192-197, (1989)
- [14] TREMAIN T., The Government Standard Linear Predictive Coding Algorithm (LPC-10), Speech Technology, (1982)
- [15] CAMPBELL J., The Proposed Federal Standard 1016 4800 b/s Voice Coder (CELP), Speech Technology pp.58-64, (1990)
- [16] KEMP D., SUEDA R. A., TREMAIN T. E., An Evaluation of 4800 b/s Coders, Proceedings of ICASSP pp.200-203, (1989)
- [17] ANONIM, Speech & Audio Coders: Analysis and Evaluation, Forward Concepts Inc., (1997)
- [18] ANDERSON J.B., MOHAN S., Source and Channel Coding: An Algorithmic Approach, Kluwer Academic Publishers, ISBN: 0-7923-9210-8, (1991)
- [19] GERSHO A., GRAY R.M., Vector Quantization and Signal Compression Kluwer Academic Publishers, ISBN: 0-7923-9181-0, (1992)
- [20] LINDE Y., BUZO A., GRAY R., An Algorithm for Vector Quantiser Design, IEEE Trans. On Communications pp.84-95, (1980)
- [21] ATAL B., CUPERMAN V., GERSHO A., Advances in Speech Coding, Kluwer Academic Publishers, ISBN: 0-7923-9091-1, (1991)
- [22] PAPAMICHALIS P.E., Practical Approaches to Speech Coding, Prentice Hall, Englewood Cliffs, NJ., ISBN: 0-1368-9019-9, (1987)
- [23] SPANIAS A., Speech Coding: A Tutorial Review, Proc. IEEE pp.1541-1582, (1994)
- [24] GRAY R., Vector Quantization, IEEE ASSP Magazine pp.4-28, (1984)
- [25] BRACEWELL R.N., The Fourier Transform and its Applications, McGraw-Hill ISBN: 0-0700-7015-6, (1986)
- [26] GRIFFIN D.W., LIM J.S., Signal Estimation from Modified Short-Time Fourier Transform, IEEE Trans. On ASSP pp.236-243, (1988)
- [27] PORTNOFF M.R., Time-Frequency Representation of Digital Signals and Systems Based on Short-Time Fourier Analysis, IEEE Trans. On ASSP pp.55-69, (1980)

- [28] SAITO S., Fundamentals of Speech Signal Processing, Academic Press ISBN: 0-1261-4880-5, (1986)
- [29] KAY S., More Accurate Autoregressive Parameter and Spectral Estimates for Short Data Records, IEEE Trans. on ASSP pp.56-65, (1983)
- [30] McAULAY R.J., Maximum Likelihood Spectral Estimation and its Application to Narrow-Band Speech Coding, IEEE Trans. on ASSP pp.243-251, (1984)
- [31] McAULAY R.J., QUATIERI T.F., Speech Analysis/Synthesis Based on Sinusoidal Representation, IEEE Trans. on ASSP pp.744-754, (1986)
- [32] FETTE B., High Quality Secure Voice Communication, Speech Technology pp. 40-48, (1989)
- [33] BARNWELL P.T., NAYEBI K., RICHARDSON C.H., Speech Coding, John Wiley&Sons ISBN: 0-4715-1692-9, (1996)
- [34] GOUVIANAKIS N., XYDEAS, Advances in Analysis-by-Synthesis LPC Speech Coders, Jour. of IERE pp.272-286, (1987)
- [35] KROON P., DEPRETTERE E., A Class of Analysis-by-Synthesis Predictive Coders for High Quality Speech Coding at Rates Between 4.8 and 16 kbit/s, IEEE Jour. of Select. Areas in Comm. pp.353-363, (1988)
- [36] ATAL B., REMDE J., A New Model of LPC Excitation for Producing Natural Sounding Speech at Low Bit Rates, Proc. Of ICASSP pp.614-617, (1982)
- [37] CHUNG J., SCHAFER R., A 4.8 kbit/s Homomorphic Vocoder Using Analysis-by-Synthesis Excitation Analysis, Proc. Of ICASSP pp.144-147, (1989)
- [38] SCHROEDER M., ATAL B., Code Excited Linear Prediction: High Quality Speech at Low Bit Rates, Proc. Of ICASSP pp.937-940, (1985)
- [39] ROSE R., BARNWELL T., The Self-Excited Vocoder Alternative Approach to Toll Quality at 4800 bit/s, Proc. Of ICASSP pp.453-456, (1986)
- [40] SINGHAL S., ATAL B., Amplitude Optimization and Pitch Prediction in Multipulse Coders, IEEE Trans. On ASSP pp.317-327, (1989)
- [41] KONDOZ A.M., EVANS B.G., CELP Base-Band Coder for High Quality Speech Coding at 9.6 and 2.4 kbit/s, Proc. Of ICASSP pp.159-163, (1988)

- [42] ATAL B., Predictive Coding of Speech at Low Bit Rates, IEEE Trans. on Comm. pp.600-614, (1982)
- [43] CHEN J., High Quality 16 kbit/s speech coding with a one Way delay Less than 2 ms, Proc. of ICASSP pp.453-456, (1990)
- [44] RAMACHANDRAN R.P., KABAL P., Pitch Prediction Filters in Speech Coding, IEEE Trans. on ASSP pp.467-478, (1989)
- [45] KLEIJN W., Improved Speech Quality and Efficient Vector Quantization in SELP, Proc. of ICASSP pp.155-158, (1988)
- [46] KROON P., ATAL B., Pitch Predictors with High Temporal Resolution, Proc. of ICASSP pp.661-664, (1990)
- [47] KABAL P., RAMACHANDRAN R.P., Joint Solutions for Formant and Pitch Predictors in Speech Processing, Proc. of ICASSP pp.315-318, (1988)
- [48] ANONİM, TMS320C40 Digital Signal Processor User's Guide, Texas Instruments,(1992)
- [49] ANONİM, TMS320C5X Digital Signal Processor User's Guide, Texas Instruments, (1993)
- [50] ANONİM, TMS320C5X DSP Starter Kit User's Guide, Texas Instruments, (1994)
- [51] AŞKIN T., DURUSOY G., Konuşma Kodlama Sistemleri ve Başarım Ölçütlerine Yeni bir Yaklaşım, Sinyal İşleme Uygulamaları Kurultayı, (1997)
- [52] KITAWAKI N., Objective Quality Evaluation for Low-Bit-Rate Speech Coding Systems, IEEE Jour. on Select. Areas in Comm. Pp.242-248, (1988)
- [53] JAYANT N.S., Signal Compression: Technology Targets and Research Directions, IEEE Jour. on Select. Areas in Comm. pp.796-818, (1992)

EKLER

Sözlük

A

aliasing örtüşme
all pole *tam kutuplu*
Analysis-by-Synthesis (AbS) *Sentezleyerek Analiz*
aperture açıklık
Arithmetic Logic Unit (ALU) *Aritmetik Lojik Birim*
autocorrelation özilişki
autoregressive özbağlanımlı
assembler çevirici

B

backward *geriye doğru*
bias yanlılık
boot önyükleme
bursty yığımsal
buzz vızlama

C

cascade *peşpeşe*
centre clipping *merkez kırpma, katlama*
centroid *ortaç*
cepstral *kepstral*
chip *kırmık*
Code Excited Lineer Prediction (CELP) *Kod ile Uyarılan Doğrusal Öngörülü Kodlama*
comb filter *tarak filtre*
communication *iletişim*
companding *sıkıştırma-açma*
complex *karmaşık*
composite *bileşik*
compressor *sıkıştırıcı*
correlation *ilişki*
covariance *ortak değişken*

D

data bus veri yolu
debug hata ayıklama
decimation örnek seyreltme
Digital Signal Processor (DSP) Sayısal İşaret İşlemci
dispersion ayırma, saçılma
distortion bozulma
domain bölge
double precision çifte doğruluk

E

embedded gömük
emulation emülasyon
estimation öngörü
enhancement pekiştirme
exponent üst

F

fading sönme, bayılma
fidelity aslına sadakat
filter süzgeç
formant rezonans
Forward Error Correction (FEC) ileriye doğru hata düzeltme
frame pencere
fraction kesir

G

glottal ses telleri arası
grid ızgara

I

imaginary sanal
impuls dürtü
in-circuit emulation (ICE) devre-üzerinde emülasyon
incremental artırımlı
instruction cycle komut çevrimi
interpolation aradeğerleme
Integrated Services Digital Network (ISDN) Tümleşik Hizmetler Sayısal Şebekesi
iterative iteratif

J**jitter** *seğirme***L****lattice filter** *kafes süzgeç***lobe** *lob***Linear Predictive Coding (LPC)** *Doğrusal Öngörülü Kodlama (DÖK)***Line Spectrum Frequency (LSF)** *Çizgi Spektrum Frekansı***M****mantissa** *mantis***match** *uyumlama***maximum likelihood** *maksimum benzerlik***Motion Pictures Expert Group (MPEG)** *Hareketli Resimler Uzman Grubu***multimedia** *çokluortam***N****nasal** *burundan çıkan***O****operand** *işlem ögesi***order** *derece***overflow** *üstten taşma***P****peak** *tepe***Personal Communication Systems (PCS)** *Kişisel Haberleşme Sistemleri***pipeline** *ardışık düzen***pitch** *perde***Pitch Adaptive Mixed Excitation (PAME)** *Adaptif Perdeli Karışık Uyarma***predictor** *öngörücü***Probability Density Function (PDF)** *olasılık yoğunluk fonksiyonu***procedure** *yordam***Public Switched Telephony Network (PSTN)** *Kamu Telefon Şebekesi***Q****quantization** *nicemleme***quasi** *hemen hemen*

R

real *gerçek*
reciprocal *karşılıklı*
recursive *özyinelemeli*
residue *artık, kalıntı*

S

sampling *örnekleme*
saturation *doyma*
scalar *skalar*
shift *kaydırma*
short-time *kısa süreli*
solid state *katıhal*
spectral *spektral*
spectrum *spektrum*
spurious *yapma*
stationary *durağan*
subjective *öznel*
sub-multiple *astkat*

T

telecommunication *uziletişim, haberleşme*
text *metin*
tracking *izleme*
transmission *iletim*

U

underflow *alttan taşma*
uniform *düzgün*
unique *tek*
unvoiced *sessiz*

V

variance *varyans*
Very Small Aperture Terminal(VSAT) *çok küçük açıklıklı terminal*
vocal tract *ağızdaki ses oluşumunu sağlayan rezonans boşluğu*
voiced *sesli*

W

weighting *ağırlıklandırma*

ÖZGEÇMİŞ

Tarık Aşkın, 1964 yılında İstanbul'da doğdu. İstanbul Erkek Lisesi'nden 1982 yılında mezun olduktan sonra, aynı yıl İstanbul Teknik Üniversitesi, Elektrik-Elektronik Fakültesi, Elektronik ve Haberleşme Bölümüne girdi. Lisans öğrenimini Şubat-1987'de tamamladı ve Mart-1987'de aynı üniversitenin Haberleşme Anabilim Dalı'nda Araştırma Görevlisi olarak çalışmaya başladı. 1990 yılında İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme Programı'nda Yüksek Lisans eğitimini bitirerek, aynı programda doktora yapmaya başladı. 1993 yılında Siemens-Simko firması Araştırma-Geliştirme Bölümü'nde tasarım mühendisi olarak işe başlayan Tarık Aşkın, halen aynı bölümde Geliştirme Grup Yöneticisi olarak çalışmaktadır.

