

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**IP TABANLI SES İLETİMİ ORTAMINDA
AKUSTİK YANKI GİDERİMİ**

Mühendis Pınar ARSLAN

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

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Lütfiye DURAK

İSTANBUL, 2008

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	iv
KISALTMA LİSTESİ.....	v
ŞEKİL LİSTESİ.....	vi
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
IP TABANLI SES İLETİMİ ORTAMINDA AKUSTİK YANKI GİDERİMİ.....	x
ACOUSTIC ECHO CANCELLATION IN IP BASED VOICE TRANSMISSION	xi
1. GİRİŞ	1
1.1 Hibrit Yankı Giderimi.....	1
1.2 Akustik Yankı Giderimi.....	5
2. DÖNÜŞÜM BÖLGESİNDE İŞARET ANALİZİ.....	8
2.1 Dönüşüm Bölgeleri.....	8
2.1.1 Frekans Bölgesi Analizi	9
2.1.2 Kesirli Fourier Dönüşüm Bölgesi.....	10
3. ALTERNATİF UYARLANIR SÜZGEÇLEME YAPILARI	14
3.1 En Küçük Ortalama Kareler (LMS) Algoritması	14
3.2 Normalize LMS (NLMS) Algoritması	18
3.3 Özyineli En Küçük Kareler (RLS) Algoritması	22
3.3.1 Matris Evirme.....	25
3.3.2 Üstel Ağırlıklanan RLS Algoritması	25
3.3.3 Süzgeç Katsayı Vektörü için Zaman Güncellemesi	27
3.4 Kesirli Fourier Bölgesinde LMS Algoritması.....	28
4. IP TABANLI SES İLETİMİ ORTAMINDA AKUSTİK YANKI GİDERİMİ....	30
4.1 LMS Algoritması.....	31
4.2 NLMS Algoritması	34
4.3 RLS Algoritması.....	36
4.4 KFD Bölgesi LMS.....	38
5. Sonuç	42
KAYNAKLAR.....	44

EKLER	46
Ek 1 Uyarlamalı süzgeçleme algoritmaları kullanılarak sağlanan akustik yankı giderimi için Matlab programı	47
EK 2 Uyarlanır Süzgeçleme Akış Şeması.....	48
EK 3 LMS algoritması	49
EK 4 NLMS Algoritması	51
EK 5 RLS Algoritması.....	53
EK 6 KFD Bölgesi LMS Algoritması.....	55
ÖZGEÇMİŞ.....	58

SİMGE LİSTESİ

$\mathbf{u}(n)$	Süzgeç girişi, uzak-uç işareti
a_k	Zayıflama
t_k	Gecikme
$H(n)$	Akustik çevrenin dürtü yanıtı
$\mathbf{b}(n)$	Yakın-uç işareti
$d(n)$	İstenilen işaret, yankı işareti
$\mathbf{w}(n)$	Süzgeç katsayısı
$\delta_{\mathbf{w}}(n+1)$	Değişimin karesel normu
λ_i	Lagrange çarpanı
$J(n), \delta(n)$	Maliyet fonksiyonu
μ	Adım büyüklüğü
$e(n)$	Hata işareti
$\beta(n,i)$	Ağırlıklandırma faktörü
$\mathbf{R}(n)$	İlinti matrisi
$\mathbf{z}(n)$	Çapraz ilinti matrisi
$\mathbf{p}(n)$	Ters ilinti matrisi
$\mathbf{k}(n)$	Kazanç vektörü
$\xi(n)$	Önsel kestirim hatası
$\mathbf{m}(n)$	Mikrofon işareti

KISALTMA LİSTESİ

DSP	Digital Signal Processing, Sayısal İşaret İşleme
DTD	Double Talk Detector, Çift Konuşma Dedektörü
ERL	Echo Return Loss, Yankı Geri Dönüş Kaybı
ERLE	Echo Return Loss Enhancement, Yankı Geri Dönüş Kayıp Artışı
FIR	Finite Impulse Response, Sonlu Darbe Cevabı
FrFT	Fractional Fourier Transform
GAL	Gradient Adaptive Lattice, Eğim Uyarlanır Kafes
IIPNLMS	Improved IPNLMS, İyileştirilmiş IPNLMS
IIR	Infinite Impulse Response, Sonsuz Darbe Cevabı
IP	Internet Protocol, İnternet Protokolü
IPNLMS	Improved PNLMS, İyileştirilmiş PNLMS
ISDN	Integrated Services Digital Networks, Tümleşik Hizmetler Sayısal Şebekesi
JB	Jitter Buffer, Seçirme Belleği
KFD	Kesirli Fourier Dönüşümü
LMS	Least Mean Squares, En Küçük Ortalama Kareler
NLMS	Normalized Least Mean Squares, Normalize En Küçük Ortalama Kareler
NLP	Nonlinear Processor, Doğrusal Olmayan İşlemci
PNLMS	Proportionate NLMS, Orantılı NLMS
PSTN	Public Switched Telephone Network, Kamusal Anahtarlama Telefon Ağı
QoS	Quality of Service, Servis Kalitesi
RLS	Recursive Least Squares, Özyineli En Küçük Kareler
SGA	Stochastic Gradient Algorithm, Olasılıksal Eğim Algoritması
VC PAC	Voice Coder/Packetizer, Ses Kodlayıcı/Paketleyici
VDC DPAC	Voice Decoder/Depacketizer, Ses Kod Çözücü/Paket Ayırıcı
VLSI	Very Large-Scale Integrated, Çok Büyük Çapta Tümleşim
YGS	Yüksek Geçiren Süzgeç

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 Hibrit dönüştürücüde empedans uyumsuzluğundan kaynaklanan yankı ve hibrit yankı gidericinin yapısı	2
Şekil 1.2 Tipik bir hibrit yankı giderici ve yapısı.....	3
Şekil 1.3 PSTN ağlarda yankı gidericinin yapısı.....	3
Şekil 1.4 Paket telefon ağlarında yankı gidericinin yapısı.....	4
Şekil 1.5 Uzak uç işaretinin gecikmeye ve zayıflamaya uğramasıyla oluşan akustik yankı.....	5
Şekil 1.6 Uyarlanır süzgeçleme kullanılarak elde edilen kapalı ortamın akustik dürtü yanıtı modeli	6
Şekil 2.1 a) Sürekli zaman işareti b) Ayırık zaman işareti.....	8
Şekil 2.2 Bir konuşma işaretinin a) Zaman bölgesi b) Frekans bölgesi gösterimleri	9
Şekil 2.3 Zaman-frekans bölgesi üstünde <i>a.</i> dereceden KFD bölgesi	13
Şekil 3.1 Uyarlanır FIR süzgecin blok diyagramı	15
Şekil 3.2 LMS algoritmasının işaret-akış diyagramı	17
Şekil 3.3 RLS Süzgeç Yapısı.....	28
Şekil 3.4 KFD Bölgesi LMS algoritmasının yapısı.....	28
Şekil 4.1 IP tabanlı ağlarda giderimi için kullanılan teknik.....	30
Şekil 4.2 Uzak uç bölgesi akustik yankıyı üreten odanın dürtü yanıtı	31
Şekil 4.3 Yankı işaretinin LMS algoritması kullanılarak elde edilen kestirimi	32
Şekil 4.4 LMS anlık karesel hatasının 40.000 örnekte dağılımı.....	33
Şekil 4.5 LMS Algoritmasında 40.000 örnek için 11,67 dB ortalamalı ERLE dağılımı.....	33
Şekil 4.6 Yankı işaretinin NLMS algoritması kullanılarak elde edilen kestirimi	34
Şekil 4.7 NLMS karesel hatası NLMS MSE'nin 40.000 örnekte dağılımı.....	35
Şekil 4.8 NLMS Algoritmasında 40.000 örnek için 12,30 dB ortalamalı ERLE dağılımı.....	35
Şekil 4.9 Yankı işaretinin RLS algoritması kullanılarak elde edilen kestirimi	36
Şekil 4.10 RLS karesel hatası RLS MSE'nin 40.000 örnekte dağılımı	37
Şekil 4.11 RLS Algoritmasında 40.000 örnek için 7,15 dB ortalamalı ERLE dağılımı.....	37
Şekil 4.12 Yankı işaretinin KFD bölgesi LMS algoritması kullanılarak elde edilen kestirimi.....	38

Şekil 4.13 KFD bölgesi LMS karesel hatasının 40.000 örnekte dağılımı.....	39
Şekil 4.14 KFD bölgesi LMS Algoritmasında 40.000 örnek için 13,37 dB ortalama ERLE dağılımı.....	40

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 2.1 KFD'nin özellikleri	13
Çizelge 4.1 LMS algoritması için ERLE değerinin μ parametresine göre değişimi	32
Çizelge 4.2 KFD bölgesi LMS algoritması için ERLE değerinin μ parametresine göre değişimi	39
Çizelge 4.3 Akustik yankı gideriminde KFD bölgesi LMS algoritmasının “ a ” derecesinin güncellenmesi	41

ÖNSÖZ

Bu güncel tezin hazırlanmasında değerli zamanını bana ayıran, sahip olduğu bilgi birikimini benimle paylaşan, her fırsatta araştırmaya teşvik eden ve birlikte çalışmaktan mutluluk duyduğum değerli hocam Sn.Yrd. Doç. Dr. Lütfiye DURAK, Araş. Gör. Sultan ALDIRMAZ ve manevi desteğini her zaman hissettiğim eşime ve aileme teşekkürlerimi sunarım.

Pınar ARSLAN

IP TABANLI SES İLETİMİ ORTAMINDA AKUSTİK YANKI GİDERİMİ

Bu çalışmada, internet protokolü (Internet Protocol, IP) tabanlı bir ses iletimi ortamında akustik yankı giderimi için Kesirli Fourier Dönüşümü (KFD) bölgesi en küçük ortalama kareler (Least Mean Squares, LMS) algoritmasına dayalı uyarlanır süzgeçleme önerilmektedir. Çoğunlukla konferans tipi haberleşmenin gerçekleştirildiği IP tabanlı ses iletiminde uzak-uç bölgesinden gelen uzak-uç konuşma işareti, yakın-uç bölgesine ulaşır hoparlörden çıktıktan sonra gecikme ve zayıflamaya uğrayarak uzak-uç yankısını oluşturmaktadır. Oluşan bu yankı işareti yok edilmezse, mikrofon işaretine eklenir ve uzak-uç bölgesine yakın-uç işareti ile birlikte iletilir. Böylece ses kalitesi oldukça düşük bir haberleşmeye neden olunur. Zaman ve kesirli Fourier bölgelerinde uyarlanır süzgeçleme LMS, normalize LMS (NLMS), özyineli en küçük kareler (Recursive Least Squares, RLS) ve KFD bölgesinde LMS algoritmaları ile akustik yankı giderimi sağlanmıştır ve başarımlar performans analizleri incelenmiştir. Bu tip uyarlanır süzgeçleme algoritmalarında, istenilen yankı işaretini kestirmek için uzak-uç işareti süzgeç girişine sokulur. Her adımda, istenilen işaret ile istenilen işaretin kestirimi arasındaki fark hesaplanarak hata işareti bulunur. Süzgeç katsayıları, önceki süzgeç katsayıları, giriş işareti ve hata işaretine göre güncellenir. Seçilen uyarlanır süzgeçleme algoritmasına göre servis kalitesi daha yüksek ve en iyi yakınsama sağlanır. Bunlar doğrultusunda, performans değerlendirmesi açısından Yankı Geri Dönüş Kayıp Artışı (Echo Return Loss Enhancement, ERLE) analizleri yapılarak, akustik yankı gideriminde en iyi performansı KFD bölgesinde LMS algoritmasının ulaştığı gösterilmektedir.

ACOUSTIC ECHO CANCELLATION IN IP BASED VOICE TRANSMISSION

In this study, a Fractional Fourier Transform (FrFT) domain adaptive filtering scheme employing least mean square (LMS) algorithm is proposed for acoustic echo cancellation in an IP based voice transmission system. In IP based systems usually conference type communication is realized and far-end speech signal from far-end site arrives to near-end site and an attenuated and delayed form of the far-end speech signal in the near end generates the echo. If this echo signal is not cancelled, it is added to the microphone signal and transmitted to the far-end site. Thus, a very low quality of service is obtained. To overcome this problem, echo cancellation algorithms are developed by applying adaptive filters through LMS, normalized LMS (NLMS), recursive least squares (RLS) and FrFT domain LMS algorithms in time and FrFT domains. The performance of the algorithms are analyzed and compared to each other. In these type of adaptive filter algorithms, far-end signal is the input of the filter to estimate the echo signal (desired signal). In each step, error signal is evaluated from the difference between the desired signal and its estimate. Filter coefficients are updated by previous filter coefficients, input and error signals. Better quality of service and best convergence are achieved by the selected adaptive filter. As a result, it is shown that, FrFT domain LMS-based adaptive filtering algorithm has the best performance in terms of Echo Return Loss Enhancement (ERLE) values.

1. GİRİŞ

Orijinal bir ses işaretinin gecikmiş, bozulmaya uğramış ve daha sonra kaynağa geri yansımış olgusu olan yankı işaretinin, haberleşme servis kalitesini arttırmak için çeşitli yöntemler kullanarak giderilmesi gerekir. Haberleşme sistemlerinde yankı giderimi, ilk olarak 1960'lı yıllarda uyarlanır süzgeçlere dayalı tekniklerin kullanımıyla başlamıştır.

Pratik uygulamalar ise 1980 yılında çok büyük çapta tümleşik (Very Large-Scale Integrated, VLSI) yankı giderici tasarlanmasıyla gerçekleştirilmiştir (Murano vd., 1990). Gerçek zamanlı uygulamaları sayısal işaret işlemci (Digital Signal Processing, DSP) kartları kullanılarak yapılabilir (Chew ve Boroujeny, 1997). Yankı giderimi için öncelikle sonlu darbe cevaplı (Finite Impulse Response, FIR) veya sonsuz darbe cevaplı (Infinite Impulse Response, IIR) uyarlanır süzgeçleme algoritmaları ile geliştirilir ve DSP kartına yüklenir (Tahernezehadi ve Liu, 1995; Eneman ve Moonen, 1997).

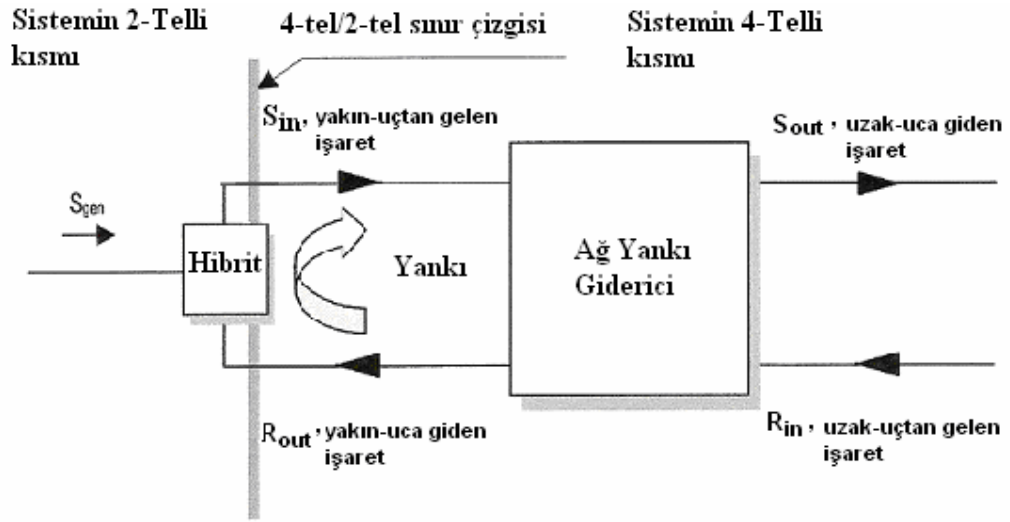
Haberleşme sistemlerinde yankı, iki telli sistemi dört telliye dönüştüren hibritteki empedans uyumsuzluğundan kaynaklanan hibrit (ağ) yankı ve kapalı bir ortamın akustiğinden kaynaklanan akustik yankı olmak üzere iki çeşittir. Akustik tip yankıları gidermek için en küçük ortalama kareler (Least Mean Squares, LMS), normalize LMS (NLMS) ve özyineli en küçük kareler (Recursive Least Squares, RLS) gibi birçok uyarlanır süzgeçleme algoritmaları kullanılmıştır (Matić ve Abadžić, 2007; Kao 2003; Rages ve Ho, 2002; Ahgren, 2005; Yuhong vd., 2005).

1.1 Hibrit Yankı Giderimi

Basit bir telefon ağı, 4- telli merkezi ağ ve 2-telli yerel ağ olmak üzere iki tip tel içerir. 2-telli ağ abone döngüsünü ve yerel ağın bazı kısımlarını içerir. 4-telli merkezi ağ ise işaret iletiminde iki yönü birbirinden ayırır ve her biri için bir çift tel kullanır. Diğer taraftan, 2-telli yerel ağ, her iki yöndeki işaretleri aynı çift tel üzerinde taşır. Bu durumda dönüştürücü bir aygıtı ihtiyaç vardır. Bunu sağlayan aygıtı hibrit denir (He vd., 2003).

Haberleşme sistemlerinde, 2-4 tel dönüşümünü sağlayan hibrit dönüştürücülerde empedans uyumsuzluğundan kaynaklanan hibrit yankı (ağ yankısı), uydu haberleşmesi, otomatik arama transferi gibi ses iletiminde, V.32 veri modemindeki gibi ses-bantlı tüm çift yönlü veri iletimi veya tümleşik hizmetler sayısal şebekesi (Integrated Services Digital Networks, ISDN) tip kullanıcı döngüsündeki gibi ana bantlı tüm çift yönlü veri iletiminde görülebilir.

Yankı giderici, Şekil 1.1’de gösterildiği gibi dört portlu bir aygıttır. Bu portlardan ikisi giriş portudur. Sistemde R_{in} alış yolu girişi veya uzak-uç işareti ve S_{in} gönderim yolu girişi veya yakın-uç işareti ile uzak-uç işaretinin yankısının karışımı ile simgelenmektedir. Diğer iki port ise çıkış portudur. R_{out} alış yolu çıkışı, G.168’e göre R_{in} ile aynı ve S_{out} gönderim yolu çıkışı (yankı gidericiden çıkış) olarak gösterilmektedir. Uzak-uç kullanıcısı konuştuğu zaman, konuşma işareti yakın-uç yankı yolu boyunca taşınır ve işaretin bir kısmı, yansiyarak uzak-uç dinleyiciye geri döner. Elektrik yankısı da denilen bu yankı, hibrit empedans uyumsuzluğundan kaynaklanır (He vd., 2003).



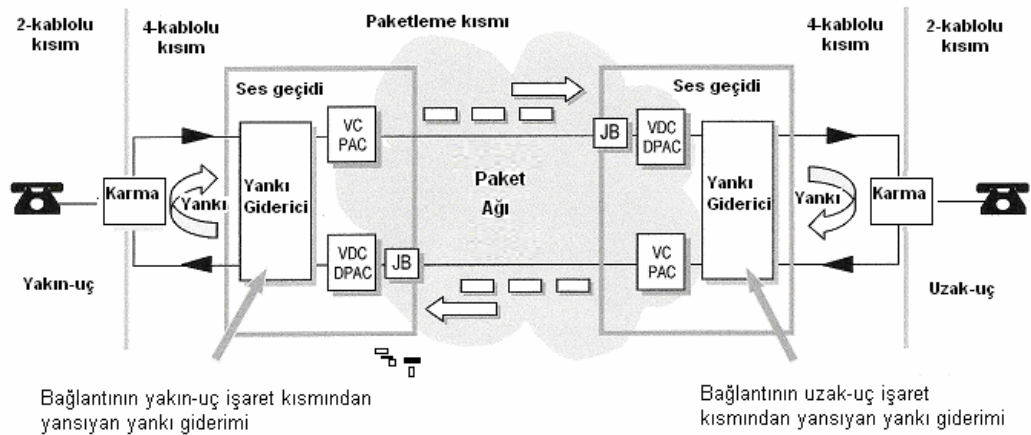
Şekil 1.1 Hibrit dönüştürücüde empedans uyumsuzluğundan kaynaklanan yankı ve hibrit yankı gidericinin yapısı

Yankı benzetimini sağlayan uyarlanır süzgeç Şekil 1.2’de gösterildiği gibi çift konuşma algılayıcı (Double Talk Detector, DTD), yüksek geçiren süzgeç (YGS), doğrusal olmayan işlemci (Nonlinear Processor, NLP), gürültü kaynağı, işaretleme tonu algılayıcı (dış kaynaklı tonları algılayıcı) ve yankı üretici kontrol biriminden oluşur.

PSTN ağ uygulamalarında, yankı gidericiyi düzgün bir şekilde çalıştırmak için sağlanması gereken asgari gereksinimler vardır. Bu gereksinimler, hibritin Yankı Geri Dönüş Kaybı (Echo Return Loss, ERL)'nin değerinin asgari değerini (tipik olarak en az 6dB olmalı) ve yankı yolu gecikmesinin azami değerini içermelidir (ITU G.168, 2007).

Paket haberleşmesindeki yankı gidericilerin yerleşimi PSTN ağlardakine eşdeğerdir. Bu ağdaki uygulaması olarak Şekil 1.4'te VC PAC, Ses Kodlayıcı/Paketleyici, VDC DPAC, Ses Kod Çözücü/Paket Ayırıcı ve JB, Seğirme Belleği görülmektedir. PSTN ağ uygulamalarındaki gibi, yankı gidericinin düzgün bir şekilde çalıştırılması için hibritin yankı dönüş kaybı ERL'nin değeri tipik olarak en az 6dB olmalı ve yankı yolu gecikmesinin azami değeri dikkate alınmalıdır. Bir paket haberleşmesi ağı için gidiş-geliş gecikmesi, PSTN ağlardaki gibi sadece iletim gecikmesini içermez; ses sıkıştırması/sıkıştırmayı açma, seğirme belleği gibi işleme gecikmesini de içerir.

Paket tabanlı yankı giderimi uygulamalarında, uyarlanır süzgeçleme NLMS algoritması kullanılarak hem darbe kod kiplenimi (Pulse Code Modulation, PCM) hem de uyarlanır darbe kod kiplenimi (Adaptive Pulse Code Modulation, ADPCM) tip konuşma işareti ile ağ yankısı yok edilmiştir ve ulaşılan sonuçlar doğrultusunda yankı gidericinin ADPCM'i tercih ederek daha iyi performans elde edebileceği kanıtlanmıştır (Periakarruppan ve Abdul Rashid, 2005). Literatürde NLMS, orantılı NLMS (Proportionate NLMS, PNLMS), iyileştirilmiş PNLMS (Improved PNLMS, IPNLMS) ve Cui vd.'nin çalışmasında ise yeni bir iyileştirilme yapılmış IPNLMS (Improved IPNLMS, IIPNLMS) ile paket tabanlı ağlarda yankı giderimi uygulamaları yapılmıştır.



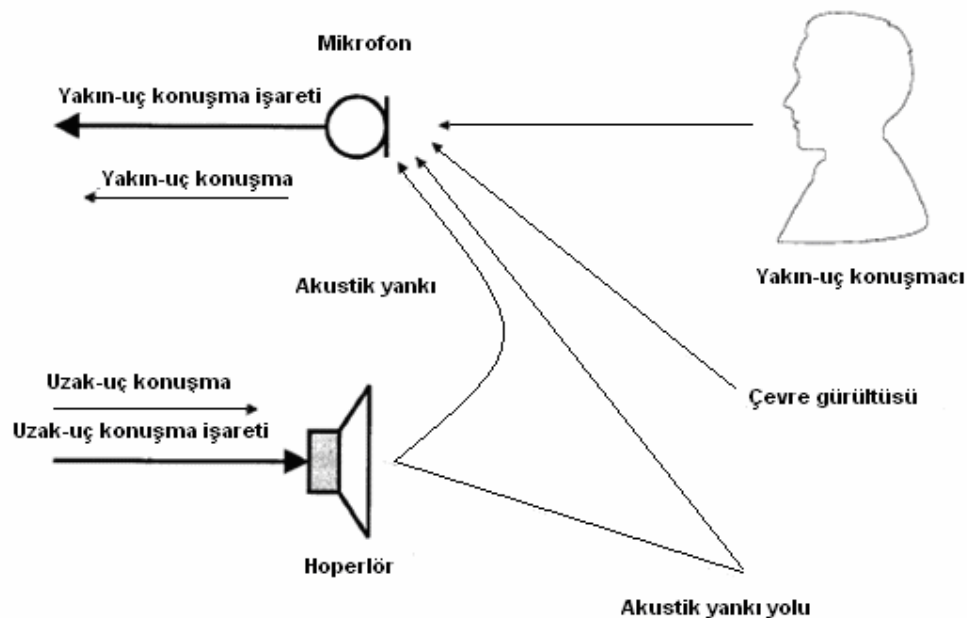
Şekil 1.4 Paket telefon ağlarında yankı gidericinin yapısı

IIPNLMS’e dayalı algoritmalar yankı giderimi ve gürültü yakınsamasında diğer uyarlanır süzgeçleme algoritmalarında göre daha iyi sonuç vermiştir. Bununla birlikte hesaplama karmaşıklığı da azaltılmıştır (Cui vd., 2004).

1.2 Akustik Yankı Giderimi

Haberleşme sistemlerinde kapalı ortamın akustüğinden kaynaklanan akustik yankı, hoparlörden çıkan ses işaretinin (uzak-uç işareti) gecikmeye ve zayıflamaya uğramasıyla oluşur ve mikrofondan iletilecek olan ses işaretine (yakın-uç işareti) eklenir. Akustik yankı, arabalarda kullanılan el-bağılantısız telefonlar, hoparlör-tip veya el-bağılantısız tipte kullanılan telefonlar ya da konferans telefonları ve bunun gibi daha başka birçok haberleşme uygulamasında sıklıkla karşılaşılan bir problemidir.

Şekil 1.5’te görüldüğü gibi, giriş işaretinin (uzak-uç konuşma işareti), yakın-uç konuşmacının hoparlöründen çıkarak, bulunduğu odadan kaynaklanan gecikme ve zayıflamalara uğrayıp, çevredeki gürültüyle beraber yakın-uç konuşmacının işaretiyle birlikte mikrofondan uzak-uca iletilir. Akustik yankı giderimi problemini çözmek için en sık kullanılan yöntem uyarlanırlı süzgeçleme uygulamalarıdır (Breining vd., 1999).



Şekil 1.5 Uzak uç işaretinin gecikmeye ve zayıflamaya uğramasıyla oluşan akustik yankı

Uzak-uç konuşma işareti $\mathbf{u}(n)$, yakın-uç konuşma işareti $\mathbf{b}(n)$, zayıflama etkisi a_k ve gecikmeyi n_k ile ifade etmek üzere ayrıık mikrofon işareti $\mathbf{m}(n)$;

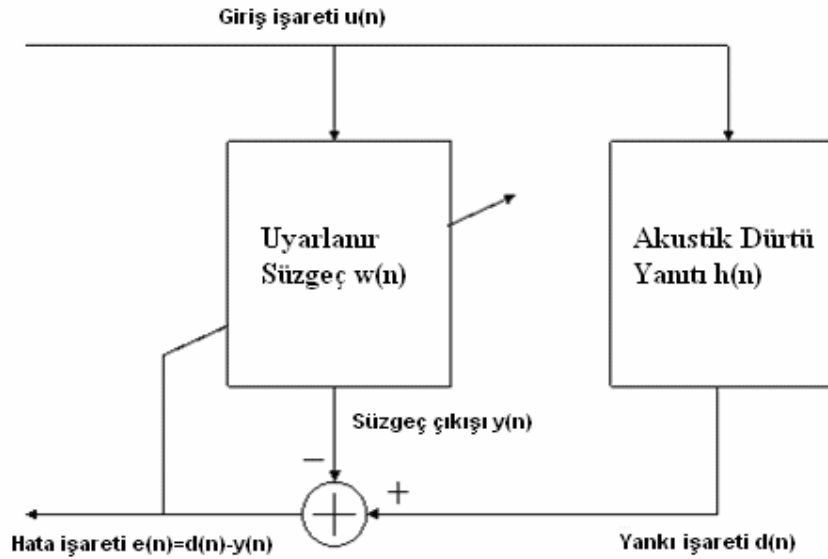
$$\mathbf{m}(n) = \sum_{k=1}^K a_k \mathbf{u}(n - n_k) + \mathbf{b}(n) \quad (1.1)$$

şeklinde tanımlanabilir. Akustik yankı gideriminde uyarlanır süzgeçleme algoritmalarının amacı mikrofon işareti $\mathbf{m}(n)$ 'den gecikmiş ve zayıflamaya uğramış uzak-uç konuşma işaretini (uzak-uç yankı işaretini) yok etmektir. Akustik yankı gideriminde uyarlanır süzgeç yapısına bakıldığında Şekil 1.6'da gösterildiği gibi, akustik ortamın dürtü yanıtı olan $\mathbf{h}(n)$, yankı giderimi için kullanılan uyarlanır süzgeç katsayıları $\mathbf{w}(n)$ ile kestirilmeye çalışılır. Burada uyarlanır süzgecin amacı, süzgeç çıkış işareti $y(n)$ 'i, istenilen işaret $d(n)$ 'e (akustik çevredeki yankı işareti) olabildiğince yakın kestirebilmektir. Her iterasyondaki hata

$$e(n) = d(n) - y(n) \quad (1.2)$$

ifadesiyle elde edilir ve süzgeç karakteristikleri bu değere bağlı olarak güncellenir.

Bugüne kadar yankı giderimi için uyarlanır süzgeçleme algoritmalarının kullanıldığı birçok çalışma yapılmıştır (Radecki vd., 2002; Choudhry vd., 2006; Yensen, vd.



Şekil 1.6 Uyarlanır süzgeçleme kullanılarak elde edilen kapalı ortamın akustik dürtü yanıtı modeli

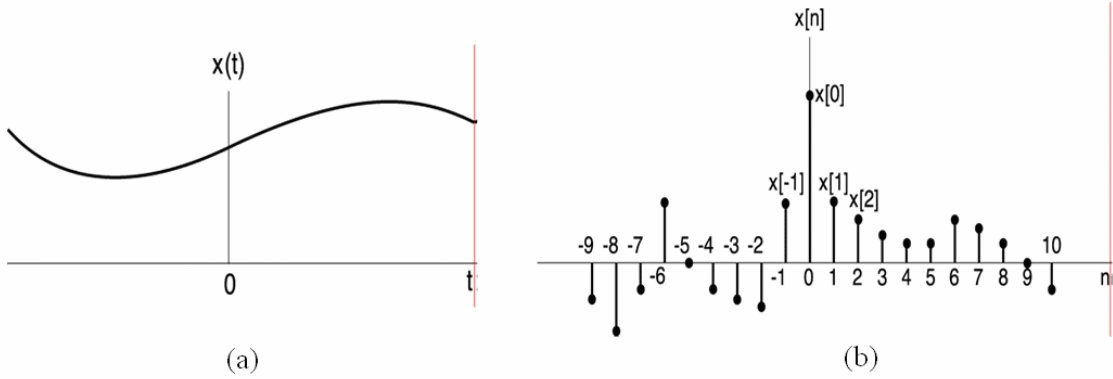
2001; Gordy ve Goubran, 2006; Jeannes vd., 2001; Zhongwei ve Tahemezhadi, 1996). IP tabanlı sistemlerde, yankı giderimi probleminin genel davranışından bahseden Radecki vd., 2002’de, zaman bölgesi uyarlanır süzgeçleme algoritmalarından RLS kullanılarak yankı giderilmiş ve hızlı yakınsama sağlanmıştır. Çok-aboneli konferans haberleşmesinde, zaman bölgesi uyarlanır süzgeçleme LMS algoritması kullanılarak ve yüksek derecede uyarlama sağlayarak akustik yankı giderilmiştir (Choudhry vd., 2006). El-bağılantısız konferans uygulamalarında yankı işaretinin, zaman bölgesi uyarlanır süzgeçleme algoritmalarından NLMS ile sentetik stereo kestirimi üretilmiş ve bunu kullanarak yankı sistemden yok edilmiştir (Yensen vd., 2001). IP ağı geniş bant telefon haberleşmesinde, zaman bölgesi NLMS algoritması kullanılarak, insan işitme sisteminin fizyokustik sınırlarının bakış açısıyla yankı giderimi performansı ölçümleri yapılmıştır (Gordy ve Goubran, 2006). IP ağlarda da kullanılan el-bağılantısız sistemlerdeki gürültü ve yankı gideriminin detaylı bir araştırmasında tek ya da iki kanallı sistemlerde, bu çift giderme için kullanılan zaman ve frekans bölgesi uyarlanır süzgeçleme teknikleri anlatılmıştır (Jeannes vd., 2001). Tüm bant ve alt bant akustik yankı giderimi, frekans bölgesi uyarlanır süzgeçleme LMS algoritması kullanılarak, zaman bölgesine göre daha hızlı yakınsama elde edilmiştir (Zhongwei ve Tahemezhadi, 1996).

Akustik yankı gideriminde dikkat edilmesi gereken hususlardan biri de çift konuşma algılandığında uyarlanır süzgeçlemeyi durdurmaktır. Bunun için DTD teknikleri kullanılarak, mikrofön işareti hem yankı işaretini hem de yakın-uç işaretini içerdiğinde uyarlanır süzgeçleme durdurulur (Valin, 2007; Ahgren, 2005; Jenq ve Hsiesh, 2001; Gansler vd., 1999).

Bu tezde IP ses iletim ortamında akustik yankı giderimi amaçlanmaktadır. Bölüm 2’de dönüşüm bölgesinde işaret analizi açıklanmakta, Bölüm 3’te tez kapsamında uygulaması yapılan uyarlanır süzgeçleme algoritmalarına yer verilmektedir. Bölüm 4, IP tabanlı ses iletimi ortamında akustik yankı giderimi için yapılan uygulamaları, analizleri ve uygulama sonuçlarını içermektedir. Sonuç bölümü olan Bölüm 5, akustik yankı giderimi için yapılan uygulamalar karşılaştırılmakta ve önerilen Kesirli Fourier Dönüşüm (KFD) bölgesi LMS algoritmasının diğer algoritmalara göre daha iyi sonuç verdiği tartışılmaktadır.

2. DÖNÜŞÜM BÖLGESİNDE İŞARET ANALİZİ

Sayısal işaret işlemede işaretler genel olarak sürekli zaman işaretleri veya ayrık zaman işaretleri olmak üzere iki şekilde ele alınmaktadır. Sürekli zaman işaretleri, Şekil 2.1a'da görüldüğü gibi zamanda sürekli değerler alır ve birçok fiziksel işaret, örneğin, konuşma işareti, elektrik akımı-gerilimi, sıcaklık, basınç veya yoğunluk yapısal olarak sürekli zaman işaretlerindendir.



Şekil 2.1 a) Sürekli zaman işareti b) Ayrık zaman işareti

Ayrık zamanlı işaretler sürekli zaman işaretinden alınmış örneklerle bir zaman serisi şeklinde Şekil 2.1b'deki gibi elde edilebilir. Ses ya da müzik işaretinin sürekli zaman işareti olarak üretildiği durumda, bu işaretler iletilirken ayrık zamanlı işarete dönüştürülmelidir. Bu işlem yapılırken, düzenli aralıklarla (T_s periyodu) sürekli zaman işaretinin anlık değerlerini alınır ve buna örnekleme denir.

Bir $x(t)$ sürekli zaman işaretini sayısallaştırmak için ayrık zamanlı vektöre dönüştürmek gerekir. Vektördeki her değer ise örnekleme periyodunun tam katlarında bu dalga şeklinin anlık değerlerini gösterir:

$$x_n = x(nT_s) \quad (2.1)$$

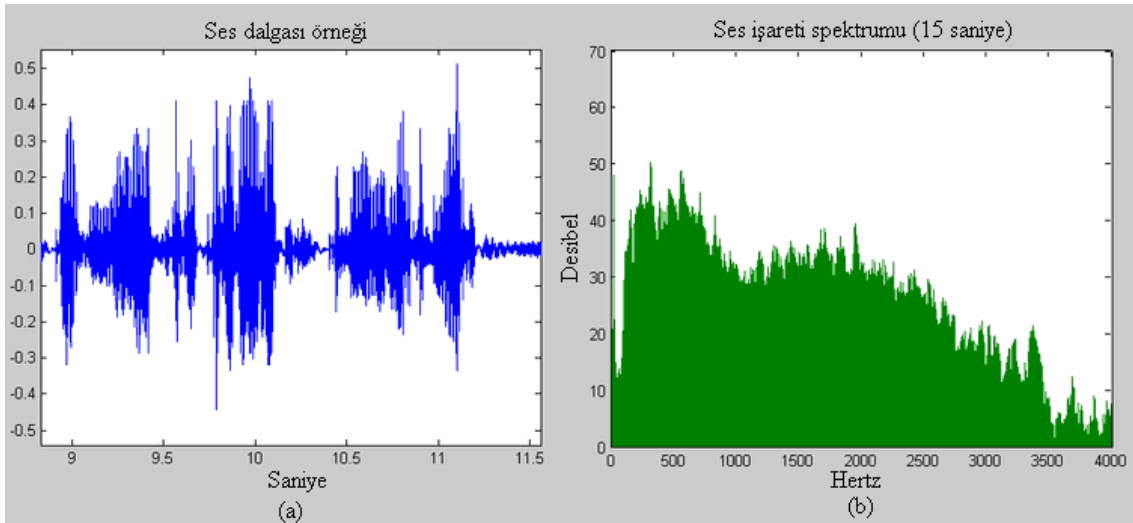
2.1 Dönüşüm Bölgeleri

Yankı giderimi için kullanılan uyarlanırlı süzgeçler zaman bölgesi analizinin yanı sıra, farklı dönüşüm bölgelerinde de ele alınabilir. Bu dönüşüm bölgeleri, frekans ve kesirli Fourier bölgelerini içerir. Uyarlanırlı süzgeçleme algoritmalarında ayrık zamanlı işaretler kullanılarak,

yakınsamasına göre, yankı gideriminde hangi dönüşüm bölgesinin daha uygun olduğu saptanmalıdır.

Zaman bölgesi, uyarlanır süzgeçleme uygulamalarında oldukça sık kullanılan bir bölgedir. Bu tezin uygulamaları arasında da yer alan zaman bölgesi ve diğer bölge uygulamalarında, ses işareti zamana bağlı olarak alınır ve örneklenerek ayrık zamanlı işarete dönüştürüldükten sonra sisteme sokulur.

Frekans bölgesi, işaretin frekansının referans alındığı bölgedir. Herhangi bir konuşma işaretine bakıldığında, zaman bölgesi gösterimi, zamanla işaretin değişimini gösterirken, frekans bölgesi gösterimi ise işaretin frekans değerlerindeki bileşenlerini verir. Şekil 2.2.a ve 2.2.b'de bir konuşma işaretinin hem zaman hem de frekans bölgesindeki dağılımı gösterilmiştir.



Şekil 2.2 Bir konuşma işaretinin a) Zaman bölgesi b) Frekans bölgesi gösterimleri

Zaman bölgesinden frekans bölgesine dönüşümde sonlu ve sonsuz sayıdaki frekanslar için Fourier Serileri veya Fourier Dönüşümü kullanılır.

2.1.1 Frekans Bölgesi Analizi

Haberleşme ve işaret işlemede, bir $x(t)$ sürekli zaman fonksiyonunun Fourier dönüşümü gerçel sayı olan her f değeri için;

$$F\{x\}(f) = X(f) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft} dt \quad (2.2)$$

şeklinde elde edilir.

Dönüşüm değişkeni frekans ise, $X(f)$ 'in ters Fourier dönüşümü, gerçel olan her t değeri için;

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{i2\pi ft} df \quad (2.3)$$

biçiminde tanımlanır. $X(f)$ 'in, $A(f)e^{i\phi(f)}$ şeklinde ifade edilmesi durumunda, genliği;

$$A(f) = |X(f)| \quad (2.4)$$

faz açısı ise ($t=0$ 'da);

$$\phi(f) = \angle X(f) \quad (2.5)$$

olur.

$X(f)$ 'in ters Fourier dönüşümü ise

$$x(t) = \int_{-\infty}^{\infty} A(f)e^{i(2\pi ft + \phi(f))} df. \quad (2.6)$$

şeklinde ifade edilir.

Fourier dönüşümünün genelleştirilmiş hali olan kesirli Fourier dönüşümü Bölüm 2.1.2'de anlatılmaktadır.

2.1.2 Kesirli Fourier Dönüşüm Bölgesi

Kesirli Fourier Dönüşümü (KFD), klasik Fourier dönüşümünün doğrusal bir dönüşümle genelleştirilmiş halidir. Klasik Fourier dönüşümünün, kesirli derecelerden hesaplanabilmesine olanak sağlayan KFD, zaman-frekans düzleminde kesirli bir açıyla dönme gibi yorumlanabilir (Almeida, 1994). Örneğin, $x(t)$ işaretinin “ a .” dereceden KFD'si;

$$x_a(t) = F_a\{x(t)\} = \int K_a(t, t')x(t')dt' \quad (2.7)$$

şeklinde tanımlanır.

Kesirli parametre “ a ”, $0 < |a| < 2$ aralığındadır. Dönüşüm çekirdek fonksiyonu $K_a(t, t')$;

$$K_a(t, t') = A_\varphi \exp(j\pi(t^2 \cot \varphi - 2tt' \csc \varphi + t'^2 \cot \varphi)) \quad (2.8)$$

biçiminde ifade edilir. Burada kullanılan A_φ parametresi aşağıdaki şekilde tanımlanır:

$$A_\varphi = \exp(-j\pi \operatorname{sgn}(\sin \varphi) / 4 + j\varphi / 2) / |\sin \varphi|^{1/2}. \quad (2.9)$$

φ ise dönüşüm açısını göstermektedir (Mendlovic ve Özaktas, 1993; Özaktas vd., 2001).

KFD derecesi ile dönüşüm açısının ilgisi;

$$\varphi = a\pi / 2 \quad (2.10)$$

olarak verilir.

KFD'nin özellikleri Çizelge 2.1'deki gibi sıralanabilir. İlk özelliğinden de görüldüğü gibi KFD doğrusaldır. Özellik 2'deki durum, KFD'nin katsayısı “ a ”, tamsayı olan j 'ye eşit olduğu zaman “ a .”dereceden KFD, klasik Fourier dönüşümünün j . tamsayı kuvveti olmasıdır. Aynı zamanda $F^2 = P$ (eşitlik operatörü), $F^3 = F^{-1} = (F)^{-1}$ (ters dönüşüm operatörü), $F^4 = F^0 = I$ (özdeş operatörü) ve $F^j = F^{j \bmod 4}$ ifadeleri Özellik 2 kullanılarak elde edilir. Özellik 3, ileri yönde dönüşümlü pozitif dereceleri, ters dönüşümlü negatif dereceleri birleştirme olanağı sunar. Çekirdek fonksiyonunda ifade edilirse bu özellik;

$$K_a^{-1}(u, u') = K_{-a}(u, u') \quad (2.11)$$

şeklinde gösterilebilir. Özellik 4'te çekirdek fonksiyonuyla gösterildiğinde;

$$K_a^{-1}(u, u') = K_a^*(u, u') \quad (2.12)$$

elde edilir ve Özellik 3 ve 4 birleştirildiğinde;

$$(F^a)^H = F^{-a} \quad (2.13)$$

veya

$$K_a^{-1}(u, u') = K_{-a}^*(u, u') = K_{-a}^H(u, u') \quad (2.14)$$

eşitliği elde edilebilir. Özellik 5 olan toplanabilirlik özelliği çekirdek fonksiyonu

$K_{a_1+a_2}(u, u')$ 'a uygulanırsa;

$$K_{a_1+a_2}(u, u') = \int K_{a_2}(u, u'') K_{a_1}(u'', u') du'' \quad (2.15)$$

eşitliği elde edilir. Özellik 6, Özellik 5'ten türetilmiştir. Özellik 7'ye yalnızca kesirli Fourier dönüşümü değil tüm doğrusal kanonik dönüşümler sahiptir. Özellik 8'de görüldüğü gibi KFD'nin özdeğerleri Hermite-Gaussian fonksiyonları, $\psi_1(u)$ 'dır. Özellik 9'da Wigner dağılımı ifade edilmiştir. Özellik 10 ise birimsellik özelliğine denktir ve bunun özel bir durumu olan enerji koruması;

$$En[f] = En[f_a] \quad (2.16)$$

veya

$$\|f\| = \|f_a\| \quad (2.17)$$

şeklinde ifade edilebilir. Bu özelliğin genel ifadesi;

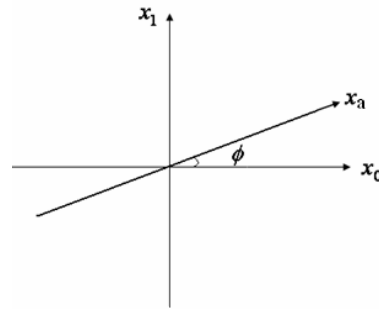
$$\langle f_{a_1}(u), g_{a_2}(u) \rangle = \langle f_{a_3}(u), g_{a_4}(u) \rangle, \quad a_1 - a_2 = a_3 - a_4 \quad (2.18)$$

şeklinde de yazılabilir (Özaktas vd., 2001).

Çekirdek fonksiyonu $K_a(t, t')$, Özellik 2'de de ifade edildiği gibi $a=1$ için $e^{-j2\pi t t'}$ olur ve birinci dereceden KFD dönüşümü, fonksiyonun Fourier dönüşümünü verir. Diğer bir durumda, $a=0$ için, çekirdek fonksiyonu, $\delta(t)$ 'ye yaklaşır ve sıfırıncı dereceden KFD özdeş dönüşüm olarak fonksiyonun kendisini verir. Fourier dönüşümünün tanımlandığı tüm fonksiyonlar için KFD de tanımlıdır. Dolayısıyla işaretlerin zaman ya da frekans gösterimlerinin zaman ve frekans bölgelerinde tanımlanabilmesi gibi, KFD kullanılarak gerçekleştirilen dönüşüm ile kesirli Fourier bölgelerinde ifade edilmeleri mümkündür. Zaman-frekans bölgesi üzerinde düşünüldüğü zaman, kesirli Fourier bölgelerinin zaman ve frekans bölgelerinin arasını aradeğerleyen bölgeler olduğu ortaya çıkmaktadır. Şekil 2.3'te bir zaman-frekans bölgesi üstünde “a.” dereceden KFD bölgesi gösterilmektedir (Özaktas vd., 1996).

Çizelge 2.1 KFD'nin özellikleri

1	Doğrusallık	$F^a \left[\sum_j a_j f_j(u) \right] = \sum_j a_j \left[F^a f_j(u) \right]$
2	Tamsayı üstel	$F^j = (F)^j$
3	Tersinire Çevirme	$(F^a)^{-1} = F^{-a}$
4	Birimsellik	$(F^a)^{-1} = (F^a)^H$
5	Toplanabilirlik	$F^{a_2} F^{a_1} = F^{a_2+a_1}$
6	Değişme Özelliği	$F^{a_2} F^{a_1} = F^{a_1} F^{a_2}$
7	Birleşme Özelliği	$F^{a_3} (F^{a_2} F^{a_1}) = F^{a_3} (F^{a_2} F^{a_1})$
8	Özdeğer ve Fonksiyonları	$F^a \psi_1 = \exp(-ial\pi/2) \psi_1$
9	Wigner dağılımına etkisi	$W_{f_a}(u, \mu) = W_f(u \cos \alpha - \mu \sin \alpha, u \sin \alpha + \mu \cos \alpha)$
10	Parseval ilişkisi	$\langle f(u), g(u) \rangle = \langle f_a(u), g_a(u) \rangle$



Şekil 2.3 Zaman-frekans bölgesi üstünde a . dereceden KFD bölgesi

3. ALTERNATİF UYARLANIR SÜZGEÇLEME YAPILARI

Akustik yankı giderimi uygulamalarında sıklıkla kullanılan yöntem olan uyarlanırlı süzgeçleme algoritmaları LMS ve RLS başlıkları altında incelenebilir. LMS metoduna göre ağırlıklı hata karelerinin toplamını içeren başarımlı göstergesi minimize edilir. RLS metodu ise uyarlanırlı çapraz süzgeçlerin tasarımı için özyineli algoritmaları geliştirmede en küçük kareler yöntemini kullanır ve $n-1$ özyinelemede süzgecin süzgeç katsayı vektörünün en küçük karesel kestirimi verilmişse, yeni veriye ulaşmada, n özyinelemede bu vektörün güncellenmiş kestirimi hesaplanır (Haykin, 1996).

LMS algoritması, bir çeşit olasılıksal eğim algoritmasıdır (Stochastic Gradient Algorithm, SGA). Belirli ardışık parametre kestirim problemlerini çözmek için kullanılan LMS ile SGA arasındaki temel fark şu şekilde açıklanabilir. LMS algoritması bir iterasyondan diğerine geçişte her bir çıktıya uygulanan düzeltmeyi kontrol etmek için sabit adım boyutu parametresi kullanırken, SGA metodunda adım boyutu parametresi, n 'nin kuvveti veya n zamanıyla ters orantılı yapılır. LMS algoritmasıyla yakın ilişkili başka bir olasılıksal eğim algoritması da eğim uyarlanırlı kafes (Gradient Adaptive Lattice, GAL) algoritmasıdır. Aralarındaki fark yapısal olup, GAL algoritması kafes tabanlı olmasına rağmen, LMS algoritması genellikle FIR ya da IIR süzgeçleri kullanır.

Uyarlanırlı süzgeç algoritmalarından RLS ailesi, standart RLS'in önerildiği 1950'den bu yana birçok araştırmacı tarafından oluşturulmaktadır. 1970'lerde geniş bir araştırma topluluğu, hesaplama karmaşıklığını azaltmak amacıyla LMS algoritmasıyla karşılaştırılabilir seviyede RLS algoritmaları geliştirmiştir (Haykin, 1996).

3.1 En Küçük Ortalama Kareler (LMS) Algoritması

Kullanım alanı oldukça geniş olan LMS uyarlama algoritması, Widrow ve Hoff (1960) tarafından geliştirilmiştir. Olasılıksal eğim algoritmaları ailesinin önemli bir üyesi olan LMS algoritması, uyarlanırlı süzgeç uygulamalarında oldukça sık kullanılan bir algoritmadır. LMS algoritmasında ilinti matrisinin hesaplanması ve matris evirmesi yapılmaz. Bu da algoritmanın kolayca hesaplanabilirlik özelliğine sahip olmasını sağlar. Ayrıca bu özelliği sayesinde diğer uyarlanırlı süzgeçleme algoritmalarında yargıya varma sürecinde de referans olarak kullanılabilir (Haykin, 1996).

LMS algoritması, iki temel işlemi içeren doğrusal bir uyarlanırlı süzgeçleme algoritmasıdır. Bunlar,

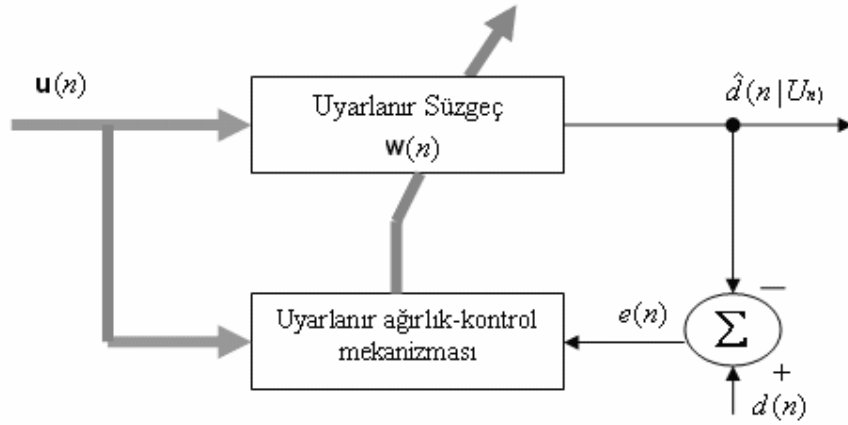
1. Süzgeçleme işlemi:

- Bir giriş işaretine cevap vermek için doğrusal süzgeç çıkışını hesaplamak
- Bu çıkışı bir cevapla karşılaştırarak bir kestirim hatası üretmektir.

2. Uyarlanırlık işlemi:

- Kestirim hatasıyla ilişkili bir süzgecin parametrelerini otomatik ayarlamayı içerir.

Bu iki işlemin birleşmesiyle Şekil 3.1’de gösterildiği gibi bir geri beslemeli döngü oluşur.



Şekil 3.1 Uyarlanırlı FIR süzgecin blok diyagramı

LMS algoritmasında FIR ya da IIR model yapıları kullanılabilir. Ancak burada FIR model yapısı kullanılarak LMS algoritması türetilmektedir çünkü hesaplaması daha kolaydır. Şekil 3.1’de uyarlanırlı ağırlık-kontrol mekanizması olarak adlandırılan mekanizma, FIR süzgecin gecikme-ağırlık üzerindeki uyarlanırlı kontrol işlemini gerçekleştirir. Gecikme girişleri $u(n), u(n-1), \dots, u(n-M+1)$, $M-1$ gecikme sayısı olmak üzere $\mathbf{u}(n)$ elamanlarını oluştururlar. $w_0(n), w_1(n), \dots, w_{M-1}(n)$; $\mathbf{w}(n)$ elemanlarını oluşturur. Süzgeçleme işlemi süresince, istenilen $d(n)$ cevabı elde edilir. Verilen bu girişe karşın FIR süzgeç, $d(n)$ ’nin kestirimi olan $\hat{d}(u|U_n)$ şeklinde bir çıkış üretir. İstenilen cevapla süzgecin gerçek çıkışı arasındaki fark olan $e(n)$ kestirim hatası olarak tanımlanır. Süzgeç katsayıları $\mathbf{w}(n)$ ve kestirim hatası $e(n)$, kontrol mekanizmasına uygulanır ve geri-besleme döngüsü kapatılır.

Eğer her n iterasyonda, eğim vektörü $\nabla J(n)$ tam olarak ölçülebilseydi ve uyum adım büyüklüğü μ uygun seçilseydi, en dik inme metodu kullanılarak hesaplanan katsayı vektörü optimum bir Wiener çözüme yakınsayacaktı (Haykin 1996). Ancak gerçek hayatta, süzgeç

katsayılarının ilinti matrisi $\mathbf{R}$ 'nin ve istenilen cevap ile süzgeç katsayıları arasındaki çapraz-ilinti vektörü $\mathbf{p}$ 'nin önceki bilgileri gerekli olması nedeniyle eğim vektörünün doğru ölçümü mümkün değildir. Burada, $\mathbf{R}$ ve $\mathbf{p}$ 'nin kestirimleri için en kolay seçenek, süzgeç katsayılarının giriş vektörünün örnek değerleri ve istenilen cevaba dayanan anlık kestirimlerin kullanılmasıdır ve;

$$\hat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}^H(n) \quad (3.1)$$

$$\hat{\mathbf{p}}(n) = \mathbf{u}(n)\mathbf{d}^*(n) \quad (3.2)$$

şeklinde tanımlanır. Ayrıca maliyet fonksiyonu $J(n)$;

$$J(n) = (d(n) - \mathbf{w}^T \mathbf{u}(n))^2 \quad (3.3)$$

şeklinde ifade edilebilir.

Sonuç olarak, eğim vektörü, bilinmeyen bir çevrede çalışıldığında mevcut verilerden yani ilinti matrisi $\mathbf{R}(n)$, çapraz-ilinti $\mathbf{p}(n)$ kestirimlerini kullanılarak;

$$\nabla J(n) = -2\mathbf{p}(n) + 2\mathbf{R}\mathbf{w}(n) \quad (3.4)$$

şeklinde kestirilebilir. (3.1) ve (3.2) eşitlikleri kullanılarak, (3.4) eşitliği yeniden düzenlenirse;

$$\nabla J(n) = -2\mathbf{u}(n)d^*(n) + 2\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{w}(n) \quad (3.5)$$

ifadesi elde edilir.

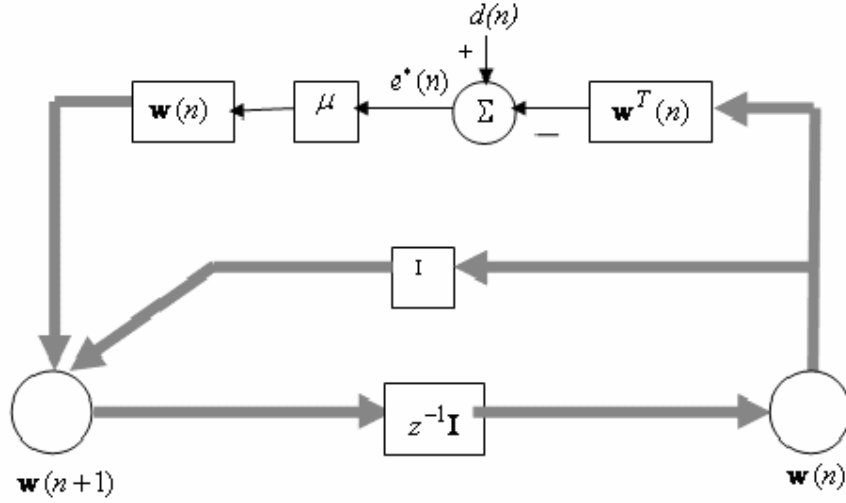
Süzgeç katsayılarının güncellenmesi, durağanlığı ve yakınsaklık oranını düzenleyen ve maliyet fonksiyonunun adım büyüklüğünü belirten μ sabiti ile çarpılarak önceki süzgeç katsayılarına eklenerek;

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla J(n)] \quad (3.6)$$

şeklinde yapılır. (3.5) eşitliği, (3.6)'da yerine konulursa, süzgeç katsayıları giriş vektörünün örnek değerleri ve istenilen cevaba göre;

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{u}(n)[d^*(n) - \mathbf{u}^H(n)\mathbf{w}(n)] \quad (3.7)$$

ifadesi ile güncellenir (Haykin, 1996).



Şekil 3.2 LMS algoritmasının işaret-akış diyagramı

Şekil 3.2'teki işaret-akış diyagramı LMS algoritmasının basitliğini açıkça göstermektedir. Bu diyagramda LMS algoritmasının $2M+1$ kompleks çarpım ve her bir iterasyonda $2M$ kompleks toplam içerdiği görülmektedir. M burada uyarlanırlı süzgeç katsayısının sayısıdır. $\mathbf{R}(n)$ ve $\mathbf{p}(n)$ 'nin anlık kestirimleri arasında büyük değişiklikler vardır. Bu nedenle, LMS algoritması anlık kestirimler kullandığından, ilk bakışta iyi bir başarıya sahip değil gibi görülebilir.

Süzgeç katsayılarını ters ilinti matrisi ve özdeğerler cinsinden yazıldığında;

$$\mathbf{w}(n+1)_j = \mathbf{w}(n)_j + \mu(\mathbf{p}_j - \lambda_j \mathbf{w}(n)_j) \quad (3.8)$$

eşitliği elde edilir. Her n değeri için (3.8) eşitliği yazılırsa $n+1$ 'deki süzgeç katsayıları;

$$\mathbf{w}(n+1)_j = \mu \mathbf{p}_j + \sum_{i=0}^n (1 - \mu \lambda_j)^i + (1 - \mu \lambda_j)^{n+1} \mathbf{w}(0)_j \quad (3.9)$$

olarak elde edilir ve yakınsama için;

$$|1 - \mu \lambda_j| < 1 \quad (3.10)$$

koşulu sağlanmalıdır. (3.10) eşitliği düzenlenirse;

$$0 < \mu < \frac{2}{\lambda_j} \quad (3.11)$$

haline dönüşür.

(3.7) eşitliği incelendiğinde, LMS algoritmasının kare almayı, ortalama almayı ya da türev almayı içermediği, dolayısıyla uygulanmasının oldukça kolay olduğu gözlemlenmektedir.

LMS algoritmasını uyarlamak için aşağıdaki üç basamağa ihtiyaç vardır;

1. Sonlu Dürtü Yanıtı (FIR) süzgeç çıkışı, $y(n)$ (3.12) eşitliği kullanılarak elde edilir:

$$y(n) = \mathbf{w}^H(n)\mathbf{u}(n) \quad (3.12)$$

2. Kestirim hatası olan $e(n)$ ise;

$$e(n) = d(n) - y(n) \quad (3.13)$$

şeklinde tanımlanır.

3. Süzgeç-katsayıları güncellemesi ise;

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{u}(n)e^*(n) \quad (3.14)$$

eşitliği ile ifade edilir.

Bu tezde akustik yankı giderimi için LMS algoritmasının performans analizi yapılmış ve ilerleyen bölümlerde anlatılan diğer uyarlanır süzgeçleme algoritmalarıyla performansı karşılaştırılmıştır.

3.2 Normalize LMS (NLMS) Algoritması

LMS algoritmasının standart formunda, $n+1$ özyinelemede, süzgeç katsayıları vektörü $\mathbf{w}(n)$ 'e eklenmiş ilinti $\mu \mathbf{u}(n)e^*(n)$, $\mathbf{u}(n)$ süzgeç katsayıları vektörüyle doğru orantılıdır. Bu nedenle, $\mathbf{u}(n)$ çok büyük olduğunda, LMS algoritmasında bir eğim gürültü yükseltici problemi ile karşılaşılır. Bu zorluğun üstesinden gelebilmek için, LMS algoritmasına eş olan *normalize* LMS (NLMS) algoritması kullanılabilir. $n+1$ özyinelemede süzgeç katsayıları vektörü $\mathbf{w}(n)$ 'e eklenmiş düzeltme, n özyinelemede süzgeç katsayıları vektörü $\mathbf{w}(n)$ 'nin karesel Öklit normu kullanılarak “normalize” edilir.

NLMS algoritmasında adım büyüklüğü giriş gücü ile normalize edildiğinden dolayı, yakınsamanın giriş işaretine olan bağımlılığı ortadan kalkar. NLMS algoritması uygulaması şu şekilde özetlenebilir. Süzgeç katsayıları vektörü $\mathbf{u}(n)$ ve istenilen cevap $d(n)$ verilmiş

olsun. Süzgeç katsayıları vektörü $\mathbf{w}(n)$ hesaplanmış olsun. Bu durumda değişimin karesel Öklit normu küçültülmesi;

$$\delta \mathbf{w}(n+1) = \mathbf{w}(n+1) - \mathbf{w}(n) \quad (3.15)$$

şeklinde elde edilir.

Eski $\mathbf{w}(n)$ değerine bağlı olarak süzgeç katsayıları vektöründe $\mathbf{w}(n+1)$, sınırlamasına bağlı olarak;

$$\mathbf{w}^H(n+1)\mathbf{u}(n) = d(n) \quad (3.16)$$

biçiminde istenilen cevap elde edilir.

Bu sınırlama eniyileme problemini çözmek için, Lagrange çoğaltıcı metodu kullanılabilir. Süzgeç katsayıları vektöründe $\mathbf{w}(n+1)$, değişimi gösteren $\delta \mathbf{w}(n+1)$ 'nin karesel normu;

$$\begin{aligned} \|\delta \mathbf{w}(n+1)\|^2 &= \delta \mathbf{w}^H(n+1) \delta \mathbf{w}(n+1) \\ &= [\mathbf{w}(n+1) - \mathbf{w}(n)]^H [\mathbf{w}(n+1) - \mathbf{w}(n)] \\ &= \sum_{k=0}^{M-1} |\mathbf{w}_k(n+1) - \mathbf{w}_k(n)|^2 \end{aligned} \quad (3.17)$$

olarak ifade edilebilir.

Süzgeç ağırlıkları $\mathbf{w}(n)$ 'leri, $k=0,1, \dots, M-1$ değerleri için tanımlanırsa, gerçel ve sanal kısımları;

$$\mathbf{w}(n) = a_k(n) + jb_k(n), \quad k = 0,1,\dots,M-1 \quad (3.18)$$

şeklinde yazabilir. Değişimdeki karesel norm ise;

$$\|\delta \mathbf{w}(n+1)\|^2 = \sum_{k=0}^{M-1} ([a_k(n+1) - a_k(n)]^2 + [b_k(n+1) - b_k(n)]^2) \quad (3.19)$$

eşitliğindeki gibi yazılır.

Gerçel ve sanal kısımlarını içeren giriş vektörü $\mathbf{u}(n-k)$ ve istenen cevap $d(n)$ ise;

$$\mathbf{u}(n-k) = \mathbf{u}_1(n-k) + j\mathbf{u}_2(n-k) \quad (3.20)$$

$$d(n) = d_1(n) + jd_2(n) \quad (3.21)$$

(3.16) eşitliğinden;

$$\sum_{k=0}^{M-1} (a_k(n+1)\mathbf{u}_1(n-k) + b_k(n+1)\mathbf{u}_2(n-k)) = d_1(n) \quad (3.22)$$

$$\sum_{k=0}^{M-1} (a_k(n+1)\mathbf{u}_1(n-k) - b_k(n+1)\mathbf{u}_2(n-k)) = d_2(n) \quad (3.23)$$

eşitlikleri elde edilir.

Gerçek değerli maliyet fonksiyonu $J(n)$, (3.19), (3.22) ve (3.23) eşitlikleri kullanılarak;

$$\begin{aligned} J(n) = & \sum_{k=0}^{M-1} ([a_k(n+1) - a_k(n)]^2 + [b_k(n+1) - b_k(n)]^2) \\ & + \lambda_1 [d_1(n) - \sum_{k=0}^{M-1} (a_k(n+1)\mathbf{u}_1(n-k) + b_k(n+1)\mathbf{u}_2(n-k))] \\ & + \lambda_2 [d_2(n) - \sum_{k=0}^{M-1} (a_k(n+1)\mathbf{u}_1(n-k) - b_k(n+1)\mathbf{u}_2(n-k))] \end{aligned} \quad (3.24)$$

biçiminde yazılabilir. Bu eşitlikte kullanılan λ_1 ve λ_2 değerleri Lagrange çarpanlarıdır.

$a_k(n+1)$ ve $b_k(n+1)$ 'in en iyi değerlerini bulmak için maliyet fonksiyonu $J(n)$ 'nin türevi alınır ve sıfıra eşitlenir:

$$\frac{\partial J(n)}{\partial a_k(n+1)} = 0 \quad (3.25)$$

(3.25) eşitliği açılırsa;

$$2[a_k(n+1) - a_k(n)] - \lambda_1 \mathbf{u}_1(n-k) - \lambda_2 \mathbf{u}_2(n-k) = 0 \quad (3.26)$$

sonucuna ulaşılır.

Benzer şekilde;

$$\frac{\partial J(n)}{\partial b_k(n+1)} = 0 \quad (3.27)$$

eşitliğini kullanılarak;

$$2[b_k(n+1) - b_k(n)] - \lambda_1 \mathbf{u}_1(n-k) - \lambda_2 \mathbf{u}_2(n-k) = 0 \quad (3.28)$$

denklemini elde edilir.

(3.18) ve (3.20) eşitlikleri yukarıdaki eşitlikte yerine koyulursa;

$$2[\mathbf{w}_k(n+1) - \mathbf{w}_k(n)] = \lambda^* \mathbf{u}(n-k), \quad k = 0, 1, \dots, M-1 \quad (3.29)$$

eşitliğine ulaşılır. Karmaşık Lagrange çarpanı λ 'nın gerçel ve sanal kısımları;

$$\lambda = \lambda_1 + j\lambda_2 \quad (3.30)$$

şeklinde ifade edilir.

Bilinmeyen λ^* 'yı bulabilmek için, (3.29) eşitliğinin her iki tarafını $\mathbf{u}^*(n-k)$ ile çarpıp, 0'dan M-1'e kadar olan k' nın olası tüm tamsayı değerleri toplanarak aşağıdaki sonuca ulaşılır:

$$\begin{aligned} \lambda^* &= \frac{2}{\sum_{k=0}^{M-1} \|\mathbf{u}(n-k)\|^2} \left[\sum_{k=0}^{M-1} \mathbf{w}_k(n+1) \mathbf{u}^*(n-k) - \sum_{k=0}^{M-1} \mathbf{w}_k(n) \mathbf{u}^*(n-k) \right] \\ &= \frac{2}{\|\mathbf{u}(n)\|^2} [\mathbf{w}^T(n+1) \mathbf{u}^*(n) - \mathbf{w}^T(n) \mathbf{u}^*(n)] \end{aligned} \quad (3.31)$$

$\|\mathbf{u}(n)\|^2$, $\mathbf{u}(n)$ giriş vektörünün Öklit normudur. (3.31) eşitliğinden yararlanılarak (3.16) eşitliğinin kompleks sınırı ve λ^* parametresinin aşağıdaki ifadesi elde edilir:

$$\lambda^* = \frac{2}{\|\mathbf{u}(n)\|^2} [d^*(n) - \mathbf{w}^T(n) \mathbf{u}^*(n)] \quad (3.32)$$

Kestirim hatası $e(n)$ 'nin tanımı olan;

$$e(n) = d(n) - \mathbf{w}^H(n) \mathbf{u}(n) \quad (3.33)$$

eşitliğinden (3.32) eşitliği

$$\lambda^* = \frac{2}{\|\mathbf{u}(n)\|^2} e^*(n) \quad (3.34)$$

şeklinde sadeleştirilebilir. (3.29) eşitliğinin içine (3.34) yerleştirilerek elde edilen değişimin karesel normu olan;

$$\begin{aligned} \delta \mathbf{w}_k(n+1) &= \mathbf{w}_k(n+1) - \mathbf{w}_k(n) \\ &= \frac{1}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n-k) e^*(n), \quad k = 0, 1, \dots, M-1 \end{aligned} \quad (3.35)$$

eşitliği vektör formunda yazılırsa;

$$\delta \mathbf{w}(n+1) = \mathbf{w}(n+1) - \mathbf{w}(n) = \frac{1}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) e^*(n) \quad (3.36)$$

ifadesi elde edilir.

Benzer şekilde süzgeç katsayılarının güncellenmesi,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) e^*(n) \quad (3.37)$$

eşitliğiyle sağlanmış olunur (Haykin, 1996).

3.3 Özyineli En Küçük Kareler (RLS) Algoritması

Uyarlanır çapraz süzgeçlerin tasarımı için özyineli algoritmaları geliştirmekte en küçük kareler yöntemleri kullanılır. $n-1$ özyinelemede süzgeç katsayı vektörünün en küçük karesel kestirimi verilmişse, yeni veriye ulaşmada, n özyinelemede bu vektörün güncellenmiş kestirimi hesaplanır. Bu çözümü RLS algoritması ile kullanarak sağlanır.

RLS algoritmasını geliştirmek için, en küçük karesel yöntemlerine ait bazı basit ilişkiler gözden geçirilmelidir. Daha sonra, matris evirme ön kuramı olarak bilinen matris cebirindeki ilişkiden faydalanılarak RLS algoritmasını geliştirilmiştir. RLS algoritmasının önemli bir özelliği, algoritma başlatıldığında zamanın geriye genişletilerek, giriş verisindeki bilgiden yararlanabilmesidir. Yakınsaması, basit LMS algoritmasından daha hızlıdır. Ancak performanstaki gelişimine bakıldığında, hesaplama karmaşıklığında oldukça büyük artış vardır (Haykin, 1996).

En küçük karesel yöntemin özyineli gerçekleştirmesinde, bilinen giriş koşullarıyla hesaplamaya başlanır ve eski kestirimlerin güncellenmesi içinde yeni veri örneklerindeki bilgileri kullanılır. Buna bağlı olarak, n 'nin gözlenebilir bir bilginin değişken uzunluğu olduğu yerde, maliyet fonksiyonu $\delta(N)$ 'ni küçültmeye çalışılır.

Maliyet fonksiyonu;

$$\delta(N) = \sum_{n=1}^N \beta(N, n) |e(n)|^2 \quad (3.38)$$

şeklinde tanımlanır. Yukarıdaki eşitlikte, $e(n)$ hata işareti, istenen cevap $d(n)$ ve

$u(n), u(n-1), \dots, u(n-M+1)$, girişlerinin çapraz süzgeçleri tarafından üretilen çıkış işareti $y(n)$ arasındaki farktır ve;

$$e(n) = d(n) - y(n) = d(n) - \mathbf{w}^H(n)\mathbf{u}(n) \quad (3.39)$$

olarak ifade edilir. n anında, $\mathbf{u}(n)$ giriş vektörü;

$$\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-M+1)]^T \quad (3.40)$$

şeklinde tanımlanır.

n anında, $\mathbf{w}(n)$ süzgeç katsayı vektörü;

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T \quad (3.41)$$

biçiminde ifade edilir.

Ağırlıklandırma faktörü $\beta(N, n)$ (3.38) eşitliğinde;

$$0 < \beta(N, n) \leq 1, \quad i = 1, 2, \dots, n \quad (3.42)$$

aralığında tanımlıdır.

Ağırlıklandırma faktörünün kullanımında, genelde, süzgeçleme işlemi durağan olmayan ortam olduğu zaman, gözlemlenebilir verinin izlenen istatistiksel değişimlerinin olasılığını kayıpsız bir şekilde sağlamak amacıyla, “unutma” uzak geçmişteki verinin temini amaçlanmıştır. Ağırlıklandırmanın özel biçimi, genelde “unutma faktörü” aşağıdaki gibi kullanılır:

$$\mathcal{J}(N) = \sum_{n=1}^N \lambda^{N-n} |e(n)|^2. \quad (3.43)$$

Süzgeç katsayı vektörü $\mathbf{w}(n)$ ’nin en iyi değerine, (3.43) eşitliğindeki $\mathcal{J}(n)$ maliyet fonksiyonunun en küçük değerine erişmek için, normal denklem ile tanımlanarak;

$$\mathbf{\Phi}(n)\mathbf{w}(n) = \mathbf{z}(n) \quad (3.44)$$

şeklinde yazılır.

$M \times M$ boyutlu ilinti matrisi $\mathbf{\Phi}(n)$;

$$\Phi(n) = \sum_{n=1}^N \lambda^{N-n} \mathbf{u}(n) \mathbf{u}^H(n) \quad (3.45)$$

olarak ifade edilir.

* işaretin karmaşık eşleniğini temsil ettiği durumda, istenilen cevap ve çapraz süzgecin girişleri arasındaki $M \times I$ boyutlu çapraz ilinti matrisi $\mathbf{z}(n)$;

$$\mathbf{z}(n) = \sum_{n=1}^N \lambda^{N-n} \mathbf{u}(n) \mathbf{d}^*(n) \quad (3.46)$$

şeklinde tanımlanır.

(3.45)'deki ilinti matrisi $\Phi(n)$, $\Phi = \mathbf{A}^H \mathbf{A}$ eşitliğinin zaman ortalamalı çeşidinden iki noktada farklıdır:

1. (3.45) eşitliğinin maliyet fonksiyonu gibi kabul edilmesinden ortaya çıkan $\Phi = \mathbf{A}^H \mathbf{A}$ eşitliğinin sağ tarafındaki toplam içindeki matris ürünü $\mathbf{u}(n) \mathbf{u}^H(n)$, üstel ağırlık faktörü λ^{N-n} tarafından ağırlıklandırılır.
2. Ön pencerelemenin kullanımı, giriş verisi $n = 1$ anından evvel sıfıra eşitliğine bağlıdır ve $n = 1$ kullanımı, toplamın alt sınırı gibidir.

(3.45) eşitliğinin sağ tarafındaki toplamından $n = N$ 'e ilişkin terimi ayrılırsa

$$\Phi(N) = \lambda \left[\sum_{n=1}^{N-1} \lambda^{N-1-n} \mathbf{u}(n) \mathbf{u}^H(n) \right] + \mathbf{u}(N) \mathbf{u}^H(N) \quad (3.47)$$

ifadesi yazılır.

(3.47) eşitliğinin sağ tarafının ilk ifadesinin yerine $\Phi(n-1)$ ilinti matrisini konulursa ve bu eşitlik yeniden yazılırsa;

$$\Phi(n) = \lambda \Phi(n-1) + \mathbf{u}(n) \mathbf{u}^H(n) \quad (3.48)$$

şekline dönüşür. Burada $\Phi(n-1)$, ilinti matrisinin önceki değeridir ve matris ürünü $\mathbf{u}(n) \mathbf{u}^H(n)$ ise güncelleme işleminde düzeltme terimi rolünü oynar.

Benzer şekilde (3.46) eşitliği düzenlenirse;

$$\mathbf{z}(n) = \lambda \mathbf{z}(n-1) + \mathbf{u}(n) \mathbf{d}^*(n) \quad (3.49)$$

eşitliği elde edilir.

(3.44) eşitliğine göre süzgeç katsayı vektörü için en küçük-karesel kestirim $\mathbf{w}(n)$ 'nin hesaplanması için $\Phi(n)$ ilinti matrisinin tersi belirlenmelidir. Eğer süzgeç katsayılarının sayısı M büyükse, belirleme işlemi oldukça zaman alacaktır. Üstelik $n=1, 2, \dots, \infty$ için süzgeç katsayı vektörünün özyinelemesi için en küçük-karesel kestirim $\mathbf{w}(n)$ 'de hesaplamak gerekmektedir. Bu hesaplamalar, matris evirme olarak bilinen basit bir sonucu kullanılarak yapılabilir.

3.3.1 Matris Evirme

$\mathbf{A}$ ve $\mathbf{B}$ matrisleri $M \times M$ boyutlu ve pozitif olmak üzere;

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C} \mathbf{D}^{-1} \mathbf{C}^H \quad (3.50)$$

şeklinde ilişkili olsun. Yukarıdaki eşitlikte $\mathbf{C}$, $M \times N$ boyutlu matrisi, $\mathbf{D}$ ise $N \times M$ boyutlu pozitif matrisi ifade eder. Matris evirmeye bağlı olarak, $\mathbf{A}$ matrisinin tersi;

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B} \mathbf{C} (\mathbf{D} + \mathbf{C}^H \mathbf{B} \mathbf{C})^{-1} \mathbf{C}^H \mathbf{B} \quad (3.51)$$

biçiminde açılabilir.

Yukarıdaki eşitlikler dikkate alındığında, eğer (3.50) eşitliğindeki gibi bir $\mathbf{A}$ matrisi varsa, onun tersi (3.51)'deki gibi alınabilir ve literatürde matris evirme olarak bilinir (Haykin, 1996).

3.3.2 Üstel Ağırlıklanan RLS Algoritması

İlinti matrisi $\Phi(n)$ 'nin pozitif tanımlandığından emin olarak matris evirme özyineli (3.48) eşitliğine eklenebilir. Öncelikle aşağıdaki

$$\mathbf{A} = \Phi(n) \quad \mathbf{B}^{-1} = \lambda \Phi(n-1) \quad \mathbf{C} = \mathbf{u}(n) \quad \mathbf{D} = 1 \quad (3.52)$$

ifadeleri (3.51) eşitliğinde yerlerine konulursa, ilinti matrisinin tersi için;

$$\Phi^{-1}(n) = \lambda^{-1} \Phi^{-1}(n-1) - \frac{\lambda^{-2} \Phi^{-1}(n-1) \mathbf{u}(n) \mathbf{u}^H(n) \Phi^{-1}(n-1)}{1 + \lambda^{-1} \mathbf{u}^H(n) \Phi^{-1}(n-1) \mathbf{u}(n)} \quad (3.53)$$

özyineli eşitliği elde edilir.

Hesaplamayı kolaylaştırmak için;

$$\mathbf{P}(n) = \Phi^{-1}(n) \quad (3.54)$$

ve

$$\mathbf{k}(n) = \frac{\lambda^{-1} \mathbf{P}(n-1) \mathbf{u}(n)}{1 + \lambda^{-1} \mathbf{u}^H(n) \mathbf{P}(n-1) \mathbf{u}(n)} \quad (3.55)$$

eşitlikleri yazılabilir.

Bunları kullanarak (3.53) eşitliği;

$$\mathbf{P}(n) = \lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}^H(n) \mathbf{P}(n-1) \quad (3.56)$$

şeklinde yeniden yazılır.

$M \times M$ boyutlu $\mathbf{P}(n)$ matrisi, ters ilinti matrisidir. $M \times 1$ boyutlu $\mathbf{k}(n)$ vektörü ise kazanç vektörüdür. (3.56) eşitliği ise RLS algoritması için Riccati eşitliği olarak adlandırılır.

(3.55)'in yeniden düzenlemesi ile;

$$\begin{aligned} \mathbf{k}(n) &= \lambda^{-1} \mathbf{P}(n-1) \mathbf{u}(n) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}^H(n) \mathbf{P}(n-1) \mathbf{u}(n) \\ &= [\lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}^H(n) \mathbf{P}(n-1)] \mathbf{u}(n) \end{aligned} \quad (3.57)$$

eşitliği elde edilir.

(3.57) eşitliğinin sağ tarafındaki köşeli parantez içindeki ifadeye bakıldığında, (3.56) eşitliğinde tanımlanan $\mathbf{P}(n)$ 'e eşit olduğunu görülür. Bu doğrultuda (3.57);

$$\mathbf{k}(n) = \mathbf{P}(n) \mathbf{u}(n) \quad (3.58)$$

şeklinde basitleştirebilir.

Bu sonuç $\mathbf{P}(n) = \Phi^{-1}(n)$ ile de düzenlenirse, kazanç faktörü

$$\mathbf{k}(n) = \Phi^{-1} \mathbf{u}(n) \quad (3.59)$$

olarak yazılır. Diğer bir deyişle kazanç faktörü $\mathbf{k}(n)$, giriş vektörü $\mathbf{u}(n)$ 'nin ilinti matrisi $\Phi(n)$ 'nin tersi ile çarpımıdır.

3.3.3 Süzgeç Katsayı Vektörü için Zaman Güncellemesi

Bu bölümde süzgeç katsayı vektörü için en küçük karesel kestirim $\mathbf{w}(n)$ 'nin güncellemesi için özyineli eşitliği ifade edilmektedir. Bunu yapmak için, (3.44), (3.49) ve (3.53) eşitlikleri kullanılarak n özyinelemede süzgeç katsayı vektörü için en küçük karesel kestirim $\mathbf{w}(n)$;

$$\begin{aligned}\mathbf{w}(n) &= \Phi^{-1}(n)\mathbf{z}(n) \\ &= \mathbf{P}(n)\mathbf{z}(n) \\ &= \lambda\mathbf{P}(n)\mathbf{z}(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n)\end{aligned}\tag{3.60}$$

şeklinde açılır.

(3.56)'deki $\mathbf{P}(n)$ ifadesini kullanılarak, (3.60) eşitliğinin sağ tarafı yeniden düzenlenir:

$$\begin{aligned}\mathbf{w}(n) &= \mathbf{P}(n-1)\mathbf{z}(n-1) - k(n)\mathbf{u}^H(n)\mathbf{P}(n-1)\mathbf{z}(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n) \\ &= \Phi^{-1}(n-1)\mathbf{z}(n-1) - k(n)\mathbf{u}^H(n)\Phi^{-1}(n-1)\mathbf{z}(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n) \\ &= \mathbf{w}(n-1) - k(n)\mathbf{u}^H(n)\mathbf{w}(n-1) + \mathbf{P}(n)\mathbf{u}(n)d^*(n)\end{aligned}\tag{3.61}$$

Son olarak, (3.60) eşitliğindeki kazanç faktörü $\mathbf{k}(n)$ ifadesi yerine konulduğunda (3.61) eşitliği yeniden yazılarak;

$$\begin{aligned}\mathbf{w}(n) &= \mathbf{w}(n-1) + \mathbf{k}(n)[d^*(n) - \mathbf{u}^H(n)\mathbf{w}(n-1)] \\ &= \mathbf{w}(n-1) + \mathbf{k}(n)\xi^*(n)\end{aligned}\tag{3.62}$$

şekline dönüşür. Bu ifadedeki $\xi(n)$, önsel kestirim hatasıdır ve;

$$\begin{aligned}\xi(n) &= d(n) - \mathbf{u}^T\mathbf{w}^*(n-1) \\ &= d(n) - \mathbf{w}^H(n-1)\mathbf{u}(n)\end{aligned}\tag{3.63}$$

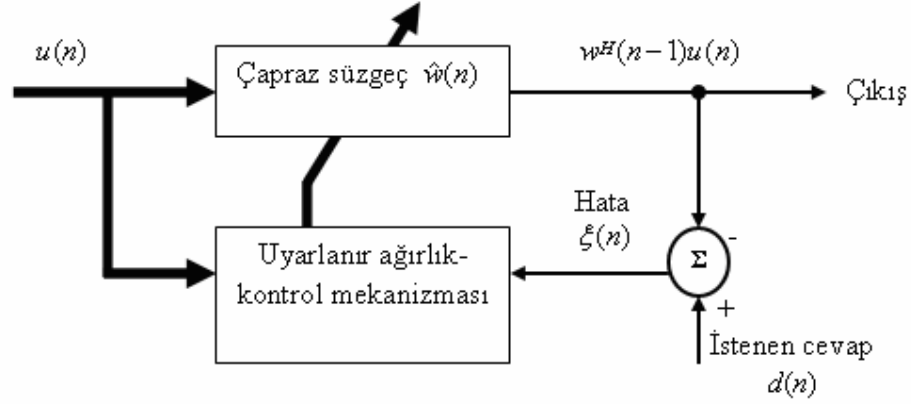
şeklinde tanımlanır.

RLS algoritması için süzgeç katsayı vektörü ayarlamasında önsel kestirim hatasının blok diyagramı Şekil 3.3'te verilmiştir.

Sonsal hata hesaplandığında ise;

$$e(n) = d(n) - \mathbf{w}^H(n)\mathbf{u}(n)\tag{3.64}$$

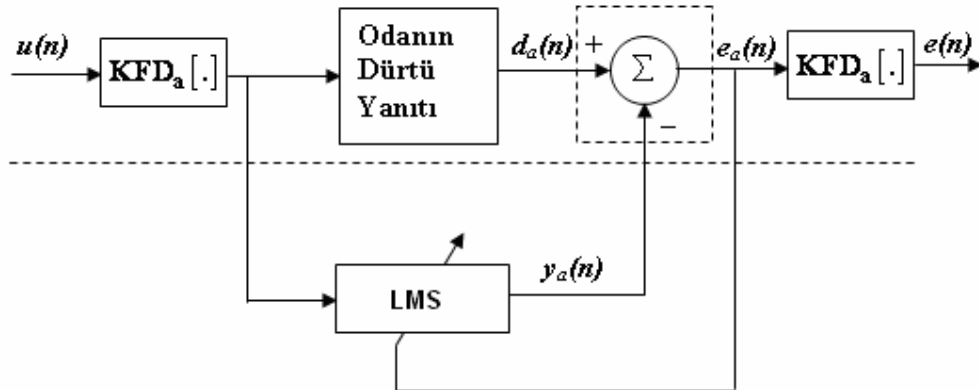
eşitliği elde edilir (Haykin, 1996).



Şekil 3.3 RLS Süzgeç Yapısı

3.4 Kesirli Fourier Bölgesinde LMS Algoritması

Uyarlamalı süzgeçlemede LMS tabanlı algoritmalar, zaman ve frekans bölgesinde ara değerlendirme yapan KFD kullanılarak kesirli Fourier bölgesinde uygulanabilir. Bu çalışma kapsamında akustik yankı giderimi için önerilen KFD bölgesinde LMS algoritması yapısı, Şekil 3.4'te görüldüğü gibi oluşturulmuştur. KFD bölgesinde LMS algoritmasında, giriş işareti $u(n)$ ve istenilen işaret $d(n)$ 'nin süzgeçleme işleminden önce KFD dönüşümleri alınır. Daha sonra KFD bölgesinde uyarlanır süzgeçleme LMS algoritması uygulanır. Hata işareti $e(n)$ ve süzgeç çıkışı $y(n)$ 'nin ters KFD'leri alınarak süzgeçleme işlemi bitirilir.



Şekil 3.4 KFD Bölgesi LMS algoritmasının yapısı

Giriş işareti $\mathbf{u}(n)$ 'nin “ a .” dereceden KFD'si;

$$\mathbf{u}(n) = \begin{bmatrix} u(n) \\ u(n-1) \\ \vdots \\ u(n-p) \end{bmatrix} \xrightarrow{FrFT} \mathbf{u}_a(n) = \begin{bmatrix} u_a(n) \\ u_a(n-1) \\ \vdots \\ u_a(n-p) \end{bmatrix} \quad (3.65)$$

olarak alınır. Akustik yankı işareti $d(n)$ 'ni modellemek için KFD bölgesinde LMS algoritmasına giriş işareti $\mathbf{u}_a(n)$ sokulur. Süzgeç çıkışı $y(n)$ ve istenilen işaret $d(n)$ arasındaki hata işareti $e(n)$ bulunur ve bu doğrultuda KFD bölgesinde LMS süzgeç katsayıları güncellenir.

Özetle KFD bölgesinde LMS algoritmasını uyarlamak için aşağıdaki bu üç basamağa ihtiyaç vardır;

1. Sonlu Dürtü Yanıtı (FIR) süzgeç çıkışı, $y_a(n)$ denklemi kullanılarak elde edilir:

$$y_a(n) = \mathbf{w}^H(n) \mathbf{u}_a(n). \quad (3.66)$$

2. KFD bölgesindeki kestirim hatası olan $e_a(n)$;

$$e_a(n) = d_a(n) - y_a(n) \quad (3.67)$$

şeklinde tanımlanır.

3. Süzgeç-katsayıları güncellemesi ise;

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{u}_a^*(n) e_a(n) \quad (3.68)$$

şeklinde ifade edilir.

Bu tezde akustik yankı giderimi için uyarlanır süzgeçleme algoritmalarından LMS, NLMS, RLS ve KFD bölgesinde LMS kullanımı ile sağlanmıştır. İstenilen yankı işareti $d(n)$, uyarlanır süzgeçleme algoritmalarıyla kestirilmiştir. Böylece mikrofon işaretinden yankı işareti süzölmüş ve uzak-uç bölgesine sadece yakın-uç işaretinin iletimi sağlanmıştır. Kullanılan bu algoritmaların yakınsaması, gerçek zamanlı haberleşmeye uygunluğu ve işlemsel kolaylığı karşılaştırılmıştır.

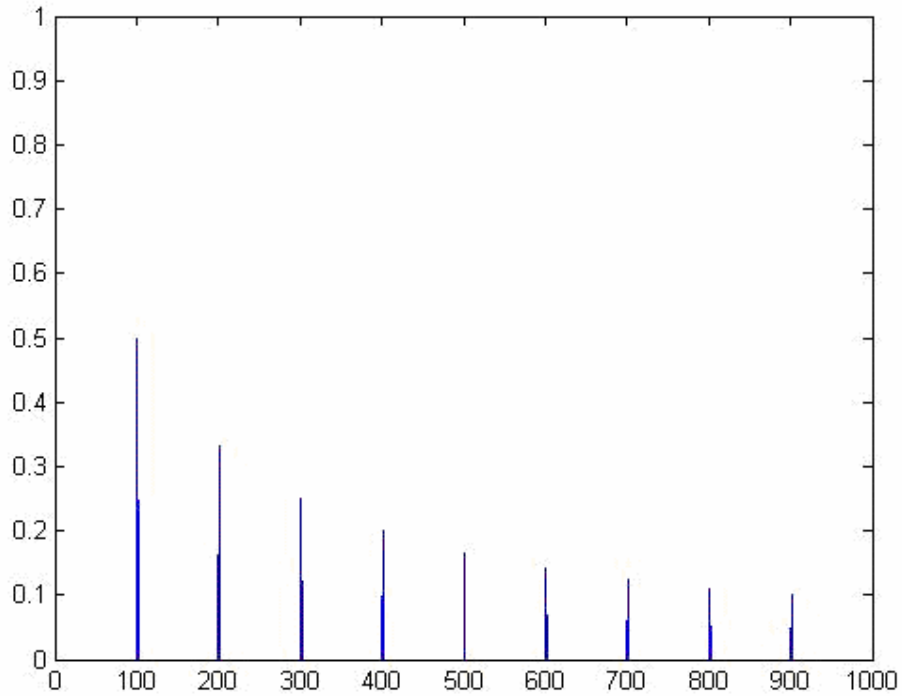
Bu tezde, zaman bölgesinde LMS, NLMS, RLS ve farklı derecelerde KFD bölgesi LMS algoritmaları kullanılarak akustik yankı yok edilmektedir. Algoritmalarının performansları yankı giderimi performans ölçüm kriteri olan Yankı Geri Dönüş Kayıp Artışı (ERLE);

$$ERLE_{dB}(n) = 10 \log \left[\frac{\text{referans (yankı) işaretinin gücü}}{\text{hata işaretinin gücü}} \right] \\ = 10 \log \left[\frac{E[d^2(n)]}{E[e^2(n)]} \right] \quad (4.1)$$

eşitliği ile değerlendirilmiştir (Gordy ve Goubran, 2005).

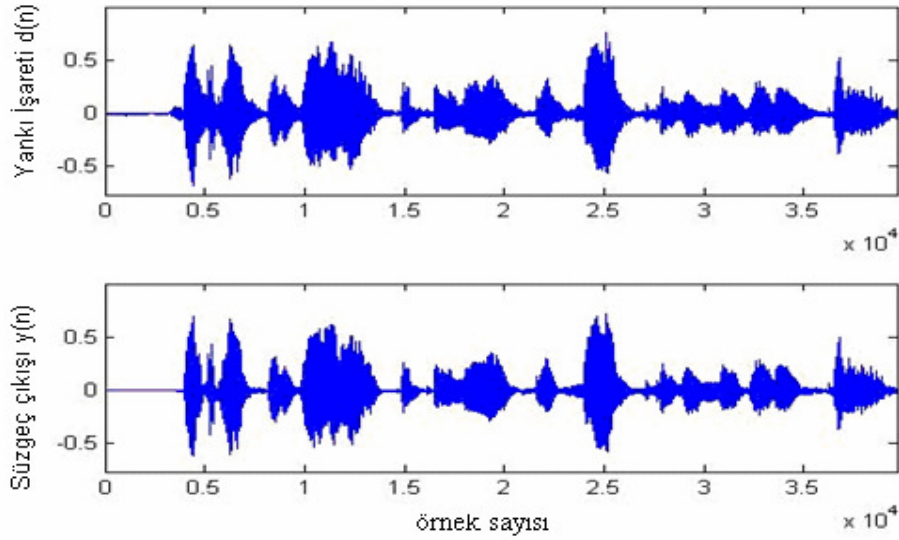
4.1 LMS Algoritması

Zaman bölgesi LMS algoritması kullanılarak akustik yankıyı yok etme uygulamasında uzak-uç işareti $\mathbf{u}(n)$ 'nin 40.000 örnek olarak alındığı bir benzetim yapılmaktadır. Uzak uç-ışaretinin akustik yankısını oluşturan odanın dürtü yanıtı Şekil 4.2'de gösterildiği gibi alınmaktadır.



Şekil 4.2 Uzak uç bölgesi akustik yankıyı üreten odanın dürtü yanıtı

Uygulanan zaman bölgesi LMS algoritması, süzgeç katsayılarının sayısının $l=128$ olması durumunda en iyi performansı adım büyüklüğü $\mu = 0,2$ değerinde vermiştir. Bu değerler için Şekil 4.3’de gösterildiği gibi akustik yankı işareti $d(n)$ ’nin, uyarlanır süzgeçleme ile kestirimi yapılmıştır. Bu kestirim esnasında oluşan anlık karesel hata Şekil 4.4’te gösterilmektedir.



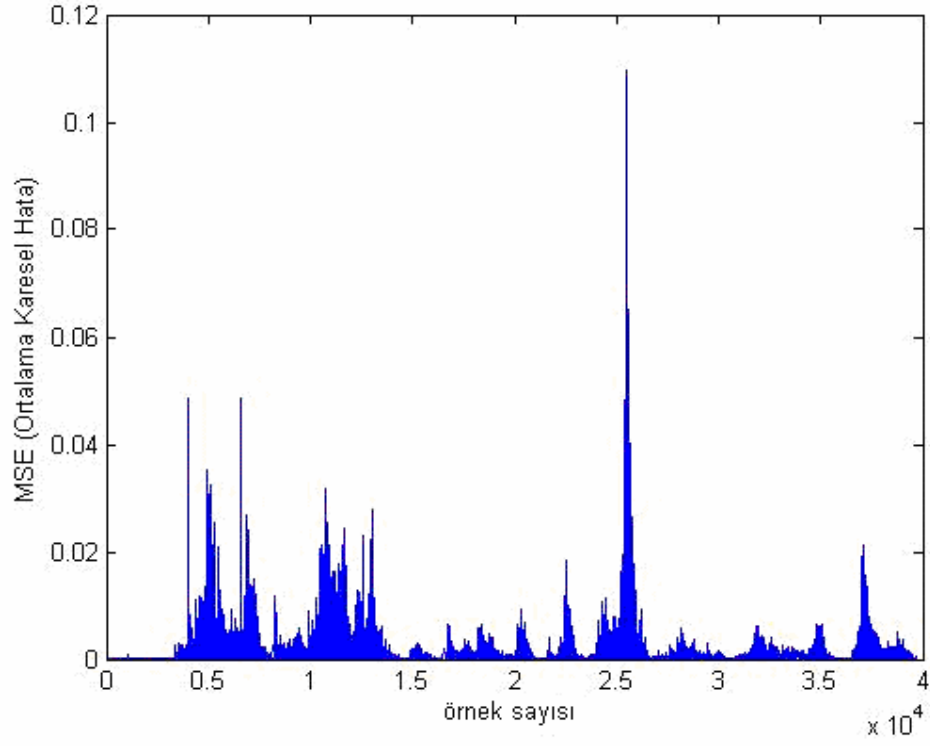
Şekil 4.3 Yankı işaretinin LMS algoritması kullanılarak elde edilen kestirimi

Yankı giderimi performansı ölçüm kriteri olan ERLE, 11,67 dB değeri olarak elde edilmektedir ve süzgeçleme süresince dağılımı Şekil 4.5’te gösterilmektedir.

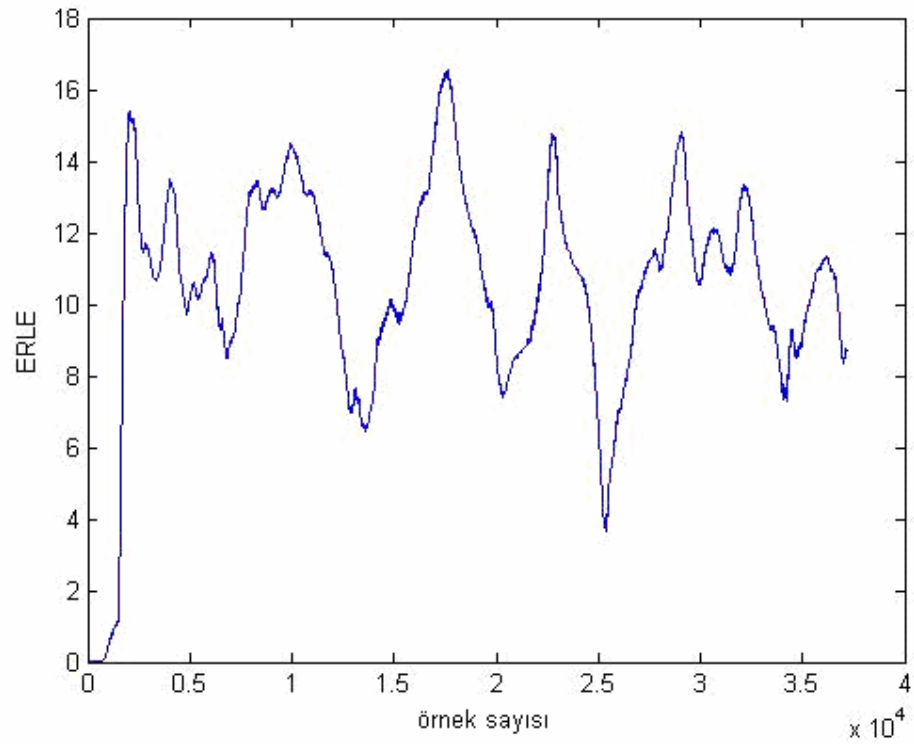
LMS algoritmasında eğer süzgeç katsayılarının sayısı $l=256$ ve $\mu=0,1$ adım büyüklüğü seçilirse, akustik yankıyı yok etme performansı ERLE artar ve 12,72 dB değerine yükselir. Süzgeç katsayılarının sayısı $l=128$ iken, adım büyüklüğü μ parametresine göre bulunan ERLE değerleri Çizelge 4.1’den de anlaşıldığı gibi adım büyüklüğü μ parametresinin LMS algoritması için önemli değişkendir.

Çizelge 4.1 LMS algoritması için ERLE değerinin μ parametresine göre değişimi

Adım Büyüklüğü μ	0,05	0,1	0,15	0,2
ERLE (dB)	9,58	10,65	11,29	11,67



Şekil 4.4 LMS anlık karesel hatasının 40.000 örnekte dağılımı

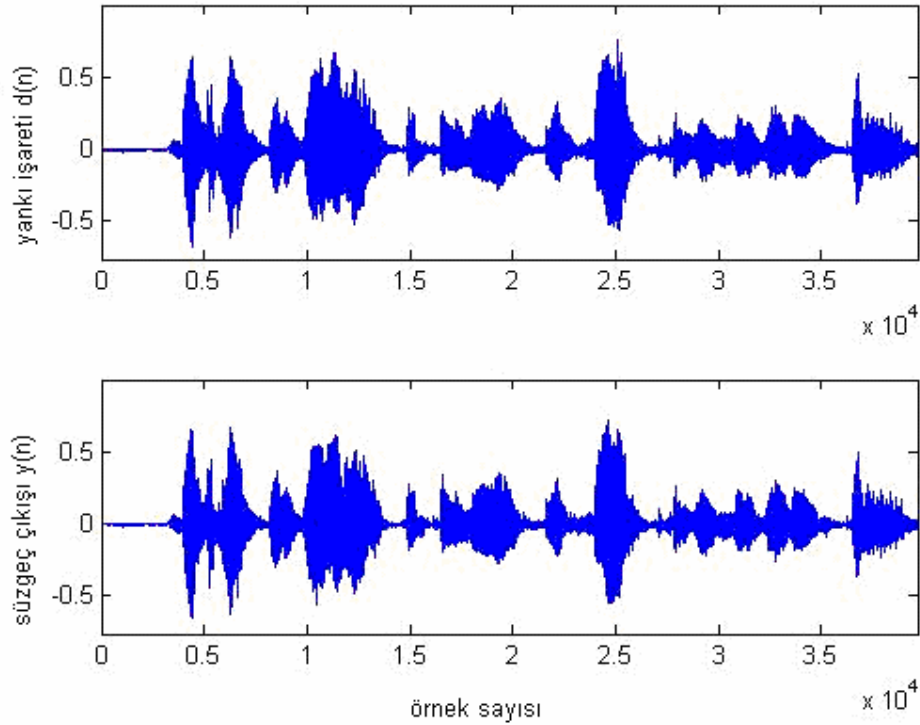


Şekil 4.5 LMS Algoritmasında 40.000 örnek için 11,67 dB ortalamalı ERLE dağılımı

Böylece LMS algoritmasının performansını süzgeç katsayısı l ve adım büyüklüğü μ değerlerinin etkilediği de gözlemlenmiştir.

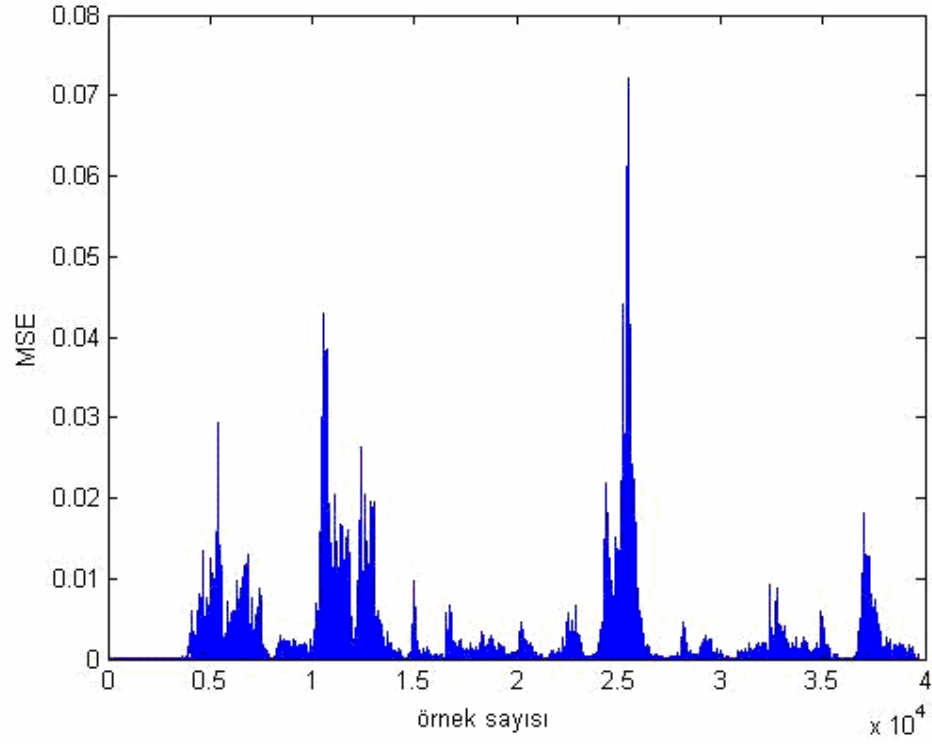
4.2 NLMS Algoritması

LMS algoritmasının normalize edilmesinden türetilen zaman bölgesi NLMS algoritması kullanılarak akustik yankıyı yok etme uygulamasında uzak-uç işareti $u(n)$ yine 40.000 örnek olarak alındığı bir benzetim yapılmaktadır. Uygulanan zaman bölgesi NLMS algoritması, süzgeç katsayılarının sayısı $l=128$ olması durumunda en iyi performansı adım büyüklüğü $\mu=0,3$ değerinde vermiştir. Bu değerler için Şekil 4.6'te gösterildiği gibi akustik yankı işareti $d(n)$ 'nin, uyarlanır süzgeçleme ile kestirimini yapılmıştır. Bu kestirim esnasında oluşan karesel hata NLMS MSE, Şekil 4.7'te gösterilmektedir.

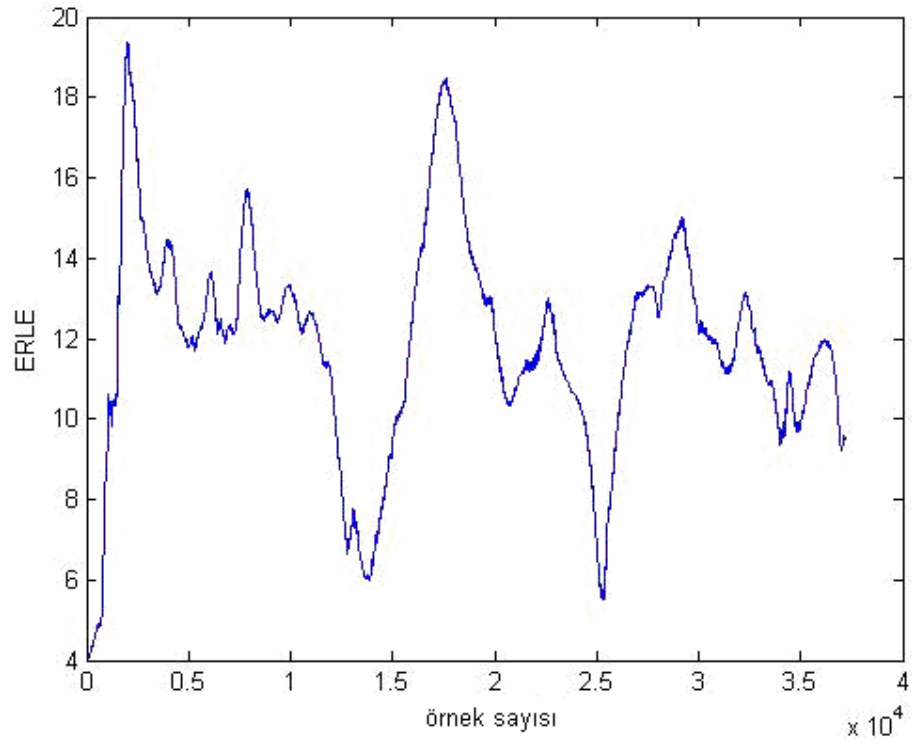


Şekil 4.6 Yankı işaretinin NLMS algoritması kullanılarak elde edilen kestirimi

Yankı giderimi performansı ölçüm kriteri olan ERLE, 12,30 dB değeri olarak elde edilmektedir ve süzgeçleme süresince dağılımı Şekil 4.8'de gösterilmektedir.



Şekil 4.7 NLMS karesel hatası NLMS MSE'nin 40.000 örnekte dağılımı



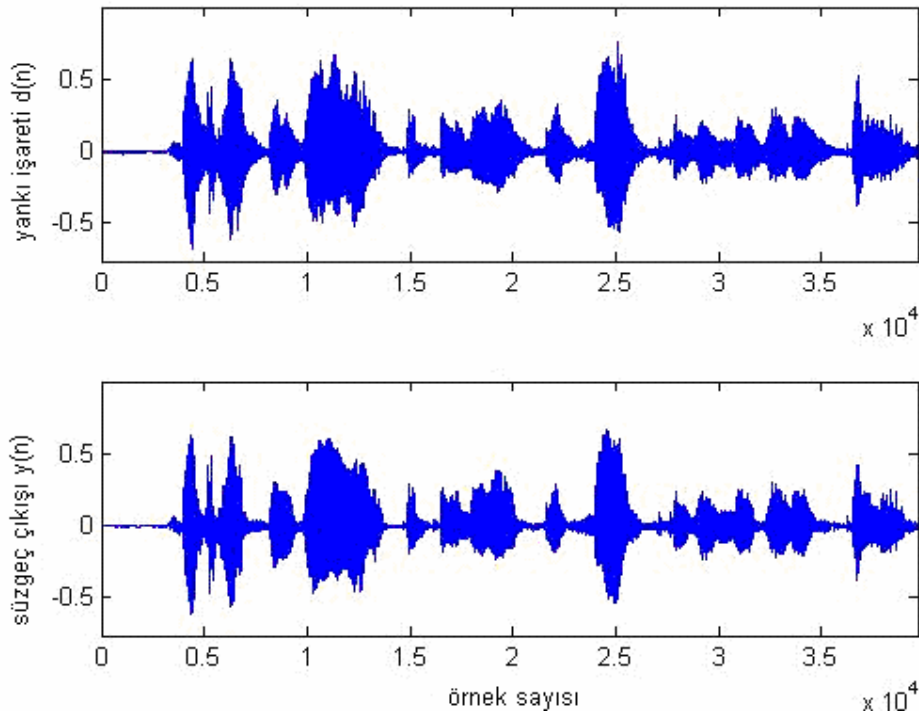
Şekil 4.8 NLMS Algoritmasında 40.000 örnek için 12,30 dB ortalamalı ERLE dağılımı

NLMS algoritmasında eğer süzgeç katsayı sayısı $l = 256$ ve $\mu = 0,3$ adım büyüklüğü seçilirse, akustik yankıyı yok etme performansı ERLE artar ve 13,47 dB değerine yükselir. Böylece NLMS algoritmasının performansını süzgeç katsayısı sayısı l ve adım büyüklüğü μ değerlerinin etkilediği de gözlemlenmiştir.

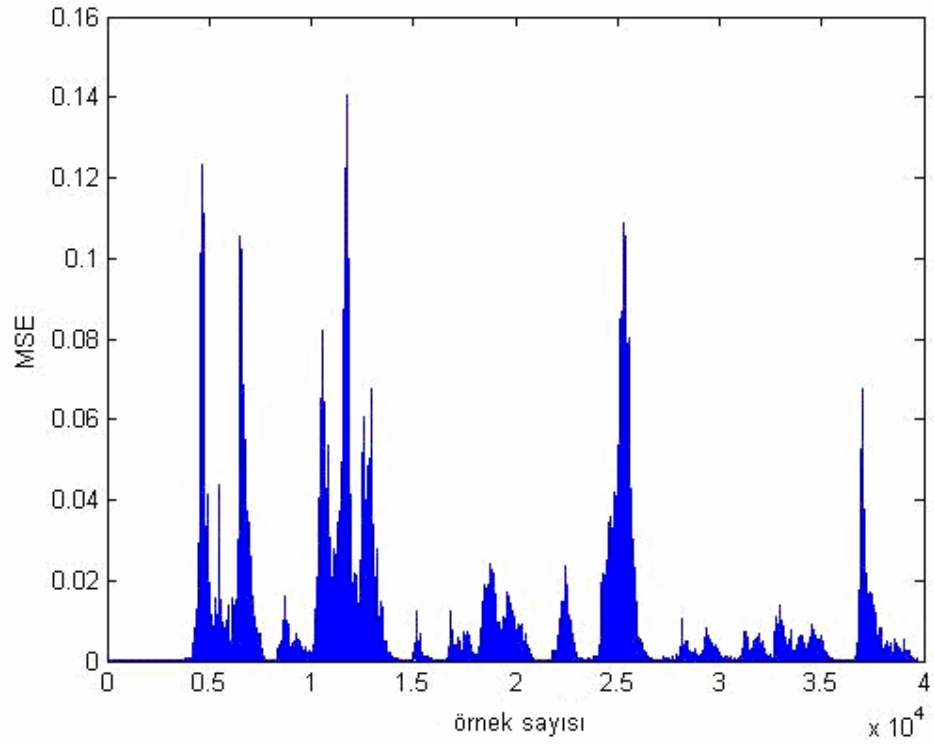
Zaman bölgesi NLMS algoritması ile elde edilen yankı performans kriteri ERLE ve ortalama karesel hata MSE sonuçları, zaman bölgesi LMS algoritması sonuçları ile karşılaştırıldığında, ERLE değerinde kazanç ve MSE'nin tepe değerinde düşüş sağlanmaktadır. Böylece zaman bölgesi LMS'e göre akustik yankı giderimi daha iyi yapılabileceği sonucuna ulaşılmaktadır.

4.3 RLS Algoritması

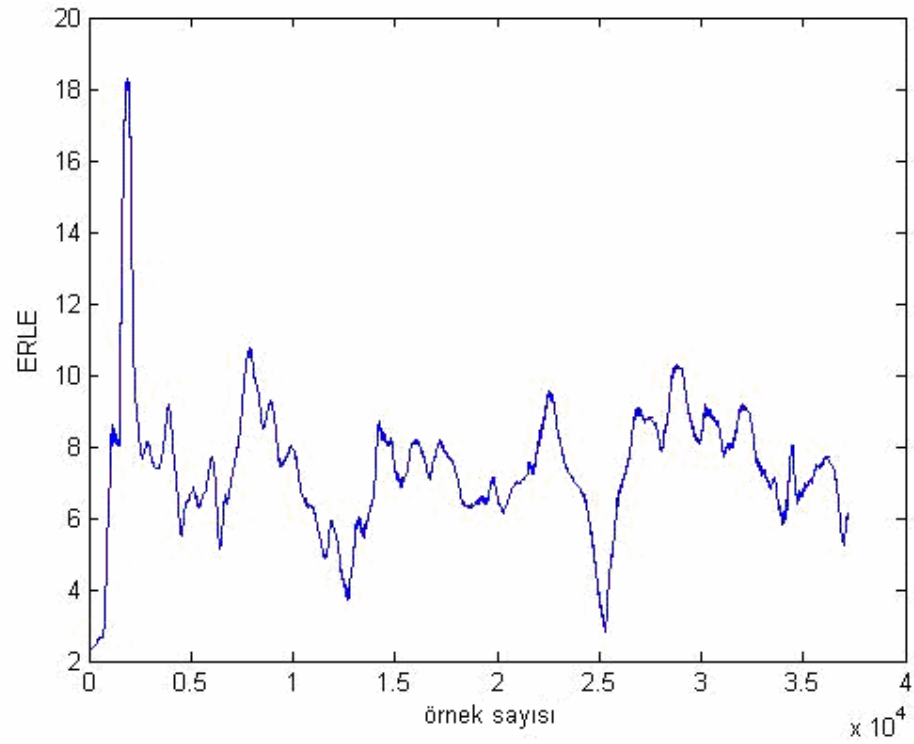
Zaman bölgesi en küçük kareler yöntemini kullanan RLS algoritması kullanılarak uzak-uç işareti yine $u(n)$ 'nin 40.000 örnek olarak alındığı bir benzetim yapılmaktadır. Uygulanan zaman bölgesi RLS algoritması, süzgeç katsayılarının sayısı $l = 128$ olması durumunda en iyi performansı üstel ağırlık faktörü $\lambda = 1$ değerinde vermiştir. Bu değerler için Şekil 4.9'da gösterildiği gibi akustik yankı işareti $d(n)$ 'nin, uyarlanır süzgeçleme ile kestirimi yapılmıştır. Bu kestirim esnasında oluşan karesel hata RLS MSE, Şekil 4.10'da gösterilmektedir.



Şekil 4.9 Yankı işaretinin RLS algoritması kullanılarak elde edilen kestirimi



Şekil 4.10 RLS karesel hatası RLS MSE'nin 40.000 örnekte dağılımı

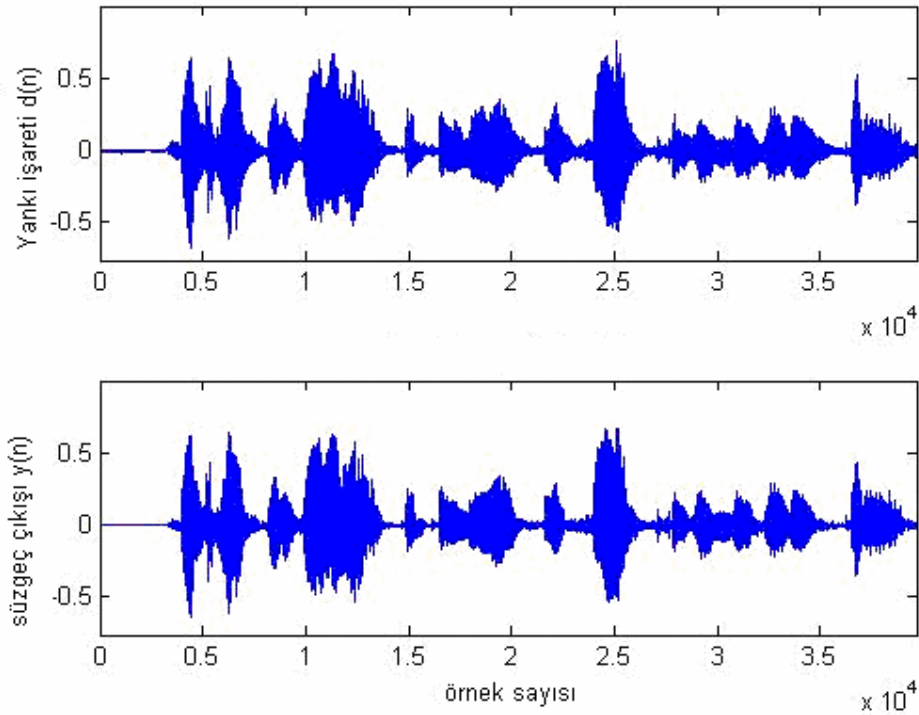


Şekil 4.11 RLS Algoritmasında 40.000 örnek için 7,15 dB ortalamalı ERLE dağılımı

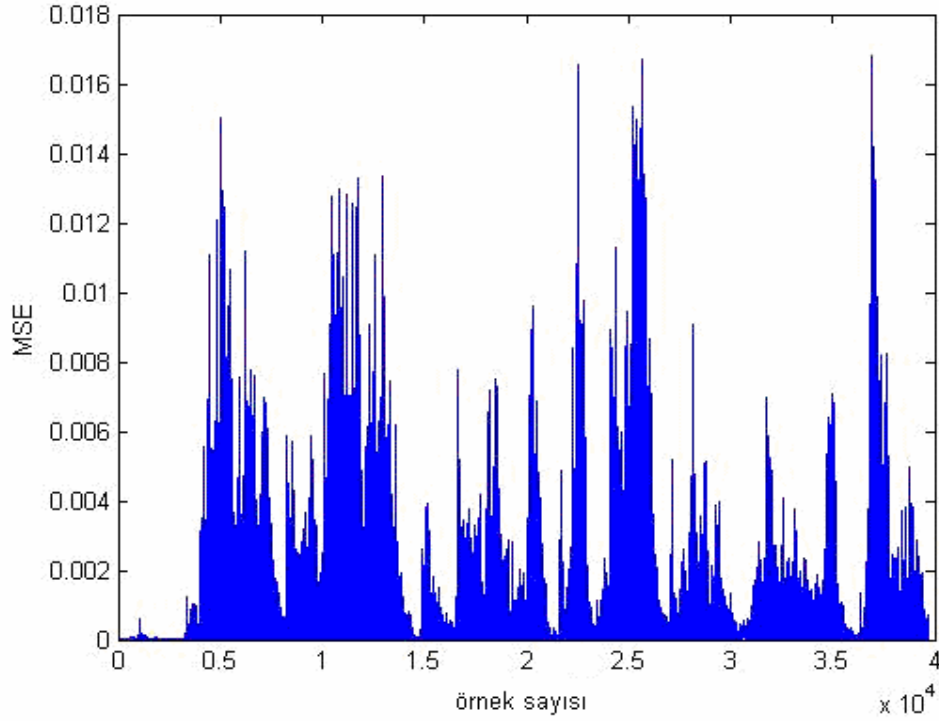
Yankı giderimi performansı ölçüm kriteri olan ERLE 7,15 dB değeri olarak elde edilmektedir ve süzgeçleme süresince dağılımı Şekil 4.11’de gösterilmektedir. RLS algoritmasında eğer süzgeç katsayı sayısı $l = 256$ seçilirse, akustik yankıyı yok etme performansı ERLE artar ve 9,04 dB değerine yükselir. Böylece RLS algoritmasının performansını süzgeç katsayısı l değerinin etkilediği de gözlemlenmiştir.

4.4 KFD Bölgesi LMS

KFD bölgesi LMS algoritması kullanılarak akustik yankıyı yok etme uygulamasında uzak-uç işareti $u(n)$ ’nin 40.000 örnek olarak alındığı bir benzetim yapılmaktadır. Uygulanan KFD bölgesi LMS algoritması, süzgeç katsayılarının sayısı $l = 128$ olması durumunda en iyi performansı, adım büyüklüğü $\mu = 0,3$, $a = 0,2$ dereceden KFD alındığı değerinde vermiştir. Bu değerler için Şekil 4.12’de gösterildiği gibi akustik yankı işareti $d(n)$ ’nin, uyarlanır süzgeçleme ile kestirimini yapmıştır. Bu kestirim esnasında oluşan karesel hata KFD bölgesi LMS MSE, Şekil 4.13’te gösterilmektedir.



Şekil 4.12 Yankı işaretinin KFD bölgesi LMS algoritması kullanılarak elde edilen kestirimi

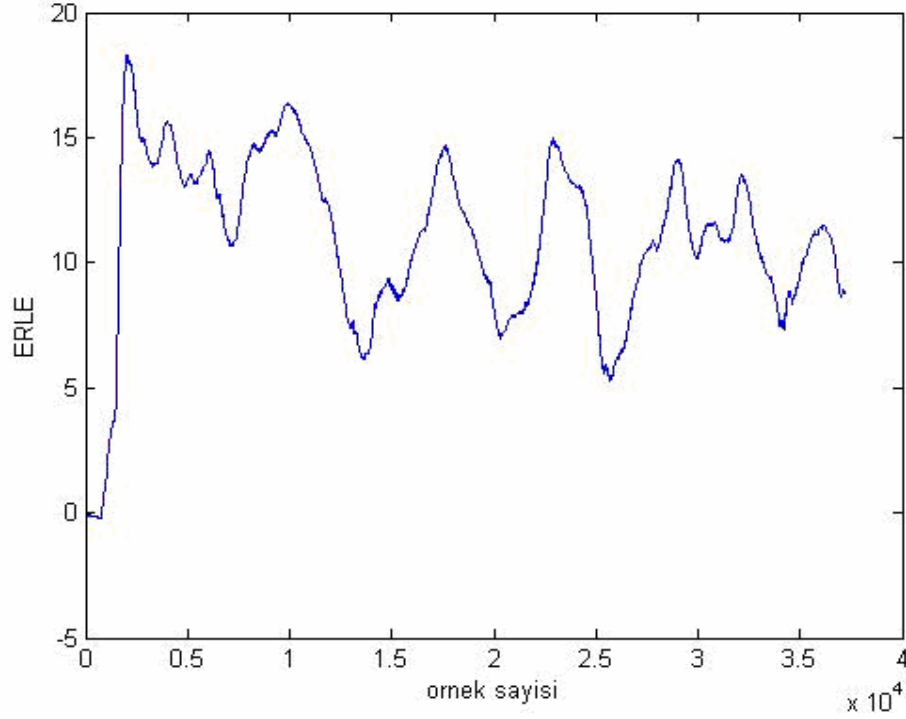


Şekil 4.13 KFD bölgesi LMS karesel hatasının 40.000 örnekte dağılımı

Yankı giderimi performansı ölçüm kriteri olan ERLE, 13,37 dB değeri olarak elde edilmektedir ve süzgeçleme süresince dağılımı Şekil 4.14'te gösterilmektedir. KFD bölgesi LMS algoritması $a=0,2$ derecesinden alınır ve süzgeç katsayılarının sayısı $l=256$ ve $\mu=0,19$ adım büyüklüğü seçilirse, akustik yankıyı yok etme performansı ERLE artar ve 14,03 dB değerine yükselir. Böylece KFD bölgesi LMS algoritmasının performansının süzgeç katsayısı l ve adım büyüklüğü μ değerlerinin etkilediği de gözlemlenmiştir. Adım büyüklüğü μ parametresine göre bulunan ERLE değerleri Çizelge 4.2'den de adım büyüklüğü μ parametresinin KFD bölgesi LMS algoritması için önemli değişkendir.

Çizelge 4.2 KFD bölgesi LMS algoritması için ERLE değerinin μ parametresine göre değişimi

Adım Büyüklüğü μ	0,05	0,1	0,15	0,2	0,25	0,3
a derecesi	0,9	0,8	0,5	0,5	0,2	0,2
ERLE (dB)	10,01	11,14	11,76	12,35	12,89	13,37



Şekil 4.14 KFD bölgesi LMS Algoritmasında 40.000 örnek için 13,37 dB ortalamalı ERLE dağılımı

Yukarıdaki sonuçlardan da görüldüğü gibi KFD bölgesi LMS algoritması zaman bölgesi süzgeçleme algoritmalarından hem ERLE değerinde artışıyla hem de MSE değerinde oldukça iyi bir düşüşle daha iyi performans ile yankıyı yok edebilmektedir.

Ses haberleşmesi gerçek zamanlı bir uygulama olduğundan ve konuşma işaretinin genliği sabit olmadığından optimum yankı gidericinin kısa sürede yakınsamasının yanısıra kendini belli periyotlarda güncellenebilmesi ses iletim kalitesini arttıracaktır. Diğer algoritmalara göre daha yüksek bir ERLE değeriyle yakınsayan KFD bölgesi LMS algoritması, belli periyotlarla veya işaretin anlık frekansı değiştiğinde “ a ” derecesini güncelleyerek konuşma süresince sabit “ a ” derecesiyle değil, o ana en uygun “ a ” derecesiyle yankıyı yok edebilir. Örneğin bir telefon konuşması süresince konuşmacının ses işaretinin genliği değişmektedir. Yüksek ya da alçak tonla konuştuğu zaman işaret değişecektir. Ya da konferans tip haberleşmede kadın konuşmacıların ve erkek konuşmacıların ses tonlarının birbirinde farklı olması nedeniyle onların yankı işaretleri de değişecektir. Dolayısıyla yankı gidericide kesirli Fourier derecesi olan “ a ” parametresi de ortama uyum sağlayarak kendini güncellenmelidir. KFD bölgesi LMS kullanarak yine 40.000 örnekli aynı işareti kullanarak her 20.000 örnek için optimum “ a ” derecesi hesaplandığında Çizelge 4.3’teki sonuçlar elde edilir.

Çizelge 4.3 Akustik yankı gideriminde KFD bölgesi LMS algoritmasının “ a ” derecesinin güncellenmesi

Örnek Aralığı	0-20000	20000-40000
“ a ” derecesi	0.2	0.1
ERLE	13.8 dB	12.8 dB

5. Sonuç

Bu çalışmada IP tabanlı ses iletimi ortamında oluşan akustik yankı giderimi, zaman ve KFD bölgesi uyarlamalı süzgeçleme algoritmaları uygulanarak sağlanmış ve akustik yankıyı gidermedeki performans analizleri yapılmıştır. Zaman bölgesi NLMS algoritması hem LMS hem de RLS algoritmalarına göre yankı geri dönüş kayıp artışı olan ERLE değerini artırarak ve ortalama karesel hata MSE değerini azaltarak iyi bir performans ile akustik yankıyı gidermektedir. Fakat akustik yankı giderimini, önerilen KFD bölgesi LMS algoritması, NLMS algoritması dahil diğer zaman bölgesi uyarlamalı süzgeçleme algoritmalarına oranla, ortalama karesel hata MSE'yi düşürerek ve performans kriteri ERLE değerini 13,37 dB değerine yükselterek sağlamıştır. Adım büyüklüğü μ ve süzgeç katsayıları sayısı l 'ye göre zaman bölgesi LMS ve KFD bölgesi LMS algoritması değişmektedir. Bu parametrelerdeki değişimler sonucunda da yine akustik yankı gideriminde en iyi performansı sağlayan ve ERLE değerini 14,3 dB'e yükselten KFD bölgesi LMS algoritmasıdır. Ayrıca RLS algoritması gibi işlem karmaşıklığı olmadığı için gerçek zamanlı uygulama olan haberleşme sistemlerinde akustik yankı giderimi için kullanımı oldukça avantajlıdır.

IP tabanlı ses iletimi ortamında çoğunlukla konferans tip haberleşmede konuşma işaretinin genliği ortama, konuşmacıya (kadın-erkek) ve zamana bağlı olarak değişmektedir. Akustik yankı gideriminde bu değişimler göz önünde tutulursa elbette yankı işaretine benzetim daha iyi olacağından daha kaliteli bir haberleşme sağlanır. KFD bölgesi LMS algoritmasını kullanarak bu değişimler neticesinde 'a' parametresini güncelleyerek o anki ortama en uygun olan 'a' parametresi ile akustik yankı benzetimi yapılabilir. Haberleşme sistemine uygulandığında herhangi bir değişim algılandığı zaman paralel bir işlem gibi arka planda o değişim sonucunda oluşan yeni akustik yankıya en iyi benzetimi yapacak uyarlamalı süzgeç katsayılarını bulmak için 'a' parametresi güncellenebilir. Böylece konuşma boyunca aynı süzgeç katsayıları kullanılarak yankı yok edilmeyecek, o anki konuşma işaretine dolayısıyla o anki akustik yankı işaretine göre güncellenen süzgeç katsayıları ile yankı yok edilecektir. Bu da daha yüksek kalitede ses haberleşmesi sağlamakla beraber akustik yankı gidericisine esneklik katacaktır.

Sonuç olarak IP tabanlı ses iletimi ortamında çoğunlukla konferans tip haberleşme esnasında karşılaşılan ve uzak-uç işaretinin gecikmesi ve yansımaları sonucu oluşan akustik yankı giderimi probleminin çözümünde, Fourier dönüşümünün genel bir hali olan KFD ile uyarlanırlı süzgeçleme algoritmalarından olan LMS'in birlikte kullanımı ile oluşan KFD bölgesi LMS algoritması zaman bölgesi diğer uyarlamalı süzgeçleme algoritmalarına göre daha iyi

performans sağlamıştır. Ayrıca hesaplanabilirliğinin kolaylığından da faydalanarak aktif konuşma-konferans esnasında bile en uygun benzetimi yapmak için KFD katsayısı ' a ' parametresini güncellenebilir. Böylece uyarlamalı süzgeçleme algoritmasına esneklik katılarak daha iyi performansla ve kaliteli haberleşmeye olanak sağlanır.

KAYNAKLAR

- Ahgren P. (2005), "Acoustic echo cancellation and doubletalk detection using estimated loudspeaker impulse responses", *Speech and Audio, IEEE Transactions*, 13:1231-1237.
- Almeida, L.B. (1994), "The Fractional Fourier Transform and Time-frequency Representations", *IEEE Transactions on Signal Process*, 42(11):3084-3091.
- Breining, C.; Dreiscitel, P.; Hansler, E.; Mader, A.; Nitsch, B.; Puder, H.; Schertler, T.; Schmidt, G.; Tilp, J. (1999) "Acoustic Echo Control. An Application of Very-high-order Adaptive Filters", *IEEE Signal Processing Magazine*, 16(4):42-69.
- Chew, W. C. ve Boroujeny, B.F. (1997), "Software simulation and real-time implementation of acoustic echocancelling", *Information, Communications and Signal Processing Conference*, 3:1270-1274.
- Choudhry, U.I.; Kim, J.W.; Kim, H.K (2006), "A Highly Adaptive Acoustic Echo Cancellation Solution for VoIP Conferencing Systems", *Computer Systems and Applications*, 2006. IEEE International Conference, 8 March.
- Eneman K. ve Moonen M. (1997), "Real-Time Implementation of An Acoustic Echo Canceller on DSP", *Proceedings of the ProRISC Workshop on Circuits, Systems and Signal Processing*.
- Gansler, T.; Gay, S.L.; Sondhi, M.M.; Benesty, J (1999), "Double-talk robust fast converging algorithms for network echocancellation", *Applications of Signal Processing to Audio and Acoustics*, 1:215-218.
- Goodwin, G. C. ve Sin, K. S., (1986), *Adaptive Filtering Prediction and Control*, Prentice-Hall, Englewood Cliffs.
- Gordy J.D. ve Goubran ,R.A. (2006), "On the Perceptual Performance Limitations of Echo Cancellers in Wideband Telephony", *Audio, Speech, and Language Processing, IEEE Transactions*, 14: 33 – 42.
- Gordy, J.D. ve Goubran, R.A. (2005), "A Perceptual Performance Measure for Adaptive Echo Cancellers in Packet-Based Telephony", *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference*, 6-6 July, Amsterdam.
- Haykin, S. (1996), *Adaptive Filter Theory*, Prentice Hall, New Jersey.
- He, P.; Dyba R. A.; Pessoa L. F. C.; (2003) "Network Echo Cancellers: Requirements, Applications and Solutions", *Motorola Inc.*
- ITU-T Recommendation G.168 (2007): "Digital network echo cancellers".
- Jeannes, W.L.B.; Scalart, P.; Faucon, G.; Beaugeant, C. (2001), "Combined Noise and Echo Reduction in Hands-free Systems: a Survey", *Speech and Audio Processing, IEEE Transactions*, 9(8):808 – 820.
- Jenq J. C. ve Hsieh S. (2001), "Acoustic echo cancellation using iterative-maximal-lengthcorrelation and double-talk detection", *Speech and Audio Processing, IEEE Transactions*, 9:932-942.
- Jingjing Cui; Naylor, P.A.; Brown, D.T. (2004), "An improved IPNLMS algorithm for echo cancellation in packet-switched networks", *IEEE International Conference, Acoustics*,

Speech, and Signal Processing (ICASSP apos;04), 4:141-144.

Kao C. (2003), "Design of echo cancellation and noise elimination for speech enhancement", IEEE Trans., 49:1468-1473.

Matić V. M. ve Abadžić S. N (2007), "Acoustic and line echo cancellation using adaptive filters", TELFOR 2007.

Mendlovic, D. ve Özaktas, H. M. (1993), "Fractional Fourier Transforms and Their Optical Implementation", I. Journal of the Optical Society of America A, 10: 1875-1881.

Murano, K.; Unagami, S.; Amano, F. (1990) "Echo Cancellation and Applications", IEEE Communications Magazine, 28:49-55.

Özaktas, H. M.; Arikan, O.; Kutay, M.A.; Bozdagi, G. (1996), "Digital Computation of the Fractional Fourier Transform", IEEE Transactions on Signal Processing, 44:2141-2150.

Özaktas, H. M.; Zalevsky, Z.; Kutay, M. A. (2001), The Fractional Fourier Transform With Applications in Optics and Signal Processing, John Wiley & Sons, Newyork.

Periakarruppan, G. ve b Abdul Rashid, H.A. (2005) "Evaluating the Performance of the Packet Based Echo Canceller During Packet Losses and the Solution for Packet Loss", 2005 13th IEEE International Conference, 16-18 Nov. 2005.

Radecki, J.; Zilic, Z.; Radecka, K. (2002), "Echo cancellation in IP networks", In Proc. The 2002 45th Midwest Symposium on Circuits and Systems MWSCAS-2002, 4-7 Aug. 2002.

Rages, M.ve Ho, K.C. (2002), "Limits on echo return loss enhancement on a voice coded speech signal", Circuits and Systems, MWSCAS-2002, 2:152-155.

Tahermezahdi, M.ve Liu, L (1995), "Real-time implementation of an IIR acoustic echo canceller onADSP21020", Conference on Acoustics, Speech, and Signal Processing, 4:2723-2726.

Valin J. (2007), " On Adjusting the Learning Rate in Frequency Domain Echo Cancellation With Double-Talk", Audio, Speech, and Language Processing, IEEE Transactions, 15:1030-1034.

Vaseghi, Saeed V. (1996), Advanced Signal Processing and Digital Noise Reduction, Wiley Teubner, Chichester.

Widrow, B. ve Hoff, M. E. (1960), Adaptive Switching Circuits, WESCON Convention Record, New York.

Yensen, T. N.; Goubran, R. A.; Lambadaris, I. (2001), "Synthetic Stereo Acoustic Echo Cancellation Structure for Multiple Participant VoIP Conferences", IEEE Transactions on Speech and Audio Processing, 9: 168-174.

Youhong Lu; Fowler, R.; Tian, W.; Thompson, L. (2005), "Enhancing echo cancellation via estimation of delay", Signal Processing, IEEE Transactions, 53:4159-4168.

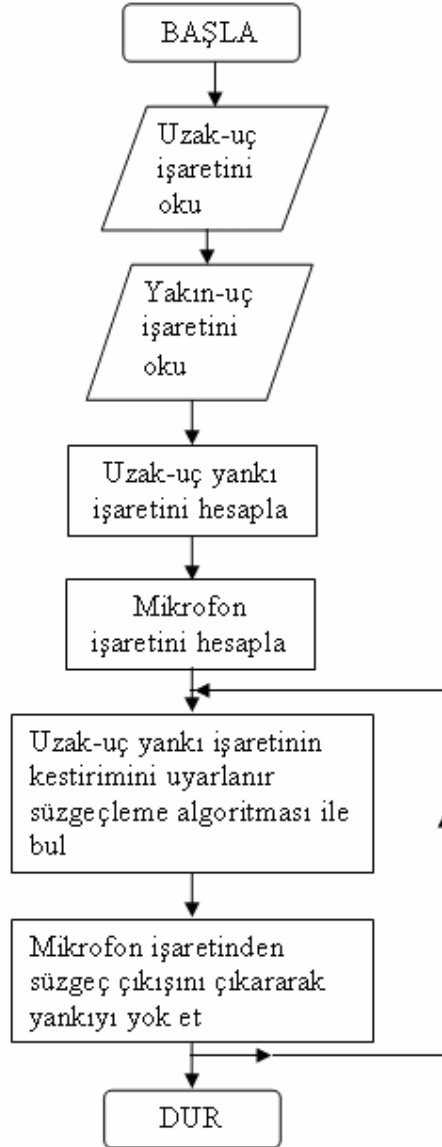
Zhongwei, Z. ve Tahermezahdi, M. (1996), "Transform Domain LMS Algorithm in Fullband and Subband Based Acoustic Echo Cancellation", Circuits and Systems, 1996., IEEE 39th Midwest symposium, 2:795-798.

EKLER

- Ek 1 Uyarlamalı süzgeçleme algoritmaları kullanılarak sağlanan akustik yankı giderimi için Matlab programı ile benzetim
- Ek 2 Uyarlanırlı Süzgeçleme Akış Şeması
- Ek 3 LMS algoritması
- Ek 4 NLMS algoritması
- Ek 5 RLS algoritması
- Ek 6 KFD Bölgesi LMS algoritması

Ek 1 Uyarlamalı süzgeçleme algoritmaları kullanılarak sağlanan akustik yankı giderimi için Matlab programı

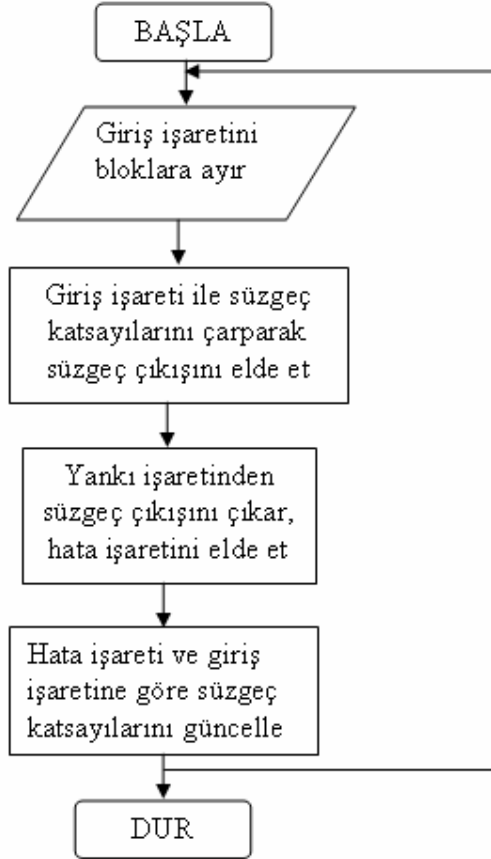
IP tabanlı ses iletimi ortamında akustik yankı giderimi için benzetim MATLAB programı kullanılarak yapılmaktadır. Program akış şeması Şekil Ek 1.1'deki gibidir.



Şekil Ek 1.1 Akustik yankı giderimi için benzetim program akış şeması

EK 2 Uyarlanır Süzgeçleme Akış Şeması

Bu tez kapsamında IP tabanlı ses iletimi ortamında akustik yankı giderimi, uyarlanır süzgeçleme algoritmaları kullanılarak sağlanmaktadır. Şekil Ek 2.1’de algoritmaların akış şeması verilmektedir.



Şekil Ek 2.1 Uyarlanır süzgeçleme algoritma akış şeması

EK 3 LMS algoritması

```
% LMS algoritması ile akustik yankı giderimi.

% Ns; işaretin uzunluğu

% filter_size; odanın dürtü yanıtı için süzgeç uzunluğu

% l; süzgeç katsayılarının sayısı


input_signal = wavread('.wav',Ns);%uzak uç işareti

near_end=wavread('.wav',Ns);%yakın uç işareti

impulse=zeros(filter_size,1);

for (i=1:10)

impulse(((i-1)*filter_size/10)+1)=1/i;

end


% Uzak-uç yankı işaretinin elde edilmesi için odanın dürtü yanıtı uzak-uç işareti
%konvolüsyona sokulur.

desired_signal = conv(input_signal, impulse);

desired_signal=desired_signal(1:Ns);% uzak-uç yankı işareti


% Mikrofon işareti uzak-uç yankı işareti ve yakın-uç konuşma işaretinden oluşur

primary = near_end + desired_signal;%microphone signal

w = zeros(1,1);%süzgeç katsayıları

mu =; % adım büyüklüğü


for n = 1:Ns;
```

%Süzgeç girişi uzak-uç işareti bloklar halinde uyarlanır süzgeçleme algoritmasına sokulur

```
frefblock = input_signal(n:-1:n-l+1);
```

```
refP(n) = conj(w')*frefblock; %yankı işaretinin kestirimi yapılır
```

```
error(n) = desired_signal(n) - refP(n); %hata işareti
```

```
w = w + mu*frefblock*error(n); %süzgeç katsayıları güncellenir
```

```
end;
```

% mikrofon işaretinden yankının süzülmesi

```
output = primary - refP';
```

EK 4 NLMS Algoritması

%NLMS algoritması ile akustik yankı giderimi.

%Ns; işaretin uzunluğu

%filter_size; odanın dürtü yanıtı için süzgeç uzunluğu

% l; süzgeç katsayılarının sayısı

input_signal = wavread('.wav',Ns);%uzak uç işareti

near_end=wavread('.wav',Ns);%yakın uç işareti

impulse=zeros(filter_size,1);

for (i=1:10)

impulse(((i-1)*filter_size/10)+1)=1/i;

end

% Uzak-uç yankı işaretinin elde edilmesi için odanın dürtü yanıtı uzak-uç işareti
%konvolüsyona sokulur.

desired_signal = conv(input_signal, impulse);

desired_signal=desired_signal(1:Ns);% uzak-uç yankı işareti

% Mikrofon işareti uzak-uç yankı işareti ve yakın-uç konuşma işaretinden oluşur

primary = near_end + desired_signal;%microphone signal

w = zeros(1,1);%süzgeç katsayıları

mu =; % adım büyüklüğü

for n = 1:Ns;

%Süzgeç girişi uzak-uç işareti bloklar halinde uyarlanır süzgeçleme algoritmasına sokulur

frefblock = input_signal(n:-1:n-l+1);

refP(n) = conj(w')*frefblock; %yankı işaretinin kestirimi yapılır

error(n) = desired_signal(n) - refP(n); %hata işareti

step_size=mu/(0.0001+frefblock'*frefblock); %normalizasyon

w = w + step_size*frefblock*conj(error(n)); %süzgeç katsayıları %güncellenir

end;

% mikrofon işaretinden yankının süzülmesi

output = primary - refP';

EK 5 RLS Algoritması

%RLS algoritması ile akustik yankı giderimi.

%Ns; işaretin uzunluğu

%filter_size; odanın dürtü yanıtı için süzgeç uzunluğu

% l; süzgeç katsayılarının sayısı

input_signal = wavread('.wav',Ns);%uzak uç işareti

near_end=wavread('.wav',Ns);%yakın uç işareti

impulse=zeros(filter_size,1);

for (i=1:10)

impulse(((i-1)*filter_size/10)+1)=1/i;

end

% Uzak-uç yankı işaretinin elde edilmesi için odanın dürtü yanıtı uzak-uç işareti
%konvolüsyona sokulur.

desired_signal = conv(input_signal, impulse);

desired_signal=desired_signal(1:Ns);% uzak-uç yankı işareti

% Mikrofon işareti uzak-uç yankı işareti ve yakın-uç konuşma işaretinden oluşur

primary = near_end + desired_signal;%microphone signal

w = zeros(l,1); %süzgeç katsayıları

P=eye(M)*100; %ters ilinti matrisi

```

input_signal= input_signal(:);

desired_signal = desired_signal(:);

lambda=1; %Lagrange çarpanı

for n=1:Ns

%Süzgeç girişi uzak-uç işareti bloklar halinde uyarlanır süzgeçleme algoritmasına sokulur

frefblock= input_signal(n:-1:n-l+1);

k=lambda^(-1)*P* frefblock/(1+lambda^(-1)* frefblock'*P* frefblock);

refP(n)=w'*frefblock; %yankı işaretinin kestirimi yapılır

error(n)=desired_signal(n)-w'*frefblock; %hata işareti

w=w+k*conj(error(n)); %süzgeç katsayıları güncellenir

P=lambda^(-1)*P-lambda^(-1)*k*frefblock'*P; %ters ilinti matrisi güncellenir

end

% mikrofon işaretinden yankının süzülmesi

output=primary-refP';

```

EK 6 KFD Bölgesi LMS Algoritması

```
% KFD bölgesi LMS algoritması ile akustik yankı giderimi.

% Ns; işaretin uzunluğu

% filter_size; odanın dürtü yanıtı için süzgeç uzunluğu

% a; KFD derecesi

% l; süzgeç katsayılarının sayısı


input_signal = wavread('.wav',Ns);%uzak uç işareti

near_end=wavread('.wav',Ns);%yakın uç işareti

impulse=zeros(filter_size,1);

for (i=1:10)

    impulse(((i-1)*filter_size/10)+1)=1/i;

end


% Uzak-uç yankı işaretinin elde edilmesi için odanın dürtü yanıtı uzak-uç işareti
%konvolüsyona sokulur.

desired_signal = conv(input_signal, impulse);

desired_signal=desired_signal(1:Ns);% uzak-uç yankı işareti


% Mikrofon işareti uzak-uç yankı işareti ve yakın-uç konuşma işaretinden oluşur

primary = near_end + desired_signal;%microphone signal


w = zeros(l,1);%süzgeç katsayıları

mu =; % adım büyüklüğü
```

```

a=;

% uzak-uç yankı işareti ve uzak-uç işaretinin KFD dönüşümleri

d1=interp(desired_signal,2);

Nd=(length(d1)-1)/2;

dd = sqrt(Nd);

desired_frft_double = frft2(d1, a, 'interp_in_and_out', dd);

desired_frft = downsample(desired_frft_double,2);


f1=interp(input_signal,2);

Nf=(length(f1)-1)/2;

df = sqrt(Nf);

fref_frft_double = frft2(f1, a, 'interp_in_and_out', df);

fref_frft = downsample(fref_frft_double,2);


for n = 1:Ns;

%Süzgeç girişi uzak-uç işareti bloklar halinde uyarlanır süzgeçleme algoritmasına sokulur

frefblock = fref_frft(n:-1:n-l+1);

refP(n) = conj(w')*frefblock; %yankı işaretinin kestirimi yapılır

error(n) = desired_frft (n) - refP(n); %hata işareti

w = w + mu*conj(frefblock)*error(n); %süzgeç katsayıları güncellenir

end;

```

% Süzgeç çıkışı ve hata işaretinin ters KFD dönüşümü alınır.

```
r1=interp(refP,2);
```

```
Nr=(length(r1)-1)/2;
```

```
dr = sqrt(Nr);
```

```
refP_frft_double = frft2(r1, -a, 'interp_in_and_out', dr);
```

```
refP_frft = real(downsample(refP_frft_double,2));
```

```
e1=interp(error,2);
```

```
Ne=(length(e1)-1)/2;
```

```
de = sqrt(Ne);
```

```
error_frft_double = frft2(e1, -a, 'interp_in_and_out', de);
```

```
error_frft = real(downsample(error_frft_double,2));
```

% mikrofon işaretinden yankının süzülmesi

```
output = primary - refP_frft';
```

ÖZGEÇMİŞ

Doğum tarihi 27.12.1981

Doğum yeri Ağrı

Lise 1996-1999 Muş Anadolu Lisesi

Lisans 1999-2003 İstanbul Üniversitesi Mühendislik Fak.
Elek-Elektronik Mühendisliği Bölümü

Yüksek Lisans 2003- Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
Elektronik-Haberleşme Müh. Anabilim Dalı,
Haberleşme Programı

Çalıştığı kurum(lar)

2003-2005 Delta Petrol A. Ş.
2005-Devam ediyor Nortel Netaş Networks