

85007

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YİNELEMELİ EN KÜÇÜK KARELER KAFES
ALGORİTMASINI KULLANARAK KANAL
DENGELEME**

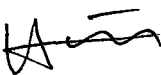
Elektronik ve Haberleşme Müh. Özgür GÖKGÜL

**F. B. E. Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Haberleşme Programında Hazırlanan**

YÜKSEK LİSANS TEZİ

**T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ**

Tez Danışmanı: Yrd. Doç. Dr. Soner ÖZGÜNEL

Yrd. Doç. Dr.,
Soner ÖZGÜNEL


Prof. Metin YÜCEL

İSTANBUL, 1999

Prof. Dr. Ümit AYGÖLÜ


İÇİNDEKİLER

Sayfa

ÖNSÖZ	v
ÖZET	vi
ABSTRACT	vii
1. GİRİŞ	1
2. DURAĞAN AYRIK ZAMANLI RASGELE SÜREÇLER	4
2.1 Ayrık Zamanlı Bir Rasgele Sürecin Kısmi Karakteristikleri	5
2.2 İlişki Matrisi	7
2.2.1 İlişki matrisinin özellikleri	8
2.3 Özdeğer Analizi	14
2.3.1 Özdeğer ve özvektörlerin özellikleri	17
3. WIENER FİLTRELER	27
3.1 Optimum Filtreleme Problemi	29
3.2 Hata Başarım Yüzeyi	31
3.3 Normal Denklemleri	34
3.4 Diklik Kuralı	35
3.5 Minimum Karesel Ortalama Hata	37
3.6 Hata Başarım Yüzeyinin Kanonik Formu	39
4. DOĞRUSAL ÖNGÖRÜ	41
4.1 İleri Yönde Doğrusal Öngörü	42
4.1.1 İleri yönde öngörü hata filtresi	46
4.1.2 İleri yönde öngörü için genişletilmiş normal denklemleri	46
4.2 Geri Yönde Doğrusal Öngörü	47
4.2.1 Geri ve ileri yönde öngörücüler arasındaki ilişki	50
4.2.2 Geri yönde öngörü hata filtresi	52
4.2.3 Geri yönde öngörü için genişletilmiş normal denklemleri	54
4.3 Levinson–Durbin Yinelemesi	55
4.3.1 Δ_{m-1} ve Γ_m parametrelerinin anlamı	60
4.4 Kafes Öngörücüler	62
4.4.1 Öngörü hatalarının derece güncelleme yinelemeleri	62
4.4.2 Kafes filtrelerin ilişki özellikleri	68
4.5 Gram–Schmidt Dikleştirme Yöntemi Kullanılarak Geri Yönde Öngörü Hata Dizisinin Üretilmesi	69
4.5.1 Tap girişlerine ait ilişki matrisinin pozitif belirli olmasının öngörü hata filtresinin minimum fazlı olmasına eşdeğerliği	71
4.6 Birleşik Süreç Kestirimi	72

5.	UYARLAMALI FİLTRELEME	76
5.1	Uyarlamalı Filtre Yapısı	77
5.2	Yinelemeli En Küçük Kareler Kestirimi	79
5.2.1	Bazı ön bilgiler	80
5.2.2	Matris tersi alma yasası	84
5.2.3	Üstel ağırlıklı yinelemeli en küçük kareler algoritması	84
5.2.3.1	Tap ağırlık vektörünün zaman güncellemesi	85
5.2.3.2	Başlangıç koşulları	87
5.2.4	Hata karelerinin ağırlıklı toplamını güncelleyen yineleme	88
5.2.5	RLS algoritmasının yakınsama analizi	89
5.2.5.1	Tap ağırlık vektörünün ortalama anlamda yakınsaması	90
5.2.5.2	Tap ağırlık vektörünün karesel ortalama anlamda yakınsaması	92
5.2.5.3	RLS algoritmasının karesel ortalama anlamda yakınsaması	93
5.3	Uyarlamalı Doğrusal Öngörü	97
5.3.1	İleri yönde uyarlamalı doğrusal öngörü	98
5.3.2	Geri yönde uyarlamalı doğrusal öngörü	104
5.3.3	Geniştirilmiş kazanç vektörü	108
5.3.4	Dönüşüm çarpanı	110
5.4	Yinelemeli En Küçük Kareler Kafes Filtreler	112
5.4.1	Derece güncelleme yinelemeleri	113
5.4.1.1	$\Delta_{m-1}(n)$ ve $\Delta'_{m-1}(n)$ parametreleri arasındaki ilişki	117
5.4.1.2	Sonraki öngörü hataları için derece güncelleme yinelemeleri	118
5.4.1.3	Sonraki öngörü hata karelerinin ağırlıklı toplamına ait derece güncelleme yinelemeleri	121
5.4.1.4	$\gamma_m(n-1)$ için derece güncelleme yinelemesi	121
5.4.2	Zaman güncelleme yinelemesi	122
5.4.3	Yinelemeli en küçük kareler kafes algoritmasının başlangıç koşulları	125
5.4.4	Birleşik süreç kestirimi	126
5.4.4.1	Yinelemeli en küçük kareler metoduna ait $\hat{w}_m(n)$ tap ağırlık vektörü ile $K_m(n)$ gerileme katsayı vektörü arasındaki ilişki	130
5.4.4.2	Gerileme katsayılarını hesaplayan algoritma	132
6.	SAYISAL UYGULAMA	138
6.1	Uyarlamalı Dengeleme Kavramı	139
6.2	Yinelemeli En Küçük Kareler Kafes Algoritmasını Kullanarak Kanal Dengeleme	143
6.2.1	Uyarlamalı dengeleyicinin öğrenme eğrisi hakkında kısa bilgi	147
6.2.2	Uyarlamalı dengeleyicinin impuls yanıtının hesaplanması	149
6.3	Bilgisayar Deneyleri	154
6.3.1	Öğrenme eğrileri	156
6.3.2	$ W(w) $, $ H(w) $ ve $ 1/H(w) $ eğrileri	215
7.	SONUÇLAR	230

KAYNAKLAR	233
EKLER	235
Ek 1 Sonraki Kestirim Hatalarına Dayalı RLSL Algoritmasını Kullanarak Uyarlamalı Kanal Dengeleme Deneyini Gerçekleyen Matlab Programı	236
Ek 2 Ayrık Zamanlı Fourier Dönüşümünü Hesaplayan Matlab Programı	239
Ek 3 Uyarlamalı Dengeleyicinin İmpuls Yanıtını $M = 6$ İçin Hesaplayan Matlab Programı	240
Ek 4 Uyarlamalı Dengeleyicinin İmpuls Yanıtını $M = 8$ İçin Hesaplayan Matlab Programı	241
Ek 5 Uyarlamalı Dengeleyicinin İmpuls Yanıtını $M = 10$ İçin Hesaplayan Matlab Programı	243
Ek 6 $ W(w) $, $ H(w) $ ve $ 1/H(w) $ Eğrilerini $M = 6$ İçin Çizen Matlab Programı	246
Ek 7 $ W(w) $, $ H(w) $ ve $ 1/H(w) $ Eğrilerini $M = 8$ İçin Çizen Matlab Programı	247
Ek 8 $ W(w) $, $ H(w) $ ve $ 1/H(w) $ Eğrilerini $M = 10$ İçin Çizen Matlab Programı	248
ÖZGEÇMİŞ	249

ÖNSÖZ

Yüksek Lisans öğrenimi görmem için ısrarlı tutumundan ve kayıtsız şartsız desteğinden dolayı, biricik **Anneme**; Yüksek Lisans öğrenimine devam etmem için verdiği nasihat ve manevi desteğinden dolayı, sevgili hocam **Yrd. Doç. Dr. Aktül KAVAS**'a ve bu tezin hazırlanması sırasında bana gösterdiği sabır ve sıcak yaklaşımından dolayı da, değerli hocam **Yrd. Doç. Dr. Soner ÖZGÜNEL**'e sonsuz teşekkürlerimi sunarım.

Özgür GÖKGÜL



ÖZET

Kanal dengelemenin amacı, kanalın bozucu etkilerini gidererek iletilen işareti alınan işaretten yeniden elde etmektir. Bir kanal dengeleyici, iletilen işaretin olabildiğince doğru kestirimini elde etmeye çalıştığı için, optimum bir filtredir. Uyarlamalı kanal dengeleme ile ilgili olarak, Wiener filtre teorisine dayalı çeşitli metotlar vardır. Bu çalışmada, doğrusal kanalları dengelemek üzere yinelemeli en küçük kareler kafes (RLSL) algoritması kullanılmaktadır. RLSL algoritmasının avantajı: modüler bir yapıya, düşük hesaplama karmaşıklığına ve hızlı yakınsama özelliklerine sahip olmasıdır.

İlk bölümde, algoritmanın başarımı öğrenme eğrileri bazında incelenmektedir. Bu başarım analizinde, farklı kanal modelleri seçilmekte ve gürültü değişintisinin, kafes filtre modül sayısının ve kanal parametresinin etkileri incelenmektedir. Bilgisayar deneylerine ait sonuçların literatürdeki teorik sonuçlar ile uyum gösterdiği görülmektedir.

Bilindiği gibi, dengeleyicinin transfer fonksiyonu kanalın transfer fonksiyonunun tersi olmaktadır. İkinci bölümde, bu özelliği göstermek için bilgisayar deneyleri gerçekleştirilmekte ve dengeleyicinin transfer fonksiyonu ile kanalın transfer fonksiyonunun tersinin birbirine uyduğu gösterilmektedir.

ABSTRACT

The purpose of channel equalization is to undo the distorting effects of the channel and recover the transmitted signal from the received signal. A channel equalizer is an optimal filter since it tries to produce as good estimate of the transmitted signal as possible. For adaptive channel equalization, there are several methods based on Wiener filter theory. In this study, the recursive least squares lattice (RLSL) algorithm is used in the equalization of linear channels. The advantage of the RLSL algorithm is that it has a modular structure, less computational complexity and rapid convergence properties.

In the first part, the performance of the algorithm is investigated in the base of learning curves. In this performance analysis, different channel models are chosen and the effects of the noise variance, the number of lattice filter modules and the channel parameter are investigated. It is seen that the results of the computer experiments are in accordance with the theoretical results in literature.

It is known that the transfer function of the equalizer is the inverse of the transfer function of the channel. In the second part, some computer experiments are performed to show this property and, it is shown that the equalizer's transfer function coincides with the reciprocal of the channel's transfer function.

1. GİRİŞ

Doğrusal kanalların dengelenmesi, bir çok araştırmacının ilgi odağı haline gelmiş ve bu konu üzerine pek çok çalışma yapılmıştır. Bu tezde, yinelemeli en küçük kareler kafes (RLSL) algoritmasını kullanarak doğrusal kanalların dengelenmesi incelenmektedir.

Kanal dengelenmede amaç: kanalın bozucu etkilerini gidererek, verici tarafından iletilen işareti alınan işaretten yeniden elde etmektir. Bir kanal dengeleyici, uyarlamalı olsun veya olmasın, optimum bir filtredir. Çünkü, iletilen işaretin olabildiğince doğru kestirimini üretmeye çalışmaktadır. Kanal dengeleyicinin aktarım işlevi, kanalın aktarım işlevinin tam tersi biçiminde oluşmakta ve böylece kanalın çıkışına yerleştirildiği zaman, kanalın iletilen işaret üzerindeki etkilerini yok ederek gerçek işaretin doğru bir biçimde kestirimini sağlamaktadır. Bir çok durumda olduğu gibi, kanalın yapısı önceden bilinmiyor veya kanal zamanla değişiyor olabilir. Bu durumda, dengeleyicinin uyarlamalı bir biçimde tasarlanması gerekir. Uyarlamalı dengeleyiciler, eğitim modu ve karar-yönetimli mod olmak üzere iki modda çalıştırılırlar. İletilen işaret alıcıda bulunmadığı için, önce eğitim modunda çalışılır ve ardından karar-yönetimli moda geçilir. Eğitim modunda, bilinen bir test işareti iletilerek kanalın tanımlanması sağlanır. Test işareti sayesinde ayarlanabilir katsayılarına başlangıç değerleri verilen dengeleyici, artık gerçek mesajı almaya hazırdır. Gerçek mesajın iletilmesiyle birlikte, dengeleyicinin çıkışı istenen yanıtın çıkarılarak bir kestirim hatası oluşturulur ve bu hata, toplam karesel hatayı azaltacak şekilde dengeleyicinin katsayılarının ayarlanmasında kullanılır.

Uyarlamalı dengeleyicilerin çalışma prensibini oluşturan yinelemeli algoritmaların çıkarılması için kullanılan, birbirinden farklı üç temel yöntem vardır: Wiener filtre teorisi, Kalman filtre teorisi ve en küçük kareler yöntemi. Wiener filtreleme problemlerinin çözümü optimumdur ve Wiener çözümü olarak bilinir. Bu üç temel yöntemle geliştirilen yinelemeli algoritmalarından bazıları: en küçük karesel ortalama (LMS) algoritması, yinelemeli en küçük kareler (RLS) algoritması ve yinelemeli en küçük kareler kafes (RLSL) algoritmalarıdır. Bu tezde kullanılan yinelemeli en küçük kareler kafes algoritmaları, filtre katsayılarının güncellenmesinde kullanılmak üzere, dik işaret bileşenleri kümesi oluşturarak yakınsama hızının artmasını sağlayan oldukça verimli bir yapıya sahiptir. Bu bileşenlerin elde edilmesi, Gram-Schmidt dikleştirme yöntemi ile sağlanmaktadır. Üretilen bu işaret bileşenleri

birbirlerine diktirler ve giriş işaretiyle aralarında bire bir ilişki vardır; öyle ki, biri diğerinden herhangi bir bilgi kaybı olmadan elde edilebilir. Böylece, herhangi bir öngörü probleminde, giriş işareti yerine bu dik bileşenler kullanılarak algoritmanın yakınsama hızının artırılması sağlanır.

Bölüm 2’de, durağan ayırık zamanlı rasgele süreçlerin kısmi karakteristikleri ele alınmaktadır. Özellikle de, bu tür bir rasgele sürecin ilişki matrisi, ilişki matrisinin özdeğerleri ve özvektörleri üzerinde durulmaktadır.

Bölüm 3’te, uyarlamalı filtrelerin çalışma prensibini belirleyen yinelemeli algoritmaların çıkarılmasına temel teşkil etmesi bakımından, Wiener filtre teorisi tanıtılmaktadır. Wiener-Hopf integral denklemlerinin ayırık zamandaki eşdeğeri olan normal denklemleri anlatılmakta ve bu konunun ardından da, çok iyi bilinen diklik kuralına değinilmektedir. Son olarak, bir enine filtrenin durağan bir ortamda çalıştırılması sonucu oluşan hata başarım yüzeyinin, ilişki matrisinin özdeğer-özvektör ayrıştırmasından yararlanmak suretiyle, kanonik formu çıkarılmaktadır.

Bölüm 4’te, Wiener filtre teorisi doğrusal öngörü problemine uygulanarak, doğrusal öngörü probleminin iki temel formu (ileri yönde doğrusal öngörü ve geri yönde doğrusal öngörü) incelenmektedir. İleri ve geri yönde doğrusal öngörücüler arasındaki ilişki sayesinde, Levinson–Durbin yinelemesi elde edilmektedir. Levinson–Durbin yinelemesinin ortaya çıkardığı en önemli ürün, yansıma katsayılarıdır. Levinson–Durbin yinelemesi kullanılarak, ileri ve geri yönde doğrusal öngörücüyü tek bir yapıda birleştiren kafes öngörücü meydana getirilmekte ve ilişki özellikleri anlatılmaktadır. Kafes öngörücülerin çalışma prensibi hakkında daha iyi fikir edinebilmek amacıyla, Gram–Schmidt dikleştirme yöntemi kısaca açıklanmaktadır. Son olarak, optimum filtreleme problemine mükemmel bir çözüm getiren birleşik süreç kestirimi, kafes tabanlı olarak ele alınmaktadır.

Bölüm 5’te, önceki bölümlerde anlatılanların ışığı altında, uyarlamalı filtre teorisine girilmekte ve ilk olarak da, uyarlamalı enine filtre yapısı anlatılmaktadır. Bu yapıdan anlaşılacağı gibi, Bölüm 3 ve Bölüm 4’te ele alınan sabit katsayılı enine filtrelerin ayarlanabilir katsayılı olan uyarlamalı enine filtreler, katsayılarının ayarlanması için yinelemeli bir algoritmaya ihtiyaç duyarlar. Wiener filtre teorisinin deterministik

(zaman ortalamalı) karşılığı olan en küçük kareler yöntemine dayalı yinelemeli en küçük kareler (RLS) algoritması, bu amaçla geliştirilen bir algoritmadır. Bu nedenle, uyarlamalı filtre yapısına değinildikten sonra, matris tersi alma yasası yardımıyla RLS algoritmasının elde edilmesi anlatılmaktadır. RLS algoritması için son olarak, yakınsama analizi yapılmaktadır. RLS algoritmasının yakınsama hızı yüksektir ama hesaplama karmaşıklığı fazladır. Hesaplama karmaşıklığının fazla oluşu, hızlı algoritmaların geliştirilmesine yol açmaktadır. Bu hızlı algoritmalarından biri olan yinelemeli en küçük kareler kafes (RLSL) algoritmasının anlatımına geçmeden önce, hızlı algoritmaların anlaşılmasında temel teşkil eden ileri ve geri yönde uyarlamalı doğrusal öngörücüler ele alınmaktadır. Daha sonra da, RLSL algoritmasına ait derece ve zaman güncelleme yinelemeleri çıkarılmaktadır. RLSL algoritması, uyarlamalı öngörücü veya uyarlamalı birleşik süreç kestirici olarak kullanılabilir. Modüler yapısı ve hesaplama karmaşıklığının az oluşu, bu algoritmayı üstün kılan özelliklerdir.

Bölüm 6'da, uyarlamalı kanal dengeleme kavramı üzerinde durulmakta ve uyarlamalı dengeleyicilerin çalışma prensibi hakkında ayrıntılı bilgiler verilmektedir. Daha sonra, RLSL algoritmasının uyarlamalı birleşik süreç kestirici olarak kullanılabilmesinden yararlanılarak oluşturulan bir uyarlamalı dengeleyici ile, doğrusal kanalların dengelenmesine yönelik sayısal uygulamalar yapılmaktadır. Bu uygulamalardan elde edilen eğriler, iki ayrı alt bölümde sunulmakta ve yorumlanmaktadır.

Bölüm 7'de, bu tez çalışmasında uygulanan yöntem ve elde edilen sonuçlar kısaca anlatılmaktadır.

2. DURAĞAN AYRIK ZAMANLI RASGELE SÜREÇLER

Stokastik süreç (stochastic process) veya rasgele süreç (random process) terimi, olasılık kanunlarına bağlı olarak istatistiksel bir olayın zamanla değişimini ifade etmek amacıyla kullanılır. Bir olayın istatistiksel olması demek, o olayla ilgili bir deney gerçekleştirmeden önce o olayın zamanla değişim şeklinin nasıl olacağının tam olarak belirlenebilmesinin imkansız olması demektir. Konuşma işaretleri, televizyon işaretleri, radar işaretleri, haberleşme kanalı çıkışındaki işaretler, sayısal bilgisayar verileri ve gürültü birer rasgele süreçtir. Burada ilgileneceğimiz rasgele süreç tipi, zamanın *ayrık ve düzgün dağılımlı* anlarında tanımlı olanıdır (Box ve Jenkins, 1976; Priestley, 1981). Rasgele süreci orijinal haliyle, gerçel değerli *sürekli* bir zaman aralığında tanımlamak da mümkündür; ama, sürecin işlenmeden önce, en büyük frekans bileşeninin iki katına eşit veya daha büyük bir örnekleme hızıyla zamanda *düzgün olarak örneklenmesi* tercih edilmektedir (Haykin, 1988).

Bir rasgele sürecin, zamanın sadece tek bir fonksiyonu olarak görülmemesi gerekir. Teoride, rasgele süreçlerin sonsuz sayıda farklı gerçeklemeler içerdikleri kabul edilmektedir. Ayrık zamanlı bir rasgele sürecin tek bir gerçeklemesine *ayrık zaman serisi* veya kısaca *zaman serisi* adı verilir. Gösterimde kolaylık sağlaması açısından burada, zamanı örnekleme periyoduna göre normalize edeceğiz. Örneğin; $u(n)$, $u(n-1)$, ..., $u(n-M+1)$ serisi, sürecin n anında yapılan $u(n)$ *şimdiki* gözlemi ile beraber $n-1$, $n-2$, ..., $n-M+1$ anlarında yapılmış $(M-1)$ adet *geçmiş* gözleminden oluşan bir zaman serisini göstermektedir.

Eğer bir rasgele sürecin istatistiksel özellikleri zamanın ötelenmesi ile birlikte herhangi bir değişikliğe uğramıyor ise, o rasgele süreç *dar anlamda durağan (strictly stationary)* olarak kabul edilir. $u(n)$, $u(n-1)$, ..., $u(n-M+1)$ zaman serisi ile belirtilen ayrık zamanlı bir rasgele sürecin dar anlamda durağan olabilmesi için: n , $n-1$, ..., $n-M+1$ anlarında yapılmış bu gözlemlerine ait *birleşik olasılık yoğunluk fonksiyonunun (joint probability density function)*, sabit bir M değerine karşılık n 'e ne değer verilirse verilsin hiç değişmeden aynı kalması gerekir.

2.1 Ayırık Zamanlı Bir Rasgele Sürecin Kısmi Karakteristikleri

Pratikte, ölçüm yapma yoluyla birleşik olasılık yoğunluk fonksiyonunun belirlenmesi imkansızdır. Diğer bir ifadeyle, rasgele süreç üzerinde yapılmış gözlem değerlerine bakarak, bu gözlemlere ait birleşik olasılık yoğunluk fonksiyonunu belirlemek mümkün değildir. Bu nedenle, sürecin birinci ve ikinci momentlerini belirleyip, kendimizi sürecin bu *kısmi karakteristikleri (partial characterization)* ile sınırlamak zorundayız.

$u(n)$, $u(n-1)$, ..., $u(n-M+1)$ zaman serisi ile gösterilen ve kompleks değerler de alabilen ayırık zamanlı bir rasgele süreç ele alınsın. Bu sürece ait *ortalama (beklendik) değer fonksiyonu (mean-value function)*

$$\mu(n) = E[u(n)] \quad (2.1)$$

ve *özilişki fonksiyonu (autocorrelation function)*

$$r(n, n-k) = E[u(n)u^*(n-k)] , \quad k = 0, \pm 1, \pm 2, \dots, \quad (2.2)$$

şeklinde tanımlanır. Burada asteriks işareti (*), *kompleks eşleniği* göstermektedir. Sürecin *özdeğişinti fonksiyonu (autocovariance function)* ise,

$$c(n, n-k) = E[(u(n) - \mu(n))(u(n-k) - \mu(n-k))^*] , \quad k = 0, \pm 1, \pm 2, \dots, \quad (2.3)$$

ile ifade edilir. Bu üç eşitlikten de görüldüğü gibi, rasgele sürecin ortalama değer, özilişki ve özdeğişinti fonksiyonları arasında

$$c(n, n-k) = r(n, n-k) - \mu(n)\mu^*(n-k) \quad (2.4)$$

şeklinde bir ilişki vardır. Öyleyse, çeşitli n ve k değerlerinde sürecin bu kısmi karakteristiklerini elde etmek için, ortalama değer fonksiyonu $\mu(n)$ ile birlikte özilişki $r(n, n-k)$ veya özdeğişinti $c(n, n-k)$ fonksiyonlarından herhangi bir tanesinin belirlenmesi gereklidir.

Kısmi karakteristikler bu halleri ile iki önemli avantaj sağlamaktadır:

1. Pratik ölçümler için oldukça elverişlidirler.
2. Rasgele süreçler ile yapılan *doğrusal (linear)* işlemlere uyumluluk gösterirler.

Dar anlamda durağan ayırık zamanlı bir rasgele süreç için (2.1), (2.2) ve (2.3) eşitlikleri daha da basitleşir. Özellikle, sürecin ortalama değer fonksiyonunun μ gibi bir sabite eşit olduğu ve bu nedenle de,

$$\mu(n) = \mu, \quad \text{her } n \text{ için} \quad (2.5)$$

şeklinde yazılabileceği görülür. Ayrıca, özilişki ve özdeğişinti fonksiyonlarının sadece (n) ile $(n-k)$ gözlem anları arasındaki farka, yani k değerine bağlı olduğu görülür. Bu durum,

$$r(n, n-k) = r(k) \quad (2.6)$$

$$c(n, n-k) = c(k) \quad (2.7)$$

şeklinde gösterilir. İki gözlem zamanı arasında geçen sürenin sıfır olmasına ($k = 0$), *sıfır zaman farkı (a time difference or lag of zero)* denir. Bu durumda, $r(0)$ değeri $u(n)$ 'in *kareseel ortalama (kareseel beklendik) değerine (mean-square value)* eşit olurken

$$r(0) = E[|u(n)|^2] \quad (2.8)$$

$c(0)$ değeri de $u(n)$ 'in *değişintisine (variance)*

$$c(0) = \sigma_u^2 \quad (2.9)$$

eşit olmaktadır.

(2.5), (2.6) ve (2.7) eşitliklerindeki şartlar, ayırık zamanlı rasgele sürecin dar anlamda durağan olmasını garanti etmek için yeterli değildir. Diğer yandan, dar anlamda durağan olmayan ama bu şartları sağlayan ayırık zamanlı rasgele süreçler *geniş anlamda durağan (wide-sense stationary)*, *zayıf durağan (weakly stationary)* veya *ikinci mertebeden durağan (stationary to the second order)* kabul edilir.

Doob'a (1953) göre geniş anlamda durağan bir $\{u(n)\}$ süreci, ancak ve ancak

$$E[|u(n)|^2] < \infty, \quad \text{her } n \text{ için}$$

olmakta ise dar anlamda durağandır. Bu koşul, mühendislikte ve fiziksel bilimlerde karşılaşılan rasgele süreçlerin pek çoğu tarafından sağlanmaktadır. Dikkat edilmesi gereken bir diğer ilginç ayrıntı da, geniş anlamda durağan olan bir *Gauss sürecinin* (*Gaussian process*) aynı zamanda dar anlamda da durağan varsayılabilmesidir. $\mu(n) = \mu = 0$ olması durumunda, geniş anlamda durağan bir rasgele sürecin özilişki ve özdeğişinti fonksiyonları birbirine eşit kabul edilir. Analizde kolaylık sağlaması açısından, bundan böyle aksi belirtilmediği sürece tüm rasgele süreçler sıfır ortalımalı kabul edilecektir.

2.2 İlişki Matrisi

$u(n)$ vektörü, $u(n), u(n-1), \dots, u(n-M+1)$ zaman serisinin elemanlarını içeren $M \times 1$ boyutlu gözlem vektörü (*observation vector*) olsun.

$$\mathbf{u}^T(n) = [u(n), u(n-1), u(n-2), \dots, u(n-M+1)] \quad (2.10)$$

Burada üstel işaret T , vektörün açılımını tek bir satıra yazabilmek amacıyla kullanılan *transpoz* (*transposition*) operatörünü belirtmektedir. Böyle bir zaman serisi ile gösterilen durağan ayrık zamanlı bir rasgele sürece ait *ilişki matrisi* (*correlation matrix*) ise, gözlem vektörünün kendisi ile *dış çarpımının* (*outer product*) beklendik değeri şeklinde tanımlanır. $\mathbf{R}$, bu tanıma uyan $M \times M$ boyutlu ilişki matrisini göstermek üzere,

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)] \quad (2.11)$$

eşitliği varolacaktır. Üstel işaret H , vektörün kompleks eşleniğinin transpozundan $\mathbf{u}^H(n) = (\mathbf{u}^*(n))^T$ oluşan *Hermitian transpozunu* (*Hermitian transposition*) belirtmektedir. (2.10) eşitliğinin (2.11) eşitliğinde yerine yazılmasına ilaveten geniş anlamda durağanlık koşulu da kullanılarak, $\mathbf{R}$ ilişki matrisinin açılımı

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & \dots & r(M-1) \\ r(-1) & r(0) & \dots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(-M+1) & r(-M+2) & \dots & r(0) \end{bmatrix} \quad (2.12)$$

biçiminde ifade edilir. Ana diyagonaldeki $r(0)$ elemanı, her zaman için gerçel değerlidir. Matrisin geriye kalan elemanlarının ne olacağı ise, gözlem değerlerine bağlıdır; bu elemanlar, kompleks değerli gözlemler için kompleks değerli, gerçel değerli gözlemler için ise gerçel değerli olmaktadır.

2.2.1 İlişki matrisinin özellikleri

Ayrık zamanlı filtre tasarımında ve istatistiksel analizde, ilişki matrisi önemli bir rol oynar. Bu nedenle, ilişki matrisine ait çeşitli özelliklerin ve bu özelliklerin ne anlama geldiklerinin çok iyi anlaşılması gerekmektedir. (2.11) eşitliğine ait tanımın kullanılmasıyla, durağan ayrık zamanlı bir rasgele sürecin ilişki matrisinin aşağıda verilen özelliklere sahip olduğu bulunur:

Özellik – 1 Durağan ayrık zamanlı rasgele süreçlerin ilişki matrisleri *Hermitian* olmaktadır.

İlişki matrisi $\mathbf{R}$, her zaman için kare matristir. Kendi transpozuna eşit olan $\mathbf{R} = \mathbf{R}^T$ kare matrislere *simetrik* matris denir. Kompleks değerler içeren bir kare matrisin simetrik olabilmesi için, kendi kompleks eşleniğinin transpozuna, yani Hermitian transpozuna eşit olması gerekir. Bu koşulu sağlayan kare matrislere *Hermitian* matris denir. Öyleyse, $\mathbf{R}$ ilişki matrisinin Hermitian özelliği

$$\mathbf{R}^H = \mathbf{R} \quad (2.13)$$

şeklinde açıklanabilir. $\mathbf{R}$ matrisinin bu özelliğini göstermenin diğer bir yolu da,

$$r(-k) = r^*(k) \quad (2.14)$$

yazmaktır. $r(k)$, özilişki fonksiyonunun k zaman farkı için aldığı değerdir. Buna göre, geniş anlamda durağan bir sürecin $\mathbf{R}$ ilişki matrisinin tanımlanabilmesi için,

$r(k)$ özilişki fonksiyonunun $k = 0, 1, 2, \dots, M-1$ zaman farkları için aldığı M adet değer bilmesi gerekir. Böylece, (2.12) ile verilen açılım tekrardan

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & \dots & r(M-1) \\ r^*(1) & r(0) & \dots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \dots & r(0) \end{bmatrix} \quad (2.15)$$

şeklinde yazılabilir. Rasgele sürece ait gözlemlerin hepsinin *gerçel değerli* olması, özel bir durum yaratır. Çünkü bu durumda, $r(k)$ özilişki fonksiyonu her k değeri için gerçel değerli olacağından, $\mathbf{R}$ ilişki matrisi de *Hermitian* matris yerine *simetrik* matris olmaktadır.

Özellik – 2 Durağan ayrık zamanlı rasgele süreçlerin ilişki matrisleri *Toeplitz* olmaktadır.

Sadece bir kare matris *Toeplitz* olabilir. Bir kare matrisin *Toeplitz* olması için, aynı diyagonal üzerinde bulunan tüm elemanlarının birbirine eşit olması gerekir. (2.15) ile verilen açılımdan da görülebileceği gibi, ana diyagonaldeki tüm elemanlar $r(0)$ değerine eşit olurken, ana diyagonalin bir üstündeki diyagonalde bulunan tüm elemanlar $r(1)$ değerine ve ana diyagonalin bir altındaki diyagonalde bulunan tüm elemanlarsa $r(-1)$ değerine eşit olmaktadır. Diğer diyagonallerdeki elemanlar da benzer şekilde olduğuna göre, $\mathbf{R}$ ilişki matrisi *Toeplitz* demektir.

Öte yandan, ilişki matrisinin *Toeplitz* özelliğinin, $\mathbf{u}(n)$ gözlem vektörü ile gösterilen ayrık zamanlı rasgele sürecin geniş anlamda durağan kabul edilmesinin bir sonucu olduğuna dikkat edilmelidir. Öyleyse, geniş anlamda durağan olan her ayrık zamanlı rasgele sürecin ilişki matrisi *Toeplitz*'dir ve bunun tersi de (ilişki matrisi *Toeplitz* olan her ayrık zamanlı rasgele süreç geniş anlamda durağandır) doğrudur.

Özellik – 3 Durağan ayrık zamanlı rasgele süreçlerin ilişki matrisleri, *her zaman negatif olmayan belirli (always nonnegative definite)* ve *hemen hemen her zaman pozitif belirli (almost always positive definite)* olmaktadır.

$\mathbf{x}$ vektörü sıfırdan farklı, kompleks değerli ve $M \times 1$ boyutlu keyfi bir vektör; y ise, $\mathbf{u}(n)$ gözlem vektörü ile $\mathbf{x}$ vektörünün *iç çarpımı* (*inner product*) olarak tanımlı skaler bir rastlantı değişkeni olsun.

$$y = \mathbf{x}^H \mathbf{u}(n)$$

Her iki tarafın Hermitian transpozu alındığında,

$$y^* = \mathbf{u}^H(n) \mathbf{x}$$

elde edilir. Burada y bir skaler olduğu için, Hermitian transpozu kompleks eşleniğine eşit olmaktadır. y rastlantı değişkeninin karesel beklendik değeri

$$\begin{aligned} E[|y|^2] &= E[y y^*] \\ &= E[\mathbf{x}^H \mathbf{u}(n) \mathbf{u}^H(n) \mathbf{x}] \\ &= \mathbf{x}^H E[\mathbf{u}(n) \mathbf{u}^H(n)] \mathbf{x} \\ &= \mathbf{x}^H \mathbf{R} \mathbf{x} \end{aligned}$$

şeklinde bulunur. $\mathbf{R}$, (2.11) eşitliğinde tanımlanan ilişki matrisidir. $\mathbf{x}^H \mathbf{R} \mathbf{x}$ ifadesi ise, *Hermitian formu* (*Hermitian form*) olarak adlandırılır.

$$E[|y|^2] \geq 0$$

olduğuna göre,

$$\mathbf{x}^H \mathbf{R} \mathbf{x} \geq 0 \tag{2.16}$$

yazılabilir ve (2.16) koşulunu sağlayan bir Hermitian formunun *negatif olmayan belirli* (*nonnegative definite*) veya *pozitif yarı belirli* (*positive semidefinite*) olduğu söylenir. Öyleyse, geniş anlamda durağan süreçlerin ilişki matrislerinin her zaman için *negatif olmayan belirli* oldukları söylenebilir.

Hermitian formu, sıfırdan farklı her $\mathbf{x}$ vektörü için $\mathbf{x}^H \mathbf{R} \mathbf{x} > 0$ koşulunu sağlıyorsa, $\mathbf{R}$ ilişki matrisinin *pozitif belirli* (*positive definite*) olduğu söylenir. Bu koşul, $\mathbf{u}(n)$ gözlem vektörünün M adet elemanını oluşturan rastlantı değişkenleri arasında

herhangi bir *doğrusal bağımlılık* olmadıkça, geniş anlamda durağan her süreç için sağlanır. Böyle bir özel durum ancak, $\{u(n)\}$ süreci ($K \leq M$ olmak üzere) K tane sinüzoidin toplamından oluşuyor ise ortaya çıkar. Pratikte, bu ideal durumla o denli az karşılaşılır ki; $\mathbf{R}$ ilişki matrisi, başta da belirtildiği gibi, *hemen hemen her zaman pozitif belirli* (*almost always positive definite*) kabul edilir.

İlişki matrisinin pozitif belirli olması demek, determinantının ve bütün *ana minörlerinin* (*principal minors*) sıfırdan büyük olması demektir. Örneğin; $M = 2$ için

$$\begin{vmatrix} r(0) & r(1) \\ r^*(1) & r(0) \end{vmatrix} > 0$$

koşulunun ve $M = 3$ için

$$\begin{vmatrix} r(0) & r(1) \\ r^*(1) & r(0) \end{vmatrix} > 0$$

$$\begin{vmatrix} r(0) & r(2) \\ r^*(2) & r(0) \end{vmatrix} > 0$$

$$\begin{vmatrix} r(0) & r(1) & r(2) \\ r^*(1) & r(0) & r(1) \\ r^*(2) & r^*(1) & r(0) \end{vmatrix} > 0$$

koşullarının sağlanıyor olması demektir. Bu koşullardan çıkarılabilecek diğer bir anlam ise, ilişki matrisinin *tekil olmayan* (*nonsingular*) bir matris olduğudur.

Bir matris tekil olmayan bir matris ise, o matrisin tersi vardır; eğer tersi yok ise, o matris *tekil* (*singular*) matristir. Öyleyse, durağan ayrık zamanlı rasgele süreçlere ait ilişki matrislerinin, *hemen hemen her zaman tekil olmayan* (*almost always nonsingular*) birer matris oldukları söylenebilir.

Özellik – 4 Durağan ayrık zamanlı bir rasgele sürecin gözlem vektörünü oluşturan elemanların *geri yönde* (*backward*) düzenlenmesi, sürecin ilişki matrisi üzerinde transpoza eşdeğer bir tesir yaratır.

$\mathbf{u}^B(n)$, $\mathbf{u}(n)$ vektörünün elemanlarının *geri yönde* düzenlenmeleri sonucunda oluşan $M \times 1$ boyutlu vektör olsun. Bu vektör,

$$\mathbf{u}^B(n) = [u(n-M+1), u(n-M+2), \dots, u(n)] \quad (2.17)$$

şeklinde olacaktır. Üstel işaret B , vektörün geri yönde düzenlendiğini gösterir. $\mathbf{u}^B(n)$ vektörünün ilişki matrisi ise,

$$E[\mathbf{u}^B(n) \mathbf{u}^{BH}(n)] = \begin{bmatrix} r(0) & r(-1) & \dots & r(-M+1) \\ r(1) & r(0) & \dots & r(-M+2) \\ \vdots & \vdots & \ddots & \vdots \\ r(M-1) & r(M-2) & \dots & r(0) \end{bmatrix} \quad (2.18)$$

haline gelir. (2.18) ile verilen matris (2.12) ile verilen matrisle karşılaştırıldığında,

$$E[\mathbf{u}^B(n) \mathbf{u}^{BH}(n)] = \mathbf{R}^T \quad (2.19)$$

olduğu görülür.

Özellik – 5 Durağan ayrık zamanlı bir rasgele sürecin M ve $(M+1)$ adet gözlem değerine ait $\mathbf{R}_M$ ve $\mathbf{R}_{M+1}$ ilişki matrisleri arasında

$$\mathbf{R}_{M+1} = \begin{bmatrix} r(0) & | & \mathbf{r}^H \\ \hline \mathbf{r} & | & \mathbf{R}_M \end{bmatrix} \quad (2.20)$$

veya eşdeğeri olarak,

$$\mathbf{R}_{M+1} = \begin{bmatrix} \mathbf{R}_M & | & \mathbf{r}^{B*} \\ \hline \mathbf{r}^{B^T} & | & r(0) \end{bmatrix} \quad (2.21)$$

eşitlikleriyle verilen bir ilişki vardır. Burada $r(0)$, sürecin özilişki fonksiyonunun sıfır zaman farkı için aldığı değer olup, $\mathbf{r}^H$ ve $\mathbf{r}^{B^T}$ vektörleri

$$\mathbf{r}^H = [r(1), r(2), \dots, r(M)] \quad (2.22)$$

$$\mathbf{r}^{BT} = [r(-M), r(-M+1), \dots, r(-1)] \quad (2.23)$$

şeklinde olmaktadır.

Özellik – 5 tanımlanırken $\mathbf{R}$ ilişki matrisine alt indis olarak yazılan M ve $(M+1)$ sembolleri, bu matrisin yapılan gözlem sayısına ne derece bağımlı olduğunu göstermek amacıyla kullanılmıştır.

$\mathbf{R}_{M+1}$ matrisinin açık formu

$$\mathbf{R}_{M+1} = \begin{bmatrix} r(0) & | & r(1) & r(2) & \dots & r(M) \\ \hline r(-1) & | & r(0) & r(1) & \dots & r(M-1) \\ r(-2) & | & r(1) & r(0) & \dots & r(M-2) \\ \vdots & | & \vdots & \vdots & \ddots & \vdots \\ r(-M) & | & r(-M+1) & r(-M+2) & \dots & r(0) \end{bmatrix} \quad (2.24)$$

şeklinde parçalara ayrılıp (2.12), (2.14) ve (2.22) eşitlikleri (2.24) içinde yerlerine konulduğunda, (2.20) ile verilen sonuca ulaşılır. Bu sonuca göre, $(M+1)$ adet gözlem değerinden oluşan $\mathbf{u}(n)_{M+1}$ vektörünün açık formu

$$\mathbf{u}(n)_{M+1} = \begin{bmatrix} u(n) \\ \hline u(n-1) \\ \vdots \\ u(n-M) \end{bmatrix} = \begin{bmatrix} u(n) \\ \hline \mathbf{u}_M(n-1) \end{bmatrix} \quad (2.25)$$

şeklinde parçalara ayrılarak yazılabilir.

$\mathbf{R}_{M+1}$ matrisinin açık formu (2.26) eşitliğindeki gibi parçalara ayrılıp (2.12), (2.14) ve (2.23) eşitlikleri (2.26) içinde yerlerine konulduğunda, (2.21) ile verilen sonuca ulaşılır.

$$\mathbf{R}_{M+1} = \left[\begin{array}{cccc|c} r(0) & r(1) & \cdots & r(M-1) & r(M) \\ r(-1) & r(0) & \cdots & r(M-2) & r(M-1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r(-M+1) & r(-M+2) & \cdots & r(0) & r(1) \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ r(-M) & r(-M+1) & \cdots & r(-1) & r(0) \end{array} \right] \quad (2.26)$$

Bu ikinci sonuca göre, $\mathbf{u}(n)_{M+1}$ vektörünün açık formu

$$\mathbf{u}(n)_{M+1} = \left[\begin{array}{c} u(n) \\ \vdots \\ u(n-M+1) \\ \text{---} \\ u(n-M) \end{array} \right] = \left[\begin{array}{c} \mathbf{u}_M(n) \\ \text{---} \\ u(n-M) \end{array} \right] \quad (2.27)$$

şeklinde parçalara ayrılabilir.

2.3 Özdeğer Analizi

$\mathbf{R}$, $M \times 1$ boyutlu $\mathbf{u}(n)$ gözlem vektörü ile gösterilen durağan ayrık zamanlı bir rasgele sürece ait $M \times M$ boyutlu ilişki matrisi olsun. Bu matris genelde, kompleks değerli elemanlar da içerebilir. Burada bulmak istediğimiz, λ gibi bir sabit için

$$\mathbf{R} \mathbf{q} = \lambda \mathbf{q} \quad (2.28)$$

koşulunu sağlayacak $M \times 1$ boyutlu $\mathbf{q}$ vektörüdür. Bu koşul, $\mathbf{q}$ vektörünün $\mathbf{R}$ matrisi tarafından doğrusal olarak $\lambda \mathbf{q}$ vektörüne dönüştürülmesini göstermektedir. λ sabit bir değer olduğu için, yapılan bu doğrusal dönüşüm M boyutlu uzayda $\mathbf{q}$ vektörünün doğrultusunda herhangi bir değişikliğe sebep olmamaktadır. $M \times M$ boyutlu tipik bir $\mathbf{R}$ matris için bu özellikte M adet $\mathbf{q}$ vektörü bulunabilir. Bunu göstermek amacıyla, (2.28) eşitliği

$$(\mathbf{R} - \lambda \mathbf{I}) \mathbf{q} = 0 \quad (2.29)$$

formunda yeniden yazılır. $\mathbf{I}$, $\mathbf{I} = \text{diag}(1, 1, \dots, 1)$ şeklinde tanımlı $M \times M$ boyutlu

birim (identity) matris ve $\mathbf{0}$ ise, $M \times 1$ boyutlu sıfır vektördür. (2.29) eşitliğinde $\mathbf{q}$ vektörünün sıfırdan farklı çözümünün olabilmesi için, $(\mathbf{R}-\lambda\mathbf{I})$ matrisinin tekil matris olması veya başka bir deyişle, determinantının sıfır olması

$$\det(\mathbf{R} - \lambda\mathbf{I}) = 0 \quad (2.30)$$

zorunludur. (2.30) eşitliği açıldığı zaman, λ 'nın M . dereceden bir polinomu olduğu görülür; bu ise, (2.30) eşitliğinin M adet kökü var demektir. Bu durum karşısında (2.28) eşitliğinin, $\mathbf{q}$ vektörüne ait M adet çözümü olacaktır. Böylelikle, $M \times M$ boyutlu tipik bir $\mathbf{R}$ matris için M adet $\mathbf{q}$ vektörünün bulunabileceği gösterilmiş olur.

(2.30) eşitliği, $\mathbf{R}$ matrisinin *karakteristik denklemi (characteristic equation)* ve bu denklemin $\lambda_1, \lambda_2, \dots, \lambda_M$ ile gösterilen M adet kökü ise, $\mathbf{R}$ matrisinin *özdeğerleri (eigenvalues)* olarak adlandırılır. M adet özdeğerin olması, bunların her zaman için birbirinden farklı değerler alacağı anlamına gelmez. Eğer karakteristik denklemin katlı kökleri varsa, $\mathbf{R}$ ilişki matrisinin *dejenere (degenerate)* özdeğerlere sahip olduğu söylenir. Örnek olarak, *beyaz-gürültü (white-noise)* sürecinin

$$\mathbf{R} = \text{diag}(\sigma^2, \sigma^2, \dots, \sigma^2)$$

şeklinde tanımlanan $M \times M$ boyutlu $\mathbf{R}$ ilişki matrisi ele alınsın. σ^2 , sürecin tek bir örneğinin değişintisini göstermektedir ve $\mathbf{R}$ ilişki matrisinin, σ^2 değişintisine eşit M katlı tek bir dejenere özdeğeri vardır. Bu nedenle, $M \times 1$ boyutlu rasgele herhangi bir $\mathbf{q}$ vektör, bu özdeğerin özvektörü olarak nitelendirilebilir. Bu durum, beyaz gürültünün rasgele bir yapıya sahip olduğunun kanıtıdır. (2 x 2) gibi basit boyutlu matrisler dışında, bir matrisin özdeğerlerini hesaplamak için o matrisin karakteristik denkleminin köklerini bulmak, kaçınılması gereken zayıf yöntemler arasında yer almaktadır. Bunun yerine, özdeğer hesabı için geliştirilmiş daha iyi yöntemlerden faydalanılmalıdır. λ_i , $\mathbf{R}$ matrisine ait herhangi bir özdeğer ve $\mathbf{q}_i$ ise, bu özdeğer için

$$\mathbf{R}\mathbf{q}_i = \lambda_i \mathbf{q}_i \quad (2.31)$$

eşitliğini sağlayan sıfırdan farklı bir vektör olsun. Bu durumda $\mathbf{q}_i$ vektör, λ_i özdeğerinin bir *özvektörü* (*eigenvector*) olarak adlandırılır. Bir özvektör sadece tek bir özdeğere karşı düşebilirken, bir özdeğerin bir çok özvektörü olabilir. Örneğin; $\mathbf{q}_i$ vektör λ_i özdeğerine ilişkin bir özvektör ise, $a \neq 0$ şeklindeki herhangi bir skaler için $a\mathbf{q}_i$ vektör de λ_i özdeğerinin bir özvektörü olmaktadır. Bu nedenle, Hayes'e (1996) göre özvektörlerin $\|\mathbf{q}_i\| = 1$ ile gösterilen *birim norma* (*unit norm*) sahip olacak şekilde normalize edilmelerinde yarar vardır. Vektörlerle gerçekleştirilen pek çok işlemde, vektör *büyüklüğü* (*magnitude*) kullanılmaktadır. *Euclidean normu*, M boyutlu bir vektörün büyüklüğünü bulmak için kullanılan norm çeşitlerinden biridir ve

$$\|\mathbf{x}\| = \left\{ \sum_{i=1}^M |x_i|^2 \right\}^{1/2} \quad (2.32)$$

eşitliğiyle hesaplanır (Hayes, 1996).

Geometrik anlamda ele alındığında Euclidean normu, o vektörün uzunluğuna eşit olmaktadır. Bu nedenle Euclidean normu, *Euclidean uzunluğu* olarak da adlandırılabilir. $\mathbf{x}^T = [x_1, x_2, \dots, x_M]$ ve $\mathbf{y}^T = [y_1, y_2, \dots, y_M]$ gibi iki vektör arasındaki uzaklık

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\| = \left\{ \sum_{i=1}^M |x_i - y_i|^2 \right\}^{1/2}$$

şeklinde elde edilirken, bu vektörlerin iç çarpımı

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^H \mathbf{y} = \sum_{i=1}^M x_i^* y_i \quad (2.33)$$

eşitliğiyle tanımlı skaler bir değerdir. İç çarpım, iki vektör arasındaki geometrik ilişkiyi tanımlar. Bu geometrik ilişki ise,

$$\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\| \|\mathbf{y}\| \cos \theta \quad (2.34)$$

eşitliği ile verilir. θ , iki vektör arasındaki açıdır. Buna göre, sıfırdan farklı $\mathbf{x}$ ve $\mathbf{y}$

vektörlerinin iç çarpımı sıfır $\langle \mathbf{x}, \mathbf{y} \rangle = 0$ ise, bu vektörlerin birbirine *dik*, yani *orthogonal* (*orthogonal*) oldukları söylenir. Hem birbirine dik ve hem de birim norma sahip olan vektörlere ise, *orthonormal* (*orthonormal*) denir.

İç çarpımın kullanımlarından biri de, ötelemeyle değişmeyen doğrusal filtrelerin çıkışının elde edilmesidir. Örneğin; $(M-1)$. dereceden bir FIR filtrenin, birim örnek yanıtı $h(n)$ ve girişi de $x(n)$ olsun. Filtre çıkışı $y(n)$, $h(n)$ ve $x(n)$ 'in

$$y(n) = \sum_{k=0}^{M-1} h(k) x(n-k) \quad (2.35)$$

şeklindeki konvolüsyonu ile elde edilir. Bunun yerine, $x(n)$ giriş işareti için $\mathbf{x}^T(n) = [x(n), x(n-1), \dots, x(n-M+1)]$ ve $h(n)$ içinse $\mathbf{h}^T = [h(0), h(1), \dots, h(M-1)]$ vektör formu kullanılarak, (2.35) eşitliğinin bu iki vektörün iç çarpımı

$$y(n) = \mathbf{h}^T \mathbf{x} \quad (2.36)$$

şeklinde yazılması da mümkündür (Hayes, 1996). $\mathbf{x}$ ve $\mathbf{h}$ vektörlerinin gerçel değerlerden oluştuğu kabul edilerek, iç çarpım ifadesinde Hermitian transpoz (H) yerine normal transpoz (T) kullanılmıştır.

2.3.1 Özdeğer ve özvektörlerin özellikleri

Bu bölümde, durağan ayrık zamanlı bir rasgele sürece ait ilişki matrisinin özdeğer ve özvektörlerinin çeşitli özellikleri incelenecektir. Bu özellikler direkt olarak, $\mathbf{R}$ ilişki matrisinin Bölüm 2.2'de açıklanan *negatif olmayan belirlilik* ve *Hermitian* özelliklerinden kaynaklanmaktadır.

Özellik – 1 $\lambda_1, \lambda_2, \dots, \lambda_M$ $\mathbf{R}$ ilişki matrisinin özdeğerlerini gösterebilir. Her $k > 0$ tamsayısı için $\mathbf{R}^k$ matrisinin özdeğerleri $\lambda_1^k, \lambda_2^k, \dots, \lambda_M^k$ olur.

(2.28) eşitliğinin her iki tarafı $\mathbf{R}$ matrisi ile k defa çarpıldığında, $\mathbf{R}^k \mathbf{q} = \lambda^k \mathbf{q}$ ifadesi elde edilir ve bu ifadeye göre, $\mathbf{R}$ matrisinin her özvektörü aynı zamanda $\mathbf{R}^k$

matrisinin de bir özvektörü olmaktadır. Eğer $\mathbf{R}$ matrisinin tersi mevcut ise, $k = -1$ özel durumu oluşur ve $\mathbf{R}^{-1}$ ters matrisinin özdeğerleri $\lambda_1^{-1}, \lambda_2^{-1}, \dots, \lambda_M^{-1}$ olur.

Özellik – 2 $\lambda_1, \lambda_2, \dots, \lambda_M$ $\mathbf{R}$ ilişki matrisinin birbirinden farklı özdeğerleri ve $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ ise, bu özdeğerlere karşı düşen sıfırdan farklı özvektörler olsun. Bu durumda $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ özvektörleri, *doğrusal bağımsız* olur.

$$\sum_{i=1}^M v_i \mathbf{q}_i = \mathbf{0} \quad (2.37)$$

eşitliğini sağlayan ve hepsi birden sıfır olmayan $v_1, v_2, \dots, v_M$ skalerleri mevcutsa, $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ özvektörlerine *doğrusal bağımlı*; mevcut değilse, *doğrusal bağımsız* denir. (2.37) eşitliğinin belli v_i skalerleri için sağlandığı kabul edilsin. Bu eşitliğin $\mathbf{R}$ matrisiyle tekrarlı çarpımı ve (2.28) eşitliğinin kullanılması sonucunda,

$$\sum_{i=1}^M v_i \lambda_i^{k-1} \mathbf{q}_i = \mathbf{0} \quad , \quad k = 1, 2, \dots, M \quad (2.38)$$

şeklinde her biri farklı bir k değerine karşı gelmekte olan toplam M adetlik bir denklem takımı elde edilir. Bu denklem takımını tek bir matris formunda

$$[v_1 \mathbf{q}_1, v_2 \mathbf{q}_2, \dots, v_M \mathbf{q}_M] \mathbf{S} = \mathbf{0} \quad (2.39)$$

yazmak mümkündür. Burada $\mathbf{S}$, *Vandermonde* matris olarak adlandırılır (Strang, 1980).

$$\mathbf{S} = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{M-1} \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \lambda_2^{M-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \lambda_M & \lambda_M^2 & \dots & \lambda_M^{M-1} \end{bmatrix}$$

$i = 1, 2, \dots, M$ olmak üzere, λ_i değerleri birbirinden farklı olduğu zaman: $\mathbf{S}$ matrisi tekil olmayan bir matris olacağından, (2.39) eşitliği $\mathbf{S}^{-1}$ ters matrisiyle sağdan çarpılabilir ve bu çarpımdan, $[v_1 \mathbf{q}_1, v_2 \mathbf{q}_2, \dots, v_M \mathbf{q}_M] = \mathbf{0}$ sonucu elde edilir. $\mathbf{q}_i$ özvektörleri sıfırdan farklı olduğuna göre, her bir sütunu $v_i \mathbf{q}_i = \mathbf{0}$ şeklinde olan bu sonucun doğru olabilmesi için, başta yapılan kabule ters düşülmesi, yani v_i

skalerlerinin hepsinin birden sıfır olarak (2.37) eşitliğini sağlamaması gerekir. Öyleyse, birbirlerinden farklı değerler almış $\lambda_1, \lambda_2, \dots, \lambda_M$ özdeğerlerine karşı düşen $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ özvektörleri, asla doğrusal bağımlı olamaz; tam aksine, her zaman için doğrusal bağımsızdır.

Doğrusal bağımsız $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ özvektörlerinin kendileri ile aynı boyutlu rasgele bir $\mathbf{w}$ vektörünün gösteriminde kaynak olarak kullanılması, doğrusal bağımsızlık özelliğinden faydalanma yollarının önemli bir tanesidir. $v_1, v_2, \dots, v_M$ sabit değerler olmak üzere, rasgele bir $\mathbf{w}$ vektörünü

$$\mathbf{w} = \sum_{i=1}^M v_i \mathbf{q}_i \quad (2.40)$$

şeklinde $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ özvektörlerinin doğrusal toplamı olarak ifade edilebiliriz. $\mathbf{R}$ matrisiyle soldan çarpılarak, $\mathbf{w}$ vektörüne doğrusal dönüşüm uygulandığı varsayılın. Bu durumda,

$$\mathbf{R} \mathbf{w} = \sum_{i=1}^M v_i \mathbf{R} \mathbf{q}_i \quad (2.41)$$

olacaktır. Tanım gereği $\mathbf{R} \mathbf{q}_i = \lambda_i \mathbf{q}_i$ olduğundan, bu doğrusal dönüşüm ifadesi

$$\mathbf{R} \mathbf{w} = \sum_{i=1}^M v_i \lambda_i \mathbf{q}_i \quad (2.42)$$

biçiminde de yazılabilir. Buna göre, rasgele bir vektöre doğrusal dönüşüm uygulandığı zaman, özvektörler birbirinden bağımsız olmaya devam etmekte ve yapılan dönüşümün tesiri, her bir özvektörün kendisine karşı düşen özdeğer ile çarpılması şeklinde ortaya çıkmaktadır.

Özellik – 3 Hermitian matrislerin bütün özdeğerleri gerçeldir. $\mathbf{R}$ ilişki matrisi de bir Hermitian matris olduğuna göre, özdeğerlerinin hepsi $\lambda_1, \lambda_2, \dots, \lambda_M$ gerçel olacaktır.

Bunu ispatlamak üzere, $\mathbf{R}$ matrisinin i . özdeğeri için yazılan (2.31) eşitliğinin iki tarafı $\mathbf{q}_i^H$ vektörüyle soldan çarpılır.

$$\mathbf{q}_i^H \mathbf{R} \mathbf{q}_i = \lambda_i \mathbf{q}_i^H \mathbf{q}_i \quad (2.43)$$

Her iki tarafın Hermitian transpozunu alındığında, $\mathbf{R}^H = \mathbf{R}$ özelliğinden dolayı

$$\mathbf{q}_i^H \mathbf{R} \mathbf{q}_i = \lambda_i^* \mathbf{q}_i^H \mathbf{q}_i \quad (2.44)$$

elde edilir. (2.43) ve (2.44) eşitliklerinin sol tarafları eşit olduğuna göre, sağ tarafları da eşit olmak $\lambda_i = \lambda_i^*$ zorundadır.

Özellik – 4 Pozitif özdeğerlere sahip Hermitian matrisler pozitif belirli, negatif olmayan özdeğerlere sahip Hermitian matrisler ise negatif olmayan belirli olurlar. Buna göre $\mathbf{R}$ ilişki matrisi, bütün özdeğerleri $\lambda_i > 0$ oluyor ise pozitif belirli, $\lambda_i \geq 0$ oluyor ise de negatif olmayan belirli olur.

(2.43) eşitliğinde her iki tarafı $\mathbf{q}_i^H \mathbf{q}_i$ iç çarpımına bölerek, Özellik – 4'te söylenenlerin doğruluğu kolayca ispatlanabilir.

$$\lambda_i = \frac{\mathbf{q}_i^H \mathbf{R} \mathbf{q}_i}{\mathbf{q}_i^H \mathbf{q}_i}, \quad i = 1, 2, \dots, M \quad (2.45)$$

$\mathbf{q}_i^H \mathbf{q}_i$ iç çarpımı, $\mathbf{q}_i$ vektörünün Euclidean normunun karesine eşit olduğundan, daima pozitif değerli bir skalerdir. Bu nedenle, geriye kalan λ_i ve $\mathbf{R}$ değerleri, birbirinin nasıl olacağını belirler. Her i için $\lambda_i \geq 0$ oluyor ise, $\mathbf{R} \geq 0$; $\lambda_i > 0$ oluyor ise, $\mathbf{R} > 0$ olacaktır. Tek bir özdeğerin bile sıfıra eşit olması, matrisin negatif olmayan belirli olmasına ve dolayısıyla, tersinin mevcut olmamasına sebep olur. Öyleyse, pozitif belirli matrisler, her zaman için tekil olmayan (tersi olan) matrislerdir. Bunun doğruluğunu, bir matrisin determinantı ile özdeğerleri arasındaki ilişkiyi veren

$$\det(\mathbf{R}) = \prod_{i=1}^M \lambda_i \quad (2.46)$$

eşitliğinden de rahatlıkla görmek mümkündür (Hayes, 1996). $\mathbf{q}_i^H \mathbf{R} \mathbf{q}_i$ Hermitian formunun $\mathbf{q}_i^H \mathbf{q}_i$ iç çarpımına oranı, yani (2.45) eşitliğinin sağ tarafı, *Rayleigh bölümü* (*Rayleigh quotient*) olarak adlandırılır. Buradan, ilişki matrisine ait herhangi bir özdeğerin, o özdeğere karşı düşen özvektörün Rayleigh bölümüne eşit olduğu görülür.

Özellik – 5 Bir Hermitian matrisin farklı özdeğerlerine $\lambda_1, \lambda_2, \dots, \lambda_M$ karşı düşen özvektörleri $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ birbirine dik (orthogonal) olmaktadır. Diğer bir ifadeyle, $\lambda_i \neq \lambda_j$ için $\langle \mathbf{q}_i, \mathbf{q}_j \rangle = \mathbf{q}_i^H \mathbf{q}_j = 0$ olmaktadır.

$\mathbf{q}_i$ ve $\mathbf{q}_j$, $\mathbf{R}$ ilişki matrisinin λ_i ve λ_j şeklinde iki farklı özdeğerine karşı düşen özvektörleri olsun. (2.28) eşitliği, $\mathbf{q}_i$ için $\mathbf{R} \mathbf{q}_i = \lambda_i \mathbf{q}_i$ ve $\mathbf{q}_j$ içinse $\mathbf{R} \mathbf{q}_j = \lambda_j \mathbf{q}_j$ halini alır. $\mathbf{q}_i$ için yazılan ifadeyi $\mathbf{q}_j^H$ vektörüyle ve $\mathbf{q}_j$ için yazılan ifadeyi ise $\mathbf{q}_i^H$ vektörüyle soldan çarparak,

$$\mathbf{q}_j^H \mathbf{R} \mathbf{q}_i = \lambda_i \mathbf{q}_j^H \mathbf{q}_i \quad (2.47)$$

$$\mathbf{q}_i^H \mathbf{R} \mathbf{q}_j = \lambda_j \mathbf{q}_i^H \mathbf{q}_j \quad (2.48)$$

ifadeleri elde edilir. (2.48) eşitliğinin her iki tarafının Hermitian transpozunu için Özellik – 3'ten ve $\mathbf{R} = \mathbf{R}^H$ olmasından yararlanılarak,

$$\mathbf{q}_j^H \mathbf{R} \mathbf{q}_i = \lambda_j \mathbf{q}_j^H \mathbf{q}_i \quad (2.49)$$

eşitliğine ulaşılır. (2.47) eşitliğinden (2.49) eşitliği çıkarıldığında,

$$(\lambda_i - \lambda_j) \mathbf{q}_j^H \mathbf{q}_i = 0 \quad (2.50)$$

sonucuna ulaşılır. Başlangıçta, özdeğerlerin birbirinden farklı oldukları $\lambda_i \neq \lambda_j$ kabul edildiği için, (2.50) eşitliği sadece $\mathbf{q}_j^H \mathbf{q}_i = 0$ ile sağlanır. Buna göre, özdeğerleri farklı olan özvektörler birbirine dik olmaktadır. Öyleyse Özellik – 5,

$$\mathbf{q}_i^H \mathbf{q}_j = 0 \quad , \quad i \neq j \quad (2.51)$$

eşitliği ile ifade edilebilir.

Özellik – 6 Birimsel Benzerlik Dönüşümü (Unitary Similarity Transformation):

$\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ $M \times M$ boyutlu $\mathbf{R}$ matrisinin birbirlerinden farklı $\lambda_1, \lambda_2, \dots, \lambda_M$ özdeğerlerine karşı düşen (doğrusal bağımsız) özvektörler olsun. Ayrıca, sütunları

$$\mathbf{q}_i^H \mathbf{q}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

şeklinde orthonormal vektörlerden oluşan $M \times M$ boyutlu $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M]$ matrisi ve yine $M \times M$ boyutlu $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M)$ diyagonal matrisi tanımlansın. Bu durumda $\mathbf{R}$ ilişki matrisi, $\mathbf{Q}^H \mathbf{R} \mathbf{Q} = \mathbf{\Lambda}$ şeklinde diyagonalleştirilebilir.

$i = 1, 2, \dots, M$ için $\mathbf{q}_i^H \mathbf{q}_i = 1$ olması, her bir özvektörün 1 uzunluğuna normalize edilmesini gerektirir. Bir $\mathbf{q}_i$ özvektörünün *karesel uzunluğu* veya diğer adıyla *karesel normu*, daha önce de belirtildiği gibi, $\mathbf{q}_i^H \mathbf{q}_i$ iç çarpımı ile bulunur. $\mathbf{q}_i^H \mathbf{q}_i = 1$ koşulu Özellik – 5'te verilen $\mathbf{q}_i^H \mathbf{q}_j = 0$ ($i \neq j$) ortogonallik koşulu ile birlikte sağlandığı zaman,

$$\mathbf{q}_i^H \mathbf{q}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (2.52)$$

$\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M$ özvektörlerine *orthonormal* vektör takımı denir. $M \times M$ boyutlu herhangi bir $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_M]$ matrisin ortogonal olabilmesi için, sütunlarının (veya satırlarının)

$$\mathbf{q}_i^T \mathbf{q}_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

orthonormal olması gerekir. Bu durumda $\mathbf{Q}$ matrisi, ortogonal bir matristir ve her

orthogonal matris gibi, $Q^T Q = I$ eşitliğini sağlar; dolayısıyla da, tersi transpozuna eşit ($Q^{-1} = Q^T$) olur. Q matrisi kompleks değerler içerdiği zaman, sütunları (veya satırları) orthonormal oluyor ise, yani (2.52) eşitliği sağlanıyor ise, Q matrisi de $Q^H Q = I$ eşitliğini sağlar ve bu eşitliği sağlayan matrislere, *birimsel (unitary)* matris denir. Birimsel bir matrisin tersi, kendisinin Hermitian transpozuna ($Q^{-1} = Q^H$) eşittir.

Tanım gereği, $q_1, q_2, \dots, q_M$ özvektörleri $i = 1, 2, \dots, M$ için $R q_i = \lambda_i q_i$ eşitliğini sağlarlar. Bu eşitlikle elde edilen M adet denklem, Q ve Λ matrisleri kullanılarak tek bir matris eşitliği

$$RQ = Q\Lambda \quad (2.53)$$

halinde yazılabilir. Yukarıda da bahsedildiği gibi, özvektörlerin (2.52) eşitliğini sağlamasından, yani orthonormal olmalarından dolayı, Q matris birimsel (unitary) bir matris olur ve tersi, $Q^{-1} = Q^H$ ile elde edilir. Tersinin bu şekilde elde edilmesi, Q matrisin her zaman için tersi var olan (tekil olmayan) bir matris olduğunu da gösterir. (2.53) eşitliği Q^H matrisiyle sol taraftan çarpılıp $Q^{-1} = Q^H$ özelliği de kullanıldığında, başta da belirtilen

$$Q^H RQ = \Lambda \quad (2.54)$$

eşitliğine ulaşılır. Bu dönüşüm, *birimsel benzerlik dönüşümü (unitary similarity transformation)* olarak adlandırılır. Böylelikle, önemli bir sonuç açıklanmış olur: R ilişki matrisi birimsel benzerlik dönüşümü ile diyagonalleştirilebilir. R matrisinin özvektörleri, bu matrisi diyagonalleştirmede kullanılan Q matrisinin sütunlarını oluştururken; özdeğerleri ise, sonuç olarak elde edilen Λ diyagonal matrisinin diyagonal elemanlarını oluşturmaktadır.

(2.53) eşitliği Q^{-1} matrisiyle sağ taraftan çarpılıp $Q^{-1} = Q^H$ özelliği de kullanıldığında, (2.54) eşitliğinden başka,

$$\mathbf{R} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H = \sum_{i=1}^M \lambda_i \mathbf{q}_i \mathbf{q}_i^H \quad (2.55)$$

eşitliğini de elde etmek mümkündür. $\mathbf{q}_i \mathbf{q}_i^H$ terimi, i . özvektörün kendisiyle dış çarpımıdır. M ise, $\mathbf{R}$ ilişki matrisinin boyutunu gösterir. (2.55) eşitliği; geniş anlamda durağan bir rasgele sürece ait ilişki matrisinin, kendilerine karşı düşen λ_i özdeğerleriyle ağırlaştırılmış $\mathbf{q}_i \mathbf{q}_i^H$ dış çarpımlarının doğrusal toplamına eşit olduğunu gösterir. (2.55) ile verilen ayrıştırma, *spektral teorem (spectral theorem)* olarak adlandırılır.

$\mathbf{R}$ matrisinin tekil olmayan Hermitian bir matris olması koşulu ile, bu matrisin tersi spektral teoremden,

$$\mathbf{R}^{-1} = (\mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H)^{-1} = (\mathbf{Q}^H)^{-1} \mathbf{\Lambda}^{-1} \mathbf{Q}^{-1} = \mathbf{Q} \mathbf{\Lambda}^{-1} \mathbf{Q}^H = \sum_{i=1}^M \frac{1}{\lambda_i} \mathbf{q}_i \mathbf{q}_i^H$$

eşitliğiyle bulunabilir. $\mathbf{R}$ matrisinin tersinin var olması, $\lambda_i \neq 0$ olmasını garanti eder. Bu nedenle, bu toplam ancak, $\mathbf{R}$ matrisi yukarıda belirtilen koşulu sağladığı sürece geçerlidir.

Özellik – 7 $\lambda_1, \lambda_2, \dots, \lambda_M$ $M \times M$ boyutlu $\mathbf{R}$ matrisinin özdeğerleri olsun. Bu özdeğerlerin toplamı, $\mathbf{R}$ matrisinin *izine (trace)* eşittir.

Ancak bir kare matrisin izi olabilir ve bu değer, o kare matrisin ana diyagonalinde bulunan elemanların toplamı olarak hesaplanır. $M \times N$ boyutlu bir $\mathbf{A}$ matris ve $N \times M$ boyutlu bir $\mathbf{B}$ matris ele alınsın. $\mathbf{AB}$ çarpımının izi, $\mathbf{BA}$ çarpımının izine eşittir $\text{tr}[\mathbf{AB}] = \text{tr}[\mathbf{BA}]$. (2.54) eşitliğinin her iki tarafının izi alındığında,

$$\text{tr}[\mathbf{Q}^H \mathbf{R} \mathbf{Q}] = \text{tr}[\mathbf{\Lambda}] \quad (2.56)$$

olur. $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M)$ diyagonal matrisinin izi,

$$\text{tr}[\mathbf{\Lambda}] = \sum_{i=1}^M \lambda_i \quad (2.57)$$

şeklinde $\mathbf{R}$ matrisinin özdeğerleri toplamına eşittir. $\text{tr}[\mathbf{AB}] = \text{tr}[\mathbf{BA}]$ özelliğinden dolayı, $\text{tr}[\mathbf{Q}^H \mathbf{R} \mathbf{Q}] = \text{tr}[\mathbf{R} \mathbf{Q} \mathbf{Q}^H]$ yazılabilir. $\mathbf{I}$, birim (identity) matris olmak üzere $\mathbf{Q}$ matris için $\mathbf{Q} \mathbf{Q}^H = \mathbf{I}$ olduğundan, $\text{tr}[\mathbf{Q}^H \mathbf{R} \mathbf{Q}] = \text{tr}[\mathbf{R}]$ haline gelir. Bu durumda (2.56) eşitliği tekrardan,

$$\text{tr}[\mathbf{R}] = \sum_{i=1}^M \lambda_i \quad (2.58)$$

şeklinde yazılabilir. Böylelikle, $\mathbf{R}$ ilişki matrisinin izinin kendi özdeğerlerinin toplamına eşit olduğu ispatlanmış olur. Bu ispatı yapmak için her ne kadar $\mathbf{R}$ matrisinin dejenere olmayan (birbirlerinden farklı) özdeğerlere sahip bir Hermitian matris olduğu varsayılmış olsa da, (2.58) eşitliği her kare matris için geçerlidir.

Özellik – 8 $\mathbf{R}$ ilişki matrisinin en büyük özdeğerinin en küçük özdeğerine oranı büyük ise, $\mathbf{R}$ matrisine *kötü durumda (ill conditioned)* denir.

Kötü durum, matrisin bir veya daha fazla özdeğerinin sıfıra yakın değer almasından kaynaklanır. Bir $\mathbf{A}$ matrisinin kötü durumunu nicelik olarak açıklayabilmek için, Strang (1980) tarafından tanımlanmış olan *koşul sayısı (condition number)* kullanılır.

$$\chi(\mathbf{A}) = \|\mathbf{A}\| \|\mathbf{A}^{-1}\| \quad (2.59)$$

$\|\mathbf{A}\|$ değeri, $\mathbf{A}$ matrisinin ve $\|\mathbf{A}^{-1}\|$ ise, $\mathbf{A}$ matrisin tersinin normudur. Daha önce Bölüm 2.3'te vektörler için olduğu gibi, bir matris için de norm, o matrisin büyüklüğünü ifade etmede kullanılan bir değerdir ve bir matrisin normunu bulmak için kullanılabilecek çeşitli yollar vardır. $\|\mathbf{A}\|$ normunu bulmak için burada izlenecek yol ise, $\mathbf{A}^H \mathbf{A}$ matris çarpımının en büyük özdeğerinin karekökü olarak tanımlanan *spektral normdur*.

$$\|\mathbf{A}\| = (\mathbf{A}^H \mathbf{A} \text{ matris çarpımının en büyük özdeğeri})^{1/2} \quad (2.60)$$

$\mathbf{R}$ ilişki matrisinin Hermitian ($\mathbf{R} = \mathbf{R}^H$) olmasından dolayı, $\mathbf{R}^H \mathbf{R} = \mathbf{R}^2$ olacaktır. Eğer $\mathbf{R}$ matrisinin en büyük özdeğeri $\lambda_{\max}$ ise, $\mathbf{R}^2$ matrisinin en büyük özdeğeri $\lambda_{\max}^2$ olacaktır (Özellik – 1). Bu durumda (2.60) eşitliğine göre, $\mathbf{R}$ matrisinin

spektral normu

$$\| \mathbf{R} \|_s = \lambda_{\max} \quad (2.61)$$

ve $\mathbf{R}$ matrisin tersinin spektral normu da

$$\| \mathbf{R}^{-1} \|_s = \frac{1}{\lambda_{\min}} \quad (2.62)$$

olacaktır. $\lambda_{\min}$, $\mathbf{R}$ matrisinin en küçük özdeğeridir ve Özellik – 1’e göre $1/\lambda_{\min}$, $\mathbf{R}^{-1}$ matrisinin en büyük özdeğeridir. (2.61) ve (2.62) eşitlikleri (2.59) eşitliğinde kullanıldığında, $\mathbf{R}$ matrisinin koşul sayısı

$$\chi(\mathbf{R}) = \| \mathbf{R} \| \| \mathbf{R}^{-1} \| = \frac{\lambda_{\max}}{\lambda_{\min}} \quad (2.63)$$

olarak elde edilir. Bu orana genellikle, *özdeğer yayılımı (eigenvalue spread)* veya *özdeğer oranı (eigenvalue ratio)* denir. Ayrıca, özdeğer yayılımının her zaman için $\chi(\mathbf{R}) \geq 1$ koşulunu sağladığı da belirtilmelidir.

$r(0)$, $\mathbf{R}$ matrisinin en büyük elemanıdır. $\mathbf{R}$ matrisinin, $r(0) = 1$ olması sağlanacak şekilde normalize edildiği varsayılın. Bu koşul altında, $\mathbf{R}$ matrisinin özdeğer yayılımı büyük oluyor ise, $\mathbf{R}^{-1}$ matrisi çok büyük elemanlar içeriyor demektir. Böyle bir davranış, $\mathbf{R}^{-1}$ matrisini kullanmayı gerektiren sistem denklemlerinin çözümünde soruna yol açar. Bu durumdaki bir $\mathbf{R}$ matrisine ise, *kötü durumlu* denir.

3. WIENER FİLTRELER

Geçmiş Gauss'a kadar dayanan *doğrusal filtreleme problemi*, birbirlerinden bağımsız yaptıkları çalışmalar sonucunda Kolmogorov (1941) ve Wiener (1942) tarafından yeniden formülize edilmiştir. Kolmogorov, ayrık zaman problemini çalışmış ve bu problemi *Wold ayrıştırması (Wold decomposition)* kullanarak çözmüştür. Diğer yandan Wiener, sürekli zaman problemini çalışmış ve oldukça ünlü olan *Wiener-Hopf integral denklemlerini* elde etmiştir. Bu denklemler, ilgilenilen işaret süreçlerinin ilişki fonksiyonlarına ait bir ön bilgi gerek duyar. Çok basit problemlerin dışındaki problemler için bu denklemlerin çözümü oldukça zordur. Sürekli zamanda ele alınan bu integral denklemlerinin ayrık zamandaki eşdeğeri, *normal denklemleri* olarak bilinir. Normal denklemlerinin ayrık zaman çözümü ile Wiener-Hopf integral denklemlerinin sürekli zaman çözümünün her ikisi de, *Wiener filtreler* ortak ismiyle adlandırılır.

Bu bölümde, Wiener filtrelerin ayrık zaman formülleri, *kompleks değerli* zaman serileri genel durumu için impuls yanıtlarına göre belirlenen filtreler kullanılarak çalışılacaktır. Kompleks değerli zaman serilerinin kullanılmasının nedeni, radar ve sonar gibi bir çok pratik uygulamada ilgilenilen temel bant işaretin kompleks formda olmasıdır. Temel bant, bilgi kaynağı tarafından üretilen orijinal işaretin frekanslarını içeren frekans bandına verilen isimdir. Genel durum olarak kompleks değerli zaman serilerinin kullanılmasından dolayı, gerçel değerli zaman serileri bu genel durumun özel bir hali olur.

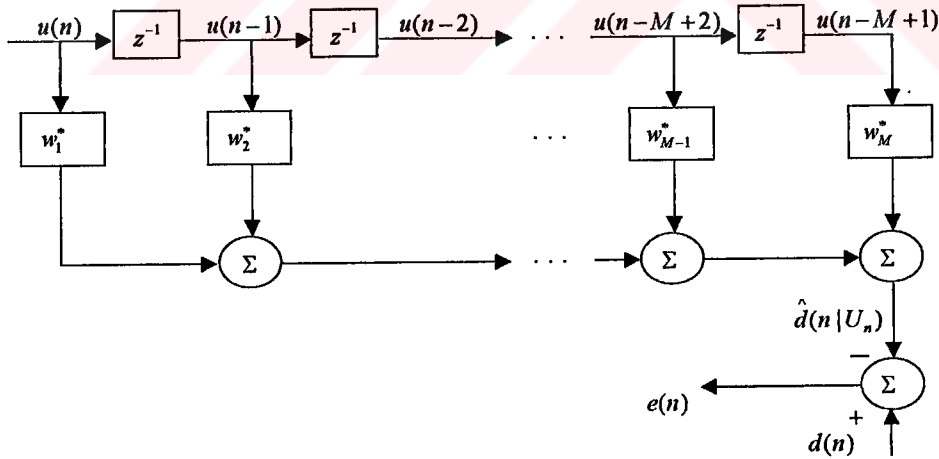
Buraya kadar, kullanılacak filtre üzerinde *doğrusal* ve *ayrık zamanlı* olması şeklinde iki kısıtlama yapıldı. Filtrenin doğrusal oluşu matematiksel analizi kolaylaştırırken, ayrık zamanda çalışması ise filtrenin sayısal devrelerde kullanılabilmesini sağlar. Bunların yanında bir diğer kısıtlama ise, filtrenin impuls yanıtının *sonlu* mu yoksa *sonsuz* süreli mi olacağıdır. Bu tezde, *sonlu süreli impuls yanıtı (FIR)* filtreler ile ilgilenilmektedir. Böyle yapmaktaki neden, FIR filtrelerin yapı itibarıyla *kararlı (stable)* olmasıdır. Çünkü bu filtrelerde giriş ile çıkış arasındaki bağlantı, sadece ve sadece ileri yönde, yani girişten çıkışa doğru olacak şekilde yapılmaktadır. Zaten bir filtreden bu şekilde işaret iletiliyor olması, o filtrenin impuls yanıtını sonlu süreli yapar. *Sonsuz süreli impuls yanıtı (IIR)* filtreler de ise, hem ileri yönde ve hem de geri yönde besleme vardır. Eğer uygun bir şekilde kontrol edilmezse, filtrede bulunan geri besleme, filtrenin *kararsız (unstable)* davranmasına ve

sonucunda *salınımına* (*oscillation*) yol açar. Böyle bir durum, kararlılığın çok önemli olduğu bir filtreleme işleminde kabul edilemez. IIR filtrelerin hem teorik hem de pratik açıdan, kendi kendilerine kararlılık probleminin üstesinden geldikleri kabul edilir; ama filtrenin *uyarlamalı* (*adaptive*) olması istendiği zaman, IIR filtrelerin yapısında bulunan geri besleme zorluklar yaratır. Bu nedenlerden dolayı, özellikle uyarlama işlemi gerektiren uygulamalarda ve diğer pek çok uygulamalarda, IIR filtrelerin yerine FIR filtrelerin kullanımı tercih edilmektedir. Buna göre, burada yapılan çalışma; tek girişi ve tek çıkışı olan, ayrık zamanda çalışan ve sonlu süreli impuls yanıtına sahip doğrusal filtreler ile sınırlıdır.

Bir filtrenin doğrusal (linear) olabilmesi için, filtre çıkışının filtre girişine uygulanan gözlem değerlerinin doğrusal bir fonksiyonu olması gerekir. Doğrusal filtreleme probleminin çözümü için yapılan istatistiksel yaklaşımda, faydalı işaret ve istenmeyen toplamsal gürültüye ait bazı istatistiksel parametrelerin (beklendidik değer ve ilişki fonksiyonları gibi) önceden bilindiği varsayılır. Bu bilgi tam olarak bilinmedikçe, doğrusal filtreyi tasarlamak veya tasarlanırsa bile optimum sonuç elde etmek imkansızdır; çünkü filtre ancak, girişine gelen işaretin istatistiği kendisinin tasarımında kullanılan ön bilgiye uyar ise optimum sonuç vermektedir. Bu durumda geriye kalan tek şey, girişindeki gürültülü işareti alıp bir takım istatistiksel kriterlere göre gürültünün etkisini minimum yaparak çıkışına ileten optimum doğrusal filtreyi tasarlamaktır. Bu filtre optimizasyonu sorununa faydalı bir yaklaşım, filtre çıkışı ile istenen işaret arasındaki fark olarak tanımlanan *hata işaretinin* (*error signal*) karesel beklendidik (karesel ortalama) değerinin minimum yapılmasıdır. Bu tür bir yaklaşımda durağan girişler için elde edilen sonuç, *Wiener filtre* olarak bilinir. Buna göre Wiener filtreler, karesel ortalama anlamda optimum olan doğrusal filtrelerdir. Öte yandan bu yaklaşımda, durağan olmayan girişler için Wiener filtre optimum değildir ve bu tür girişler için kullanılamaz, kullanılsa bile karesel ortalama hata için optimum sonuç vermez. Bu nedenle, durağan olmayan, yani zamanla değişen (time-varying) girişler için optimum filtrenin de zamanla değişen yapıda olması gerekmektedir. Bu zor soruna en iyi ve en yaygın çözüm ise, *Kalman filtre* ile bulunur.

3.1 Optimum Filtreleme Problemi

Doğrusal enine bir filtre, diğer adıyla bir FIR filtre ele alınsın (Şekil 3.1). Bu filtre; *depolama (storage)*, *çarpma (multiplication)* ve *toplama (addition)* olmak üzere üç temel işlem gerçekleştirir. Depolama işlemi, z^{-1} ile tanımlanmış tek bir örneklik gecikme sağlayan kaskad bağlı $(M-1)$ adet *birim gecikme (one-sample delay) elemanları* ile yapılır. Gecikme elemanlarının sayısı (buradaki filtrede $M-1$ adet), filtrenin *derecesini* belirler. $\{u(n)\}$ sürecine ait $u(n)$, $u(n-1)$, ..., $u(n-M+1)$ gözlem değerleri, *tap girişleri* olarak adlandırılır. $u(n)$, filtrenin *şu andaki* giriş değerini ve geriye kalan $(M-1)$ adet tap girişi ise, filtrenin *geçmiş* giriş değerlerini belirtir. $u(n)$, $u(n-1)$, ..., $u(n-M+1)$ tap girişlerinin sırasıyla $w_1, w_2, \dots, w_M$ *tap ağırlıkları (tap weights)* ile iç çarpımı, her bir iç çarpıma karşı düşen $w_1^*, w_2^*, \dots, w_M^*$ şeklinde blok gösterimli çarpıcılar tarafından gerçekleştirilir. Örneğin; $u(n)$ tap girişi ile w_1 tap ağırlığının iç çarpımı, w_1^* blok gösterimli çarpıcı tarafından gerçekleştirilir. Toplayıcıların görevi ise, filtrenin genel çıkışını oluşturmak üzere çarpıcı çıkışlarını toplamaktır. Böylece, enine bir filtrenin impuls yanıtının, $\{h_k\} = \{w_k^*\}$ $k = 1, 2, \dots, M$ şeklinde tap ağırlıkları ile ifade edildiği görülür.



Şekil 3.1 Doğrusal enine filtre blok diyagramı (Haykin, 1991)

$u(n)$, $u(n-1)$, ..., $u(n-M+1)$ tap girişleri ile temsil edilen ayrık zamanlı rasgele süreç $\{u(n)\}$, geniş anlamda durağan kabul edilecektir. Aynı zamanda, yapılan genel kabullerde hiçbir kayba neden olmayacağı için, rasgele sürecin ortalama

(beklendik) değeri de sıfır kabul edilmektedir. n anında filtre çıkışında üretilen işaret, $\hat{d}(n|U_n)$ ile gösterilir. Burada $(n|U_n)$ terimindeki n , bu çıkışın n anındaki çıkış olduğunu ve U_n ise, n anı da dahil olmak üzere n anına kadar olan $u(n)$, $u(n-1)$, ..., $u(n-M+1)$ tap girişlerinin oluşturduğu M boyutlu uzayı gösterir. (^) işareti, filtre çıkışı ile istenen yanıtı birbirinden ayırt etmek için kullanılmıştır. Çünkü, burada amaç: filtre çıkışını, istenen yanıtın *kestirimi* (*estimate*) olarak kullanmak, yani $\hat{d}(n|U_n)$ 'i $d(n)$ 'e mümkün olabildiğince benzetebilmektir. Filtre çıkışı, filtrenin tap girişleri ile impuls yanıtı arasında *doğrusal konvolüsyon toplamı* (*linear convolution sum*) kullanılarak

$$\hat{d}(n|U_n) = \sum_{k=1}^M w_k^* u(n-k+1) \quad (3.1)$$

bulunur. Şekil 3.1'de verilen filtre, istenen yanıt $d(n)$ ile kendi çıkışı arasındaki farkı belli bir istatistiksel kritere göre mümkün olduğunca azaltabilecek şekilde tasarlandığında, kestirim problemi çözülmüş olur. İstenen yanıtın n anındaki değeri ile n anındaki filtre çıkışı arasındaki fark,

$$e(n) = d(n) - \hat{d}(n|U_n) \quad (3.2)$$

kestirim hatası (*estimation error*) olarak adlandırılır. Wiener filter teorisinde optimum filtreyi tasarlamak için kullanılan istatistiksel kriter, *minimum karesel ortalama hata* (*minimum mean-square error*) olmaktadır. Bu nedenle $w_1, w_2, \dots, w_M$ tap ağırlıkları, *kestirim hatasının karesel ortalaması* veya kısaca *karesel ortalama hata* olarak adlandırılan $J(\mathbf{w})$ başarı göstergesini minimum yapacak şekilde seçilir.

$$J(\mathbf{w}) = E[e(n)e^*(n)] \quad (3.3)$$

Burada $\mathbf{w}$ vektör, $w_1, w_2, \dots, w_M$ tap ağırlıklarını içeren $M \times 1$ boyutlu tap ağırlık vektörüdür. Karesel ortalama hata, her zaman için gerçel ve pozitif bir değerdir. $J(\mathbf{w})$ başarı göstergesi minimum yapılarak, *minimum karesel ortalama anlamında* elde edilebilecek en iyi (*optimum*) filtreye ulaşılır.

3.2 Hata Başarım Yüzeyi

Wiener filtre teorisinin ayrık zamanlı versiyonunu geliştirmede, ifadelerin matris formunu kullanmak daha uygun olacaktır. Filtrenin $M \times 1$ boyutlu *tap ağırlık vektörü*

$$\mathbf{w}^T = [w_1, w_2, \dots, w_M] \quad (3.4)$$

şeklinde ve n anındaki $M \times 1$ boyutlu $\mathbf{u}(n)$ tap giriş vektörü ise

$$\mathbf{u}^T(n) = [u(n), u(n-1), \dots, u(n-M+1)] \quad (3.5)$$

şeklinde gösterilir. Bu durumda (3.1) eşitliği, bu iki vektörün iç çarpımı olarak da

$$\hat{d}(n | U_n) = \mathbf{w}^H \mathbf{u}(n) \quad (3.6)$$

yazılabilir. Her iki tarafın Hermitian transpozu alındığında,

$$\hat{d}^*(n | U_n) = \mathbf{u}^H(n) \mathbf{w} \quad (3.7)$$

elde edilir. Böylece, (3.2) eşitliğiyle verilen kestirim hatası ve kompleks eşleniği

$$e(n) = d(n) - \mathbf{w}^H \mathbf{u}(n) \quad (3.8)$$

$$e^*(n) = d^*(n) - \mathbf{u}^H(n) \mathbf{w} \quad (3.9)$$

vektörel formlarında yazılabilir. (3.8) ve (3.9) eşitlikleri (3.3) eşitliğinde yerine konulduğunda, karesel ortalama hata

$$J(\mathbf{w}) = E[(d(n) - \mathbf{w}^H \mathbf{u}(n)) (d^*(n) - \mathbf{u}^H(n) \mathbf{w})] \quad (3.10)$$

haline gelir. $\mathbf{w}$ tap ağırlık vektörü, zamanla değişmeyen sabit bir vektördür. Buna göre, (3.10) eşitliğinin sağ tarafının açılımı

$$\begin{aligned} J(\mathbf{w}) = & E[d(n) d^*(n)] - \mathbf{w}^H E[\mathbf{u}(n) d^*(n)] - E[d(n) \mathbf{u}^H(n)] \mathbf{w} \\ & + \mathbf{w}^H E[\mathbf{u}(n) \mathbf{u}^H(n)] \mathbf{w} \end{aligned} \quad (3.11)$$

şeklinde olur.

Tap giriş vektörü $\mathbf{u}(n)$ ile $d(n)$ istenen yanıtının *birleşik durağan* olduğu varsayılarak, (3.11) eşitliğinde bulunan dört adet beklendik değer şu şekilde yorumlanabilir:

1. İstenen yanıt $d(n)$ sıfır ortalamalı kabul edildiğinde, $E[d(n) d^*(n)]$ beklendik değeri istenen yanıtın değişintisine eşit olur.

$$\sigma_d^2 = E[d(n) d^*(n)] \quad (3.12)$$

2. $E[\mathbf{u}(n) d^*(n)]$ beklendik değeri, tap giriş vektörü $\mathbf{u}(n)$ ile $d(n)$ istenen yanıtı arasındaki $M \times 1$ boyutlu *çapraz ilişki (cross-correlation) vektörü*dür ve $\mathbf{p}$ ile gösterilir.

$$\mathbf{p} = E[\mathbf{u}(n) d^*(n)] \quad (3.13)$$

$$p(1-k) = E[u(n-k+1) d^*(n)] \quad , \quad k = 1, 2, \dots, M \quad (3.14)$$

$$\mathbf{p}^T = [p(0), p(-1), \dots, p(1-M)] \quad (3.15)$$

3. $E[d^*(n) \mathbf{u}^H(n)]$ beklendik değeri, $\mathbf{p}$ vektörünün Hermitian transpozudur.

$$\mathbf{p}^H = E[d(n) \mathbf{u}^H(n)] \quad (3.16)$$

4. $E[\mathbf{u}(n) \mathbf{u}^H(n)]$ beklendik değeri, Bölüm 2.2’de de açıklandığı gibi, $\mathbf{u}(n)$ tap giriş vektörünün $M \times M$ boyutlu ilişki matrisine eşittir.

$$\mathbf{R} = E[\mathbf{u}(n) \mathbf{u}^H(n)] = \begin{bmatrix} r(0) & r(1) & \dots & r(M-1) \\ r(-1) & r(0) & \dots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(-M+1) & r(-M+2) & \dots & r(0) \end{bmatrix} \quad (3.17)$$

Satır sayısı k ve sütun sayısı da i ile gösterildiğinde, $k, i = 1, 2, \dots, M$ olmak üzere $\mathbf{R}$ ilişki matrisinin (k, i) . elemanı

$$r(i-k) = E[u(n-k+1) u^*(n-i+1)] \quad (3.18)$$

ifadesi ile tanımlanır. Burada $\mathbf{R}$ matrisinin Hermitian özelliğinden dolayı,

$$r^*(i-k) = r(k-i) \quad (3.19)$$

olduğu unutulmamalıdır. Öyleyse, durağan bir sürecin $M \times M$ boyutlu $\mathbf{R}$ ilişki matrisi, girişin özilişki fonksiyonunun aldığı $r(0), r(1), \dots, r(M-1)$ değerleriyle tek olarak tanımlanabilir.

(3.12), (3.13), (3.16) ve (3.17) eşitlikleri (3.11) eşitliğinde yerlerine yazıldığında, karesel ortalama hata için daha kısa bir ifade

$$J(\mathbf{w}) = \sigma_d^2 - \mathbf{p}^H \mathbf{w} - \mathbf{w}^H \mathbf{p} + \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (3.20)$$

elde edilir. (3.20) eşitliğindeki $\mathbf{w}^H \mathbf{R} \mathbf{w}$ terimi, şu anlama gelir: tap giriş vektörü $\mathbf{u}(n)$ ile istenen yanıt $d(n)$ *birleşik durağan* olduğu zaman, karesel ortalama hata tap ağırlık vektörünün ikinci dereceden bir fonksiyonu olur. Bu nedenle, karesel ortalama hatanın tap ağırlık vektörünün elemanlarına olan bağımlılığı, tek bir minimum noktası olan $(M+1)$ boyutlu kase şekilli (bowl-shaped) bir yüzey olarak düşünülebilir. Bu yüzeye, enine filtrenin *hata başarımlı yüzeyi* (*error-performance surface*) adı verilir.

Bu durumda yapılması gereken tek şey, hata başarımlı yüzeyinin *minimum noktasında* (diğer bir deyişle, *dibinde*) çalışacak filtreyi tasarlamaktır. Hata başarımlı yüzeyinin minimum noktasında karesel ortalama hata $J_{\min}$ ile gösterilen *en küçük değerini* alırken, bu hata değerine karşı düşen tap ağırlık vektörü de $\mathbf{w}_o$ ile gösterilen *optimum* değerini alır. Bu şekilde elde edilen enine filtreye ise, *karesel ortalama hata anlamında optimum* enine filtre denir.

$\mathbf{w}_o$ optimum tap ağırlık vektörünü elde edebilmek için, karesel ortalama hatanın (3.20) tap ağırlık vektörüne göre türevi alınarak sıfıra eşitlenir. Elde edilen sonuç, optimum tap ağırlık vektörü $\mathbf{w}_o$ olur. Fakat bundan önce, skaler değerli bir fonksiyonun ($J(\mathbf{w})$) kompleks değerli bir vektöre ($\mathbf{w}$) göre türevinin nasıl alınacağını bilmesi gerekir. Burada bu konu üzerinde durulmayacağı için, ayrıntılı bilgi edinmek üzere Haykin (1991) kaynağına başvurulabilir.

3.3 Normal Denklemleri

σ_d^2 değışintisi sabit bir değer olduđu için, $\mathbf{w}$ vektörüne göre türevi de sıfır olur. (3.20) eşitliğindeki diğeri üç terimin $\mathbf{w}$ vektörüne göre türevleri ise,

$$\frac{d}{d\mathbf{w}}(\mathbf{p}^H \mathbf{w}) = 0 \quad , \quad \frac{d}{d\mathbf{w}}(\mathbf{w}^H \mathbf{p}) = 2 \mathbf{p} \quad , \quad \frac{d}{d\mathbf{w}}(\mathbf{w}^H \mathbf{R} \mathbf{w}) = 2 \mathbf{R} \mathbf{w}$$

şeklinde olmaktadır. Böylece, karesel ortalama hatanın tap ağırlık vektörüne göre türevi şeklinde tanımlanan ∇ *gradyen vektör (gradient vector)* için

$$\nabla = \frac{dJ(\mathbf{w})}{d\mathbf{w}} = -2 \mathbf{p} + 2 \mathbf{R} \mathbf{w} \quad (3.21)$$

ifadesine ulaşılır. $\mathbf{w}_o$, gradyen vektörü sıfır vektör yapan optimum tap ağırlık vektörünü gösterebilir. Bu durumda (3.21) eşitliğinden,

$$\mathbf{R} \mathbf{w}_o = \mathbf{p} \quad (3.22)$$

sonucu elde edilir. (3.22) eşitliği, Wiener-Hopf integral denklemlerinin ayrık zamandaki halidir ve *normal denklemleri* olarak adlandırılır. Wiener-Hopf integral denklemine neden böyle bir ikinci isim verildiği Bölüm 3.4'te açıklanacaktır. (3.22) eşitliğinin her iki yanının sol taraftan $\mathbf{R}^{-1}$ ters matrisi ile çarpımı, $\mathbf{w}_o$ optimum tap ağırlık vektörünü verir.

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p} \quad (3.23)$$

Görüldüğü gibi, $\mathbf{w}_o$ optimum tap ağırlık vektörünü bulabilmek için, iki istatistiksel parametrenin ($\mathbf{u}(n)$ tap giriş vektörüne ait $\mathbf{R}$ ilişki matrisi ve $\mathbf{u}(n)$ ile $d(n)$ istenen yanıtı arasındaki $\mathbf{p}$ çapraz ilişki vektörü) önceden biliniyor olması gerekmektedir. (3.4), (3.15) ve (3.17) eşitliklerini (3.23) eşitliğinde yerlerine yazarak, bu matrisel eşitliğin açık formu

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r(-1) & r(0) & \cdots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(-M+1) & r(-M+2) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} w_{o1} \\ w_{o2} \\ \vdots \\ w_{oM} \end{bmatrix} = \begin{bmatrix} p(0) \\ p(-1) \\ \vdots \\ p(1-M) \end{bmatrix} \quad (3.24)$$

şeklinde ve bu açık formun eşdeğeri de

$$\sum_{i=1}^M w_{oi} r(i-k) = p(1-k) \quad , \quad k = 1, 2, \dots, M \quad (3.25)$$

şeklinde yazılır.

3.4 Diklik Kuralı

(3.22) eşitliği, minimum karesel ortalama hata anlamında optimum olan bir enine (transversal) filtreye ait w_o tap ağırlık vektörünü tanımlar. Bu eşitlik, (3.13) ve (3.17) eşitlikleri kullanılarak yeniden

$$E[\mathbf{u}(n)\mathbf{u}^H(n)]\mathbf{w}_o = E[\mathbf{u}(n)d^*(n)] \quad (3.26)$$

şeklinde yazılabilir. w_o vektörü, zamana bağlı olmayan sabit değerler içeren bir vektör olduğundan, beklendik değer operatörünün içine alınabilir. Daha sonra, sağ taraftaki beklendik değer operatörü sol taraftaki beklendik değer operatörünün içine yazılarak,

$$E[\mathbf{u}(n)(d^*(n) - \mathbf{u}^H(n)\mathbf{w}_o)] = 0 \quad (3.27)$$

eşitliğine ulaşılır. (3.9) eşitliğinden yararlanılarak, optimum filtreye ait kestirim hatası $e_o(n)$

$$e_o^*(n) = d^*(n) - \mathbf{u}^H(n)\mathbf{w}_o \quad (3.28)$$

biçiminde yazılır. Bu durumda (3.27) eşitliği,

$$E[\mathbf{u}(n)e_o^*(n)] = 0 \quad (3.29)$$

olarak yazılabilir. Burada $0, M \times 1$ boyutlu sıfır vektördür.

(3.29) eşitliği gösterir ki: bir enine filtre optimum koşulda çalıştığı zaman, kestirim hatası $e_o(n)$ ile $\mathbf{u}(n)$ tap giriş vektörünün her bir elemanı birbirine dik (orthogonal) olmaktadır. Bu sonuç ise, *diklik kuralı (principle of orthogonality)* olarak bilinir. Bu da şunu ortaya koyar: tap girişleri ile kestirim hatası arasındaki diklik kriteri ve minimum karesel ortalama hata kriteri, birbirine özdeş iki kriterdir. Diğer bir ifadeyle, bu iki kriterden birinin gerçekleşmesiyle elde edilen sonuç, diğerinin gerçekleşmesiyle elde edilecek sonuca özdeş olmaktadır.

Optimum filtrenin bir başka yararlı özelliği, (3.29) eşitliğini optimum tap ağırlık vektörünün Hermitian transpozu ile sol taraftan çarparak elde edilir.

$$\mathbf{w}_o^H E[\mathbf{u}(n) e_o^*(n)] = 0 \quad (3.30)$$

Tap ağırlık vektörü yine, beklendik değer operatörünün içine alınır.

$$E[\mathbf{w}_o^H \mathbf{u}(n) e_o^*(n)] = 0 \quad (3.31)$$

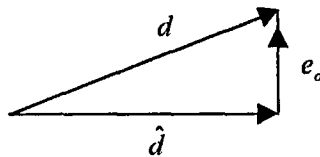
$\mathbf{u}(n)$ tap girişine karşı optimum filtrenin vereceği çıkışın

$$\hat{d}(n|U_n) = \mathbf{w}_o^H \mathbf{u}(n) \quad (3.32)$$

olduğu göz önünde bulundurularak, (3.29) eşitliği yeniden

$$E[\hat{d}(n|U_n) e_o^*(n)] = 0 \quad (3.33)$$

şeklinde yazılabilir. Bu eşitlik de gösterir ki: bir enine filtre optimum koşulda çalıştığı zaman, kestirim hatası $e_o(n)$ ile filtrenin çıkışındaki $\hat{d}(n|U_n)$ kestirimi de birbirine dik olmaktadır. (3.33) eşitliği, optimum filtrenin çıkışında varolan koşullara ait ilginç bir geometrik yorumun (Şekil 3.2) yapılmasını sağlar.



Şekil 3.2 İstenen yanıt, filtre çıkışındaki kestirim ve kestirim hatası arasındaki ilişki

Şekilde istenen yanıt d , filtre çıkışı $\hat{d}$ ve bu ikisine karşı gelen kestirim hatası ise e_o ile gösterilmektedir. Görüldüğü gibi, bir optimum filtre için, kestirim hatasını gösteren vektör filtre çıkışını gösteren vektöre *normal (dik)* olmaktadır. İşte bu nedenle de, optimum filtreyi tanımlayan (3.22) eşitliğine *normal denklemleri* denilmektedir.

3.5 Minimum Karesel Ortalama Hata

(3.20) eşitliği, herhangi bir w tap ağırlık vektörü için elde edilen karesel ortalama hatayı verir. Tap ağırlık vektörü optimum değerini w_o aldığı zaman, karesel ortalama hata da $J_{\min}$ ile gösterilen minimum değerini

$$J_{\min} = \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o - \mathbf{w}_o^H \mathbf{p} + \mathbf{w}_o^H \mathbf{R} \mathbf{w}_o \quad (3.34)$$

alır. (3.22) ile verilen $\mathbf{R} \mathbf{w}_o = \mathbf{p}$ normal denklemleri kullanılarak,

$$\begin{aligned} J_{\min} &= \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o \\ &= \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p} \end{aligned} \quad (3.35)$$

eşitliği elde edilir. Bu eşitliğin eşdeğeri ise,

$$J_{\min} = \sigma_d^2 - \sum_{k=1}^M w_{ok} p^* (1 - k) \quad (3.36)$$

olmaktadır. Minimum karesel ortalama hata $J_{\min}$ için bir başka yararlı ifade,

$$d(n) = \hat{d}(n|U_n) + e_o(n) \quad (3.37)$$

eşitliğinin her iki tarafının karesel beklendik değeri alınarak bulunur.

$$E[d(n)d^*(n)] = E[(\hat{d}(n|U_n) + e_o(n))(\hat{d}(n|U_n) + e_o(n))^*] \quad (3.38)$$

Optimum filtrelerde, filtre çıkışı $\hat{d}(n|U_n)$ ile kestirim hatası $e_o(n)$ birbirine dik (orthogonal) olduğundan dolayı, (3.38) eşitliğinde

$$E[\hat{d}^*(n | U_n) e_o(n)] = E[e_o^*(n) \hat{d}(n | U_n)] = 0 \quad (3.39)$$

değerleri yerlerine yazılarak,

$$E[|d(n)|^2] = E[|\hat{d}(n | U_n)|^2] + E[|e_o(n)|^2] \quad (3.40)$$

$$\sigma_d^2 = \sigma_{\hat{d}}^2 + J_{\min} \quad (3.41)$$

elde edilir. Burada, $d(n)$ istenen yanıtı ve $u(n)$ tap girişi sıfır ortalama kabul edilmiştir. (3.41) eşitliğinin minimum karesel ortalama hata için çözümü

$$J_{\min} = \sigma_d^2 - \sigma_{\hat{d}}^2 \quad (3.42)$$

şeklinde olacaktır. Bu eşitlikten, optimum filtre için minimum karesel ortalama hatanın, istenen yanıtın değışintisi ile filtre çıkışında üretilen kestirimin değışintisi arasındaki farka eşit olduğu görülür.

(3.42) eşitliğinin, karesel ortalama hatanın alacağı minimum değerin daima sıfır ile 1 arasında değışecek şekilde normalize edilmesi, bazı açıklamaların yapılmasında kolaylık sağlayacaktır. Bu nedenle, eşitliğin her iki tarafı istenen yanıtın değışintisine bölünerek

$$\frac{J_{\min}}{\sigma_d^2} = 1 - \frac{\sigma_{\hat{d}}^2}{\sigma_d^2} \quad (3.43)$$

bu normalizasyon gerçekleştirilir. Eşitliğin her iki tarafının istenen yanıtın değışintisine bölünmesinde herhangi bir sakınca yoktur. Çünkü σ_d^2 , bir tek özel durum dışında asla sıfır olmaz. Bu özel durum ise, $d(n)$ istenen yanıtının her n değeri için sıfır olması ile ortaya çıkar. (3.43) eşitliğinin sol tarafındaki bölüm, *normalize karesel ortalama hata (normalized mean-squared error)* olarak adlandırılır ve ε ile gösterilir. Öyleyse (3.43) eşitliği,

$$\varepsilon = 1 - \frac{\sigma_{\hat{d}}^2}{\sigma_d^2} \quad (3.44)$$

şeklinde yazılabilir. $\sigma_{\hat{d}}^2 / \sigma_d^2$ bölümü daima 1'den küçük ve pozitif değerkler aldığı için, normalize karesel ortalama hatanın alacağı değerkler, yukarıda da söylendiği gibi,

$$0 \leq \varepsilon \leq 1 \quad (3.45)$$

sınırları arasında olacaktır. Eğer ε sifira eşit ise; optimum filtre, $\hat{d}(n|U_n)$ kestirimi ile $d(n)$ istenen yanıtı arasında tam bir uyum sağlayabilme bakımından, mükemmel çalışıyor demektir. Diğer yandan, ε 1'e eşit ise; $\hat{d}(n|U_n)$ ile $d(n)$ arasında hiçbir uyum yok demektir ve bu durum, karşılaşılabilecek en kötü durumdur.

Filtredeki tap ağırlıklarının sayısı (Şekil 3.1'de M adet vardır) arttıkça, herhangi bir $d(n)$ istenen yanıtı için optimum enine filtrenin elde edeceği başarımlar da artmaktadır. Filtrenin başarımı arttıkça da ε değeri azalacaktır.

3.6 Hata Başarım Yüzeyinin Kanonik Formu

İlk olarak, (3.35) eşitliğinin ilk satırındaki ifade (3.20) eşitliğinden çıkarılır. Sonra, çıkarma işlemi sonucunda elde edilen ifadede bulunan $\mathbf{p}$ vektörlerini ortadan kaldırmak üzere (3.22) eşitliği kullanıldığında, karesel ortalama hata

$$J(\mathbf{w}) = J_{\min} + (\mathbf{w} - \mathbf{w}_o)^H \mathbf{R} (\mathbf{w} - \mathbf{w}_o) \quad (3.46)$$

şeklinde *kuadratik formda* yazılmış olur. Eşitlikten de görülebileceği gibi, karesel ortalama hatayı minimum yapabilen sadece tek bir tap ağırlık vektörü ($\mathbf{w}_o$) vardır. Bu ise, $\mathbf{w}_o$ vektöründen başka hiçbir optimal vektörün olmadığı anlamına gelir. Yeterince açıklayıcı olmasına karşın, hata başarım yüzeyinin gösterimini basitleştirebilmek amacıyla, (3.46) eşitliğinin tabanında bir dönüşüm yapılacaktır. Bu dönüşüm için

$$\mathbf{R} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H \quad (3.47)$$

eşitliğinden yararlanılacaktır. Buna göre (3.46) eşitliği,

$$J(\mathbf{w}) = J_{\min} + (\mathbf{w} - \mathbf{w}_o)^H \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H (\mathbf{w} - \mathbf{w}_o) \quad (3.48)$$

haline gelir. Tap ağırlık vektörü $\mathbf{w}$ ile optimum çözüm olan $\mathbf{w}_o$ vektörü arasındaki fark,

$$\mathbf{v} = \mathbf{Q}^H (\mathbf{w} - \mathbf{w}_o) \quad (3.49)$$

şeklinde bir $\mathbf{v}$ vektörüne dönüştürülsün. Bu durumda (3.48) eşitliğindeki kuadratik form,

$$J(\mathbf{v}) = J_{\min} + \mathbf{v}^H \mathbf{\Lambda} \mathbf{v} \quad (3.50)$$

şeklinde tanımlanan *kanonik forma* dönüşür. $\mathbf{v}$ vektörünün k . elemanı v_k ile gösterilmek üzere, (3.50) eşitliğinin açık formu

$$\begin{aligned} J(\mathbf{v}) &= J_{\min} + \sum_{k=1}^M \lambda_k v_k v_k^* \\ &= J_{\min} + \sum_{k=1}^M \lambda_k |v_k|^2 \end{aligned} \quad (3.51)$$

olmaktadır. Buradan da görülmektedir ki: $\mathbf{w}$ vektöründen $\mathbf{v}$ vektörüne yapılan taban dönüşümü ile elde edilen kanonik formdaki karesel ortalama hata formülü, kuadratik formdaki karesel ortalama hata formülüne göre daha basittir. Hata başarımlı yüzeyinin gösterilmesinde kanonik formu daha uygun kılan özellik, $\mathbf{v}$ dönüşüm vektörünün elemanlarının hata başarımlı yüzeyinin *ana eksenlerini (principal axes)* oluşturuyor olmasıdır.

4. DOĞRUSAL ÖNGÖRÜ

Durağan ayrık zamanlı bir rasgele sürecin elde bulunan geçmiş örnek değerlerine bakılarak gelecekteki bir örnek değerinin *öngörülmesi* (*predicting*), zaman serilerinin analizinde karşılaşılan en meşhur problemlerden biridir. n anı da dahil olmak üzere, n anına kadar olan $(M+1)$ adet örnek değerini gösteren $u(n), u(n-1), \dots, u(n-M)$ zaman serisi ele alınsın. Öngörme işlemi; $u(n)$ örneğinin kestirimini elde etmek için, zaman serisinin bu örnekten önceki $u(n-1), u(n-2), \dots, u(n-M)$ geçmiş örneklerinin kullanılması olarak tanımlanabilir. $u(n-1), u(n-2), \dots, u(n-M)$ örnekleri tarafından oluşturulan M boyutlu uzay, U_{n-1} ile ve $u(n)$ örneğinin bu örnekler kullanılarak elde edilen *öngörülen* (*predicted*) değeri de, $\hat{u}(n|U_{n-1})$ ile gösterilsin. Genelde öngörülen değer, öngöründe kullanılan örneklerin $u(n-1), u(n-2), \dots, u(n-M)$ oluşturduğu bir θ fonksiyonu ile

$$\hat{u}(n|U_{n-1}) = \theta(u(n-1), \dots, u(n-M)) \quad (4.1)$$

ifade edilir. Yapılan öngörünün doğrusal olup olmadığı ise, bu fonksiyonun kendisini oluşturan örneklerin doğrusal bir toplamından oluşup oluşmadığına göre belirlenir. Eğer θ fonksiyonu $u(n-1), u(n-2), \dots, u(n-M)$ örneklerinin aşağıdaki gibi

$$\hat{u}(n|U_{n-1}) = \sum_{k=1}^M w_{ok}^* u(n-k) \quad (4.2)$$

doğrusal toplamından oluşuyor ise, yapılan öngörünün *doğrusal öngörü* (*linear prediction*) olduğu söylenir. $w_{o1}, w_{o2}, \dots, w_{oM}$ sabit katsayılardır. Bölüm başlığından da anlaşılacağı gibi, bu ve bundan sonraki tüm bölümlerde sadece doğrusal öngörü üzerinde durulacaktır.

$u(n-1), u(n-2), \dots, u(n-M)$ geçmiş örnekleri kullanılarak $u(n)$ örneğinin öngörülen değerinin elde edilmesine, *ileri yönde tek adımlık öngörü* (*one step prediction the forward direction*) veya kısaca *ileri yönde öngörü* (*forward prediction*) adı verilir. Bir başka öngörü biçimi de, $u(n-M)$ geçmiş örneğine ait öngörülen değerin bu örnekten sonra gelen $u(n), u(n-1), \dots, u(n-M+1)$ örneklerinin kullanılarak elde edilmesidir. Bu öngörü biçimi ise, *geri yönde öngörü* (*backward prediction*) olarak adlandırılır.

Bu bölümde, ileri yönde doğrusal öngörü ve geri yönde doğrusal öngörü konuları ele alınacaktır. Wiener filtre teorisi kullanılarak, karesel ortalama anlamda optimum olan ileri ve geri yönde doğrusal öngörücüler tasarlanacaktır. Wiener filtre teorisini uygulayabilmek için de, ayrık zamanlı rasgele sürecin geniş anlamda durağan olduğu varsayılacaktır. Ayrıca, hesaplamalarda kolaylık sağlaması açısından, sürecin sıfır ortalamalı olduğu da kabul edilecektir. Bölüm 2’de de anlatıldığı gibi, bu özelliklere sahip bir rasgele sürecin ilişki matrisi Toeplitz yapıdadır. Toeplitz yapı ise, hesaplama yönünden daha verimli algoritmaların elde edilmesine imkan sağlar.

4.1 İleri Yönde Doğrusal Öngörü

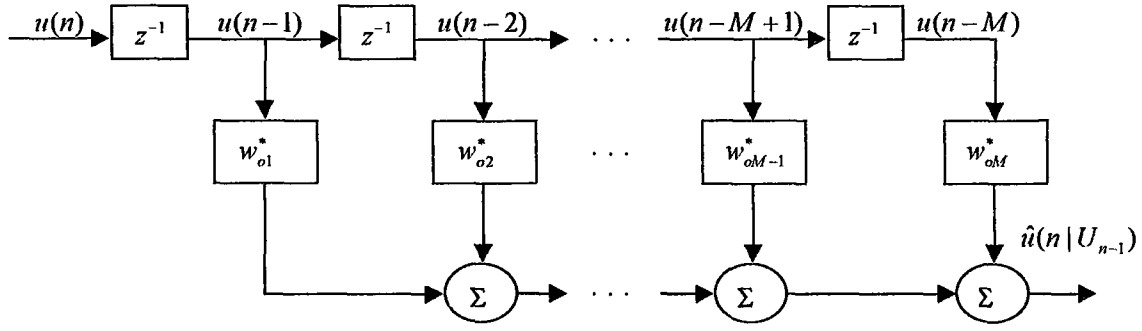
$u(n-1), u(n-2), \dots, u(n-M)$ tap girişlerine ve bu girişlere karşı düşen $w_{o1}, w_{o2}, \dots, w_{oM}$ tap ağırlıklarına sahip doğrusal bir enine filtre ile oluşturulan *ileri yönde öngörücü (forward predictor)*, Şekil 4.1(a)’da gösterilmiştir. Tap girişlerinin, sıfır ortalamalı geniş anlamda durağan bir rasgele süreçten elde edildiği varsayılacak ve tap ağırlıklarının ise, Wiener filtre teorisine uyacak şekilde karesel ortalama anlamda optimum oldukları kabul edilecektir. Öngörülen değer, (4.2) eşitliğinde verilen $\hat{u}(n|U_{n-1})$ ifadesine ve istenen yanıt $d(n)$ ise, giriş sürecinin n anındaki örnek değeri $u(n)$ ’e eşit olmaktadır.

$d(n) = u(n)$ olduğu için, n anındaki istenen yanıt $d(n)$ ile öngörülen değeri $\hat{u}(n|U_{n-1})$ arasındaki fark olarak tanımlanan *ileri yönde öngörü hatası*

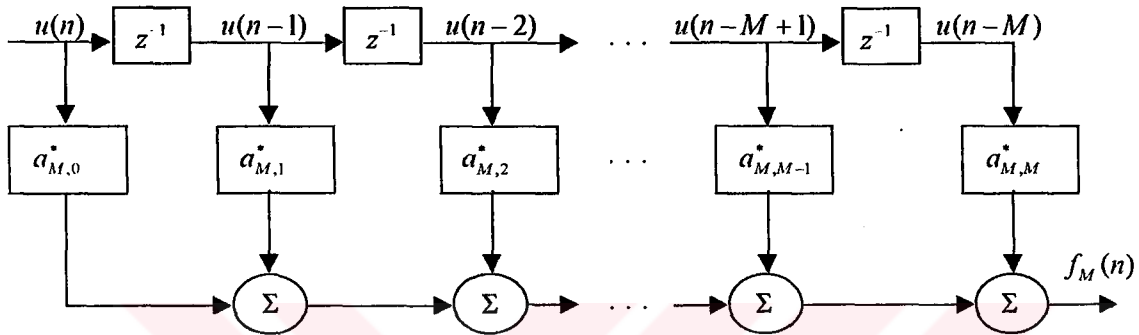
$$f_M(n) = u(n) - \hat{u}(n|U_{n-1}) \quad (4.3)$$

olarak yazılır. Alt indis M , birim gecikme elemanlarının sayısını, yani *öngörücünün derecesini* belirtir. P_M , *minimum karesel ortalama öngörü hatasını (minimum mean-squared prediction error)* belirtsin.

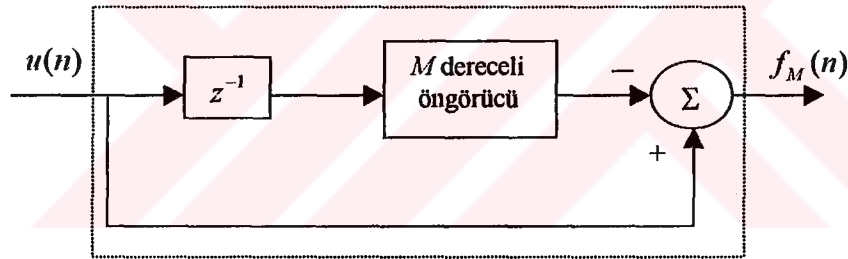
$$P_M = E[|f_M(n)|^2] \quad (4.4)$$



(a)



(b)



Öngörü hata filtresi

(c)

Şekil 4.1 (a) Tek adımlı ileri yönde öngörücü; (b) öngörücü hata filtresi; (c) öngörücü ile öngörü hata filtresi arasındaki ilişki (Haykin,1991)

Tap girişlerinin sıfır ortalamalı kabul edilmesi nedeniyle, (4.3) eşitliğine göre, ileri yönde öngörü hatası da sıfır ortalamalı olacaktır. Bu durumda P_M 'in alacağı değer, ileri yönde öngörü hatasının değışintisine eşit olur. Öyleyse, bu koşullar altında P_M 'e, “minimum karesel ortalama öngörü hatası” dışında, *ensemble ortalamalı ileri yönde öngörü hata gücü* (*ensemble-averaged forward prediction error power*) ismi de verilebilir. Bu nedenle, P_M bundan sonra, ileri yönde öngörü hata gücü olarak adlandırılacaktır.

$$\mathbf{w}_o^T = [w_{o1}, w_{o2}, \dots, w_{oM}] \quad (4.5)$$

Şekil 4.1(a)'da verilen ileri yönde öngörücüye ait $M \times 1$ boyutlu $\mathbf{w}_o$ optimum tap ağırlık vektörünü elde edebilmek için, (3.22) eşitliğiyle verilen normal denklemlerini çözmek gerekir. Bunun için de; ilki, $u(n-1)$, $u(n-2)$, ..., $u(n-M)$ tap girişlerine ait $M \times M$ boyutlu ilişki matrisi ve ikincisi ise, bu tap girişleri ile istenen yanıt $u(n)$ arasındaki $M \times 1$ boyutlu çapraz ilişki vektörü olmak üzere, iki değere gerek duyulur. P_M 'i hesaplayabilmek içinse, üçüncü bir değer olarak $u(n)$ 'in değişintisine ihtiyaç vardır. Aşağıda, bu üç değer tek tek ele alınmaktadır:

1. $u(n-1)$, $u(n-2)$, ..., $u(n-M)$ tap girişlerine ait $M \times 1$ boyutlu tap giriş vektörü

$$\mathbf{u}^T(n-1) = [u(n-1), u(n-2), \dots, u(n-M)] \quad (4.6)$$

ile gösterildiği için, bu tap girişlerine ait ilişki matrisi önceki ilişki matrislerinden farklı olarak

$$\mathbf{R} = E[\mathbf{u}(n-1)\mathbf{u}^H(n-1)] \quad (4.7)$$

eşitliği ile tanımlanır; ama bu matrisin açılımı da, (2.12) ve (3.17) eşitlikleriyle verilen açılımların aynısıdır.

2. $u(n-1)$, $u(n-2)$, ..., $u(n-M)$ tap girişleri ile istenen yanıt $u(n)$ arasındaki çapraz ilişki vektörü

$$E[\mathbf{u}(n-1)u^*(n)] = \begin{bmatrix} E[u(n-1)u^*(n)] \\ E[u(n-2)u^*(n)] \\ \vdots \\ E[u(n-M)u^*(n)] \end{bmatrix} = \begin{bmatrix} r(-1) \\ r(-2) \\ \vdots \\ r(-M) \end{bmatrix} = \mathbf{r} \quad (4.8)$$

ile gösterilir.

3. $u(n)$ sıfır ortalamalı olduğu için, değişintisi $r(0)$ değerine eşit olacaktır.

Wiener filtreye (Şekil 3.1) ait değerler ve bu değerlere karşı düşen ileri yönde öngörücü (Şekil 4.1(a)) değerleri, Çizelge 4.1'de gösterilmiştir. Çizelgenin son sütunu, daha sonra ele alınacak geri yönde öngörücüye aittir.

Çizelge 4.1 Wiener filtre değişkenlerinin özeti

Değişkenler	Wiener filtre (Şekil 3.1)	İleri yönde öngörücü (Şekil 4.1(a))	Geri yönde öngörücü (Şekil 4.2(a))
Tap giriş vektörü	$\mathbf{u}(n)$	$\mathbf{u}(n-1)$	$\mathbf{u}(n)$
İstenen yanıt	$d(n)$	$u(n)$	$u(n-M)$
Tap ağırlık vektörü	$\mathbf{w}_o$	$\mathbf{w}_o$	$\mathbf{g}$
Kestirim hatası	$e(n)$	$f_M(n)$	$b_M(n)$
Tap girişlerinin ilişki matrisi	$\mathbf{R}$	$\mathbf{R}$	$\mathbf{R}$
Tap girişleri ile istenen yanıt arasındaki çapraz ilişki vektörü	$\mathbf{p}$	$\mathbf{r}$	$\mathbf{r}^{B*}$
Minimum karesel ortalama hata	$J_{\min}$	P_M	P_M

Böylece, çizelgede birbirine karşı düşen değerler kullanılarak, durağan girişler için (3.22) eşitliğiyle verilen normal denklemlerini buradaki ileri yönde öngörü problemine

$$\mathbf{R} \mathbf{w}_o = \mathbf{r} \quad (4.9)$$

şeklinde kolayca uyarlayabiliriz. Benzer biçimde, (3.35) eşitliğinin ilk satırı kullanılarak ileri yönde öngörü hata gücü için

$$P_M = r(0) - \mathbf{r}^H \mathbf{w}_o \quad (4.10)$$

eşitliği yazılabilir. (4.9) ve (4.10) eşitliklerinden görüldüğü gibi, giriş sürecine ait $r(k)$ özilişki fonksiyonunun $k = 0, 1, 2, \dots, M$ zaman farkları için aldığı $(M+1)$ adet değer, $\mathbf{w}_o$ tap ağırlık vektörünü ve P_M ileri yönde öngörü hata gücünü belirleyebilmek için tek başına yeterli olmaktadır.

4.1.1 İleri yönde öngörü hata filtresi

(4.2) eşitliği (4.3) eşitliğinde yerine yazıldığında, ileri yönde öngörü hatası

$$f_M(n) = u(n) - \sum_{k=1}^M w_{ok}^* u(n-k) \quad (4.11)$$

olarak elde edilir. $k = 0, 1, 2, \dots, M$ olmak üzere $a_{M,k}$, Şekil 4.1(a)'da verilen ileri yönde öngörücünün tap ağırlıkları ile

$$a_{M,k} = \begin{cases} 1 & , k = 0 \\ -w_{ok}^* & , k = 1, 2, \dots, M \end{cases} \quad (4.12)$$

şeklinde bağlantılı yeni bir enine filtreye ait tap ağırlıkları olsun. Bu yeni tap ağırlıkları kullanılarak, (4.11) eşitliği

$$f_M(n) = \sum_{k=0}^M a_{M,k}^* u(n-k) \quad (4.13)$$

şeklinde tek bir toplam içerisinde yazılabilir. $u(n)$, $u(n-1)$, ..., $u(n-M)$ örneklerini giriş olarak kullanıp çıkışında (4.13) eşitliğindeki ileri yönde öngörü hatasını üreten enine filtreye, *M. dereceden ileri yönde öngörü hata filtresi (forward prediction error filter of order M)* denir (Şekil 4.1(b)).

İleri yönde öngörü hata filtresi ile ileri yönde öngörücü arasında Şekil 4.1(c)'de verilen blok diyagramdaki gibi bir ilişki vardır. İleri yönde öngörü hata filtresinin uzunluğu (tap ağırlık sayısı) öngörü filtresinden bir fazla olmasına karşın, her iki filtrenin derecesi birbirine (M dereceli) eşittir. Çünkü, her iki filtrenin birim gecikme elemanlarının sayısı aynıdır.

4.1.2 İleri yönde öngörü için genişletilmiş normal denklemleri

İleri yönde öngörücünün tap ağırlık vektörünü tanımlayan normal denklemleri (4.9) ile ileri yönde öngörü hata gücünü (4.10) birleştirip (4.14) eşitliğindeki gibi tek bir matris formunda yazmak mümkündür.

$$\begin{bmatrix} r(0) & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R} \end{bmatrix} \begin{bmatrix} 1 \\ -\mathbf{w}_o \end{bmatrix} = \begin{bmatrix} P_M \\ \mathbf{0} \end{bmatrix} \quad (4.14)$$

Eşitliğin solunda yer alan $(M+1) \times (M+1)$ boyutlu matris, daha önce Bölüm 2.2’de (2.20) ve (2.24) eşitlikleriyle de ele alınmış olup, $u(n)$, $u(n-1)$, ..., $u(n-M)$ tap girişlerine ait ilişki matrisidir. $\mathbf{R}$ matris ise (4.7)’de verilmiştir. Sol tarafın ikinci terimi olan $(M+1) \times 1$ boyutlu vektörün açılımı (4.12) eşitliğinden görülebilir. Sağ taraftaki vektörün içinde yazılı $\mathbf{0}$ vektör ise, $M \times 1$ boyutlu bir sıfır vektördür. Öyleyse, (4.14) eşitliğinin matris formunda açılımı

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M) \\ r(-1) & r(0) & \cdots & r(M-1) \\ \vdots & \vdots & \ddots & \vdots \\ r(-M) & r(-M+1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_{M,0} \\ a_{M,1} \\ \vdots \\ a_{M,M} \end{bmatrix} = \begin{bmatrix} P_M \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.15)$$

olacaktır. (4.14) eşitliğinin sistem denklemi cinsinden açılımı

$$\sum_{l=0}^M a_{M,l} r(l-i) = \begin{cases} P_M, & i = 0 \\ 0, & i = 1, 2, \dots, M \end{cases} \quad (4.16)$$

şeklinde $(M+1)$ adet denklemden oluşur. Gösterimde farklı ama içerikte birbirinin aynısı olan (4.15) ve (4.16) eşitliklerinin her ikisi de, M . dereceden ileri yönde öngörü hata filtresinin *genişletilmiş normal denklemleri* olarak adlandırılır.

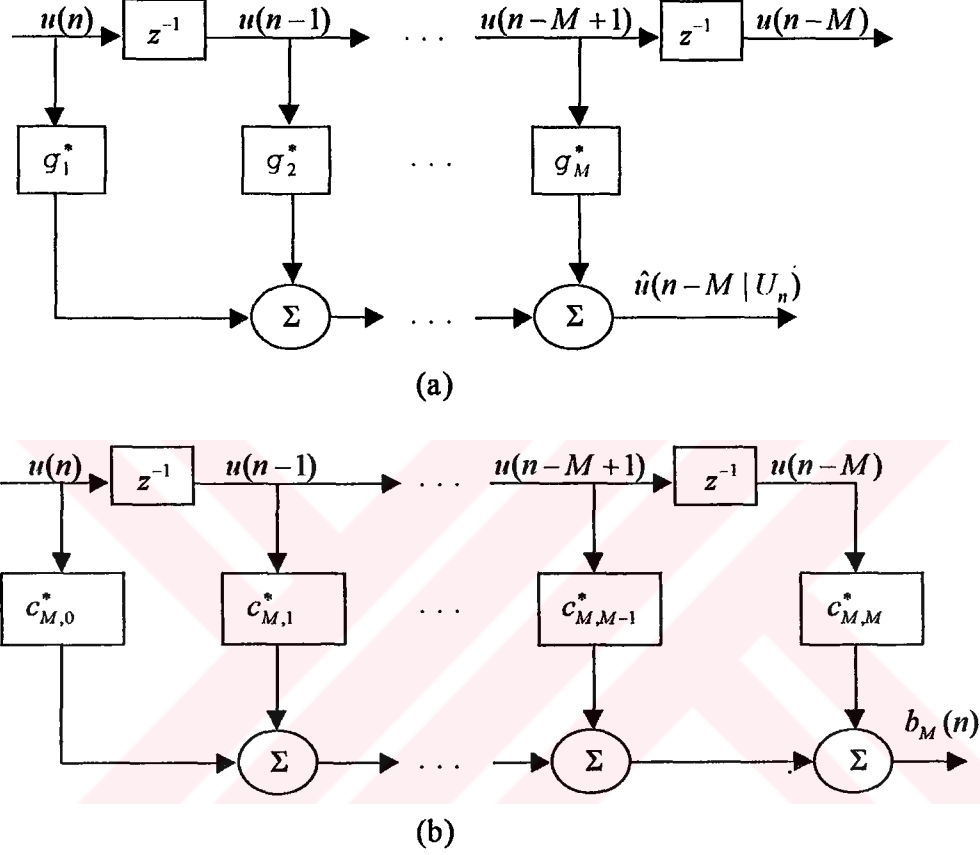
4.2 Geri Yönde Doğrusal Öngörü

Başlangıçta da belirtildiği gibi, geri yönde tek adımlık öngörü veya kısaca geri yönde öngörü işlemi; bize verilmekte olan bir $u(n)$, $u(n-1)$, ..., $u(n-M)$ zaman serisi içinden $u(n)$, $u(n-1)$, ..., $u(n-M+1)$ örneklerini kullanarak, $u(n-M)$ geçmiş örneğine ait öngörülen değerin elde edilmesidir.

$u(n)$, $u(n-1)$, ..., $u(n-M+1)$ örnekleri tarafından oluşturulan M boyutlu uzay, U_n ile ve $u(n-M)$ örneğinin bu örnekler kullanılarak öngörülen değeri de, $\hat{u}(n-M|U_n)$ ile gösterilsin. Geri yönde doğrusal öngörü işlemi ile elde edilen $\hat{u}(n-M|U_n)$ değeri

$$\hat{u}(n-M | U_n) = \sum_{k=1}^M g_k^* u(n-k+1) \quad (4.17)$$

eşitliği ile gösterilir. $g_1, g_2, \dots, g_M$ tap ağırlıklarıdır. (4.17) eşitliği ile tanımlanan geri yönde öngörücü Şekil 4.2(a)'da verilmiştir.



Şekil 4.2 (a) Tek adımlı geri yönde öngörücü; (b) geri yönde öngörü hata filtresi (Haykin,1991)

Geri yönde öngöründe istenen yanıt $d(n) = u(n-M)$ olduğuna göre, istenen yanıt ile öngörülen değeri $\hat{u}(n-M | U_n)$ arasındaki fark

$$b_M(n) = u(n-M) - \hat{u}(n-M | U_n) \quad (4.18)$$

geri yönde öngörü hatasını (*backward prediction error*) verir. İleri yönde öngöründe olduğu gibi burada da alt indis M , birim gecikme elemanlarının sayısını, yani geri yönde öngörücünün derecesini belirtmektedir.

Geri yönde öngörücüye ait *minimum karesel ortalama öngörü hatası*

$$P_M = E[|b_M(n)|^2] \quad (4.19)$$

olmaktadır. P_M aynı zamanda, *ensemble ortalamalı geri yönde öngörü hata gücü* olarak da adlandırılabilir. Çünkü, tap girişleri burada da sıfır ortalamalı kabul edilmektedir.

$$\mathbf{g}^T = [g_1, g_2, \dots, g_M] \quad (4.20)$$

Şekil 4.2(a)'da verilen geri yönde öngörücüye ait $M \times 1$ boyutlu $\mathbf{g}$ optimum tap ağırlık vektörünü elde edebilmek için, (3.22) eşitliğiyle verilen normal denklemlerini çözmek gerekir. Bunun için de; ilki, $u(n), u(n-1), \dots, u(n-M+1)$ tap girişlerine ait $M \times M$ boyutlu ilişki matrisi ve ikincisi ise, bu tap girişleri ile istenen yanıt $u(n-M)$ arasındaki $M \times 1$ boyutlu çapraz ilişki vektörü olmak üzere, iki değere gerek duyulur. P_M 'i hesaplamak içinse, üçüncü bir değer olarak $u(n-M)$ 'in değişintisine ihtiyaç vardır. Aşağıda, bu üç değer tek tek ele alınmaktadır:

1. $u(n), u(n-1), \dots, u(n-M+1)$ tap girişlerine ait $M \times 1$ boyutlu tap giriş vektörü

$$\mathbf{u}^T(n) = [u(n), u(n-1), \dots, u(n-M+1)] \quad (4.21)$$

ile gösterildiği için, bu tap girişlerine ait ilişki matrisi (2.11) ve (2.12) eşitlikleriyle verilen ilişki matrisinin aynısıdır.

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)] \quad (4.22)$$

2. $u(n), u(n-1), \dots, u(n-M+1)$ tap girişleri ile istenen yanıt $u(n-M)$ arasındaki çapraz ilişki vektörü

$$E[\mathbf{u}(n)\mathbf{u}^*(n-M)] = \begin{bmatrix} E[u(n)\mathbf{u}^*(n-M)] \\ E[u(n-1)\mathbf{u}^*(n-M)] \\ \vdots \\ E[u(n-M+1)\mathbf{u}^*(n-M)] \end{bmatrix} = \begin{bmatrix} r(M) \\ r(M-1) \\ \vdots \\ r(1) \end{bmatrix} = \mathbf{r}^{B*} \quad (4.23)$$

ile gösterilir. Burada, özilişki fonksiyonunun $r(-k) = r^*(k)$ şeklinde tanımlanan Hermitian özelliğinden yararlanılmıştır.

3. $u(n-M)$ sıfır ortalamalı olduğu için, değışintisi $r(0)$ değerine eşit olacaktır.

Çizelge 4.1'in son sütununda özetlenmiş geri yönde öngörücüye ait çeşitli değerler ile bu değerlere karşı düşen diğer değerler kullanılarak, durağan girişler için (3.22) eşitliğiyle verilen normal denklemlerini buradaki geri yönde öngörü problemine

$$\mathbf{R} \mathbf{g} = \mathbf{r}^{B*} \quad (4.24)$$

şeklinde kolayca uyarlayabiliriz. Benzer biçimde, (3.35) eşitliğinin ilk satırı kullanılarak geri yönde öngörü hata gücü için

$$P_M = r(0) - \mathbf{r}^{BT} \mathbf{g} \quad (4.25)$$

eşitliği yazılabilir. (4.24) ve (4.25) eşitliklerinden görüldüğü gibi, giriş sürecine ait $r(k)$ özilişki fonksiyonunun $k = 0, 1, 2, \dots, M$ zaman farkları için aldığı $(M+1)$ adet değer, $\mathbf{g}$ tap ağırlık vektörünü ve P_M geri yönde öngörü hata gücünü belirleyebilmek için tek başına yeterli olmaktadır.

4.2.1 Geri ve ileri yönde öngörücüler arasındaki ilişki

İleri yönde öngörücüye ait (4.9) ve geri yönde öngörücüye ait (4.24) eşitlikleri birbiriyle karşılaştırıldığında, (4.24) eşitliğinin sağ tarafındaki vektörün (4.9) eşitliğinin sağ tarafındaki vektörden iki şekilde farklı olduğu görülür:

1. Elemanları geri yönde düzenlenmiştir.
2. Elemanlarının kompleks eşleniği kullanılmıştır.

Bu iki farkı ortadan kaldırmak için, (4.24) eşitliğinin sağındaki vektörün elemanları önce geri yönde düzenlenecek ve sonra da elde edilen eşitliğin kompleks eşleniği alınacaktır.

Geri yönde düzenleme işleminin (4.24) eşitliğinin solunda kalan $\mathbf{R}$ ilişki matrisi üzerindeki etkisi, bu matrisin transpozuna $\mathbf{R}^B = \mathbf{R}^T$ eşit olmaktadır. Aslında, bir matrisin geri yönde düzenlenmesi demek, o matrisin önce sütunlarının sonra da

satırlarının (veya önce satırlarının sonra da sütunlarının) geri yönde düzenlenmesi demektir. Ancak, $\mathbf{R}$ ilişki matrisi Toeplitz özellikli olduğu için, geri yönde düzenleme işlemi ile elde edilen $\mathbf{R}^B$ matrisi onun transpozuna $\mathbf{R}^T$ eşit olmaktadır. Öyleyse, (4.24) eşitliğinin geri düzenlenmiş versiyonu

$$\mathbf{R}^T \mathbf{g}^B = \mathbf{r}^* \quad (4.26)$$

olacaktır. İkinci farkı ortadan kaldırmak için, geri yönde düzenlenmiş bu eşitliğin her iki tarafının kompleks eşleniği alınır ve

$$\mathbf{R}^H \mathbf{g}^{B*} = \mathbf{r} \quad (4.27)$$

elde edilir. $\mathbf{R}$ matris Hermitian matris ($\mathbf{R} = \mathbf{R}^H$) olduğuna göre, bu son eşitlik

$$\mathbf{R} \mathbf{g}^{B*} = \mathbf{r} \quad (4.28)$$

haline gelir. Görüldüğü gibi, geri yönde öngörü için (4.24) eşitliğiyle verilen normal denklemleri, sağ taraftaki farkların ortadan kaldırılması sonucu (4.28) ile verilen son halini almıştır. (4.9) ile (4.28) karşılaştırıldığında, geri yönde öngöründe kullanılan tap ağırlıkları ile ileri yönde öngöründe kullanılan tap ağırlıkları arasında

$$\mathbf{g}^{B*} = \mathbf{w}_o \quad (4.29)$$

şeklinde bir ilişkinin olduğu ortaya çıkar. (4.29) eşitliği şunu ortaya koyar: bir geri yönde öngörücü, tap ağırlıkları geri yönde düzenlenip kompleks eşlenikleri alınmak suretiyle, ileri yönde öngörücüye dönüştürülebilir.

Şimdi ise, geri yönde öngörü hata gücü ile ileri yönde öngörü hata gücünün birbirine eşit olduğu gösterilmeye çalışılacaktır. Bunun için önce, $(\mathbf{r} \mathbf{g})^B = \mathbf{r}^B \mathbf{g}^B$ olduğunun anlaşılması gerekir. Buna göre, geri yönde düzenleme işlemi transpoz işleminden farklıdır ve bir vektör çarpımının geri yönde düzenlenmesi demek, çarpımdaki her bir vektörün tek tek geri yönde düzenlenmesi demektir. Öyleyse, $\mathbf{r}^{BT} \mathbf{g} = \mathbf{r}^T \mathbf{g}^B$ eşitliği göz önünde bulundurularak geri yönde öngörü hata gücü için verilen (4.25) eşitliği,

$$P_M = r(0) - \mathbf{r}^T \mathbf{g}^B \quad (4.30)$$

şeklinde yazılabilir. P_M ve $r(0)$, her zaman için gerçel değerlidir. Bu nedenle, her iki tarafın kompleks eşleniği alındığında,

$$P_M = r(0) - \mathbf{r}^H \mathbf{g}^{B*} \quad (4.31)$$

eşitliği elde edilir. (4.29) eşitliği (4.31) eşitliğinde yerine yazılacak olursa, (4.10) ile verilen ileri yönde öngörü hata gücüne ulaşılır. Buradan da görülür ki: ileri ve geri yönde öngörü hata güçleri birbirine eşittir. Dolayısıyla, her iki gücün de P_M ile gösterilmesinde hiçbir sakınca yoktur.

4.2.2 Geri yönde öngörü hata filtresi

Geri yönde doğrusal öngörü tanımlayan (4.17) eşitliği (4.18) eşitliğinde yerine yazılarak, geri yönde öngörü hatası

$$b_M(n) = u(n-M) - \sum_{k=1}^M g_k^* u(n-k+1) \quad (4.32)$$

biçiminde de ifade edilebilir. Giriş-çıkış ilişkili bir eşitlik elde edebilmek için, (4.32) eşitliğinin tek bir toplam içinde yazılması gerekir. Bu amaçla, geri yönde öngörü filtresine ait g_k tap ağırlıkları ile $u(n-M)$ istenen yanıtına ait 1 katsayısını

$$c_{M,k} = \begin{cases} -g_{k+1}^*, & k = 0, 1, \dots, M-1 \\ 1, & k = M \end{cases} \quad (4.33)$$

şeklinde birlikte içeren yeni tap ağırlıkları tanımlansın. Bu durumda (4.32) eşitliği,

$$b_M(n) = \sum_{k=0}^M c_{M,k}^* u(n-k) \quad (4.34)$$

halini alır. Giriş ile çıkış arasındaki ilişkiyi belirten türden olan bu eşitliğe göre geri yönde öngörü hatası $b_M(n)$, girişleri $u(n)$, $u(n-1)$, ..., $u(n-M)$ ve tap ağırlıkları ise $c_{M,0}, c_{M,1}, \dots, c_{M,M}$ olan bir enine filtrenin çıkışı olarak gösterilebilir. Böyle bir

enine filtre Şekil 4.2(b)'de gösterilmiş olup, *M. dereceden geri yönde öngörü hata filtresi* olarak adlandırılır.

Geri yönde öngörücünün tap ağırlıklarını ileri yönde öngörücünün tap ağırlıkları cinsinden tanımlayan (4.29) eşitliğinin açık formu ve bu açık formun eşdeğeri

$$\begin{aligned} g_{M-k+1}^* &= w_{ok} \quad , \quad k = 1, 2, \dots, M \\ g_k &= w_{o,M-k+1}^* \quad , \quad k = 1, 2, \dots, M \end{aligned} \quad (4.35)$$

şeklinde olmaktadır. (4.35) eşitliği (4.33) eşitliğinde kullanıldığında,

$$c_{M,k} = \begin{cases} -w_{o,M-k}^* & , \quad k = 0, 1, \dots, M-1 \\ 1 & , \quad k = M \end{cases} \quad (4.36)$$

geri yönde öngörü hata filtresinin tap ağırlıkları ileri yönde öngörü filtresinin tap ağırlıkları cinsinden yazılmış olur. Benzer şekilde, (4.12) eşitliği de (4.36) eşitliğinde kullanılarak,

$$c_{M,k} = a_{M,M-k}^* \quad , \quad k = 0, 1, \dots, M \quad (4.37)$$

ileri yönde öngörü hata filtresinin tap ağırlıkları ile geri yönde öngörü hata filtresinin tap ağırlıkları arasındaki ilişki elde edilir. Bu son eşitlik (4.34) eşitliğinde yerine yazılırsa,

$$b_M(n) = \sum_{k=0}^M a_{M,M-k} u(n-k) \quad (4.38)$$

olduğu görülür. (4.37) ve (4.38) eşitliklerinden şu sonuç çıkarılır: durağan girişler için ileri ve geri yönde öngörü hata filtreleri birbiriyle tam bir ilişkili içindedir ve bu ilişki sayesinde (tap ağırlıkları geri yönde düzenlenip kompleks eşlenikleri alınarak), ileri yönde öngörü hata filtresi kendisine karşı düşen geri yönde öngörü hata filtresine dönüştürülebilir.

4.2.3 Geri yönde öngörü için genişletilmiş normal denklemleri

Geri yönde öngörücünün tap ağırlık vektörünü tanımlayan normal denklemleri (4.24) ile geri yönde öngörü hata gücünü (4.25) birleştirip (4.39) eşitliğindeki gibi tek bir matris formunda yazmak mümkündür.

$$\begin{bmatrix} \mathbf{R} & \mathbf{r}^{B*} \\ \mathbf{r}^{BT} & r(0) \end{bmatrix} \begin{bmatrix} -\mathbf{g} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ P_M \end{bmatrix} \quad (4.39)$$

Eşitliğin solunda yer alan $(M+1) \times (M+1)$ boyutlu matris, daha önce Bölüm 2.2’de (2.21) ve (2.26) eşitlikleriyle de ele alınmış olup, $u(n)$, $u(n-1)$, ..., $u(n-M)$ tap girişlerine ait ilişki matrisidir. $\mathbf{R}$ matris ise (4.22)’de verilmiştir. Sol tarafın ikinci terimi olan $(M+1) \times 1$ boyutlu vektörün açılımı (4.33) eşitliğinden görülebilir. Öyleyse, (4.39) eşitliğinin matris formunda açılımı

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M) \\ r(-1) & r(0) & \cdots & r(M-1) \\ \vdots & \vdots & \ddots & \vdots \\ r(-M) & r(-M+1) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} a_{M,M}^* \\ \vdots \\ a_{M,1}^* \\ a_{M,0}^* \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ P_M \end{bmatrix} \quad (4.40)$$

olacaktır. (4.39) eşitliğinin sistem denklemleri cinsinden açılımı

$$\sum_{l=0}^M a_{M,M-l}^* r(l-i) = \begin{cases} 0 & , \quad i = 0, 1, \dots, M-1 \\ P_M & , \quad i = M \end{cases} \quad (4.41)$$

şeklinde $(M+1)$ adet denklemden oluşur. Gösterimde farklı ama içerikte birbirinin aynısı olan (4.40) ve (4.41) eşitliklerinin her ikisi de, M . dereceden geri yönde öngörü hata filtresinin *genişletilmiş normal denklemleri* olarak adlandırılır.

(4.40) eşitliğindeki ilişki matrisi ile (4.15) eşitliğindeki ilişki matrisinin tıpa tıpa aynı olması, ileri ve geri yönde öngörü hata filtrelerinin aynı tap girişlerine $u(n)$, $u(n-1)$, ..., $u(n-M)$ sahip olmasından dolayıdır. Ayrıca (4.40) eşitliği, (4.15) eşitliğinin geri yönde düzenlenip kompleks eşleniğinin alınmış halidir.

4.3 Levinson–Durbin Yinelemesi

Burada, öngörü hata filtresinin tap ağırlıklarını ve öngörü hata gücünü hesaplamak için genişletilmiş normal denklemlerini çözen bir metot tanıtılacaktır. Bu metot, yinelemeli (recursive) bir yapıya sahiptir ve filtrenin tap girişlerine ait ilişki matrisinin Toeplitz özelliğinden faydalanmaktadır. *Levinson–Durbin yinelemesi* (*Levinson–Durbin recursion*) olarak bilinen bu metot, ilk defa Levinson (1947) tarafından ortaya konulmuş ve daha sonra da Durbin (1960) tarafından bağımsız olarak yeniden formüle edilmiştir. Bu metot m . dereceden öngörü hata filtresine ait genişletilmiş normal denklemlerin çözümünü bulmak için, $(m-1)$. dereceden öngörü hata filtresine ait genişletilmiş normal denklemlerin çözümünden yararlanır. Filtrenin son derecesi M ile gösterildiğinde, $m = 1, 2, \dots, M$ olmaktadır. Bu metodun önemli özelliği, hesaplama verimliliğidir; çünkü, yapılan işlemlerin (çarpma veya bölme) sayısında ve kullanılan bellek miktarında diğer metotlara göre büyük bir tasarruf sağlamaktadır. Bu metodu elde etmek için, ileri ve geri yönde öngörücülerin matris formundaki gösterimleri kullanılacaktır.

m . dereceden ileri yönde öngörü hata filtresine ait $(m+1) \times 1$ boyutlu tap ağırlık vektörü $\mathbf{a}_m$ ile gösterilsin. Bu durumda bu ileri yönde öngörü hata filtresine karşı düşen geri yönde öngörü hata filtresine ait $(m+1) \times 1$ boyutlu tap ağırlık vektörü, $\mathbf{a}_m$ vektörünün elemanlarının geri yönde düzenlenip kompleks eşleniklerinin alınması ile elde edilen $\mathbf{a}_m^{B*}$ vektörü ile gösterilir. Öyleyse, $(m-1)$. dereceden ileri yönde öngörü hata filtresine ve bu filtreye karşı düşen geri yönde öngörü hata filtresine ait $m \times 1$ boyutlu tap ağırlık vektörleri de, $\mathbf{a}_{m-1}$ ve $\mathbf{a}_{m-1}^{B*}$ vektörleri ile gösterilir. Levinson–Durbin yinelemesi, aşağıda verilen iki eşdeğer gösterimden birisi kullanılarak ifade edilir:

1. İleri yönde öngörü hata filtresinin tap ağırlık vektörü

$$\mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + \Gamma_m \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} \quad (4.42)$$

şeklinde derece bakımından güncelleştirilir (order-updated). Γ_m , sabit bir sayıdır.

Bu derece güncelleme eşitliğinin skaler formda yazılımı

$$a_{m,k} = a_{m-1,k} + \Gamma_m a_{m-1,m-k}^*, \quad k = 0, 1, \dots, m \quad (4.43)$$

şeklindedir. $a_{m,k}$, m . dereceden ileri yönde öngörü hata filtresinin k . tap ağırlığıdır. Benzer şekilde $a_{m-1,k}$, $(m-1)$. dereceden ileri yönde öngörü hata filtresinin k . tap ağırlığı ve $a_{m-1,m-k}^*$ ise, buna karşı düşen $(m-1)$. dereceden geri yönde öngörü hata filtresinin k . tap ağırlığıdır. Ayrıca (4.43) eşitliğinde, (4.12) eşitliğinden dolayı $a_{m-1,0} = 1$ ve (4.42) eşitliğinden dolayı da $a_{m-1,m} = 0$ olmaktadır.

2. Geri yönde öngörü hata filtresinin tap ağırlık vektörü

$$\mathbf{a}_m^{H*} = \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} + \Gamma_m^* \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} \quad (4.44)$$

şeklinde derece bakımından güncelleştirilir. Bu derece güncellemenin skaler formda yazılımı

$$a_{m,m-k}^* = a_{m-1,m-k}^* + \Gamma_m^* a_{m-1,k}, \quad k = 0, 1, \dots, m \quad (4.45)$$

şeklindedir. $a_{m,m-k}^*$, m . dereceden geri yönde öngörü hata filtresinin k . tap ağırlığıdır. Diğer elemanlar ise, (4.43) eşitliğindeki gibi tanımlanır.

Γ_m sabitinin, Levinson–Durbin yinelemesinin geçerliliğini doğrulamak için sağlamak zorunda olduğu koşulu belirlemek üzere şu dört aşama takip edilir:

1. (4.42) eşitliğinin her iki tarafı m . dereceden ileri yönde öngörü hata filtresinin $u(n)$, $u(n-1)$, ..., $u(n-m)$ tap girişlerine ait $(m+1) \times (m+1)$ boyutlu $\mathbf{R}_{m+1}$ ilişki matrisi ile soldan çarpıldığında, (4.15) eşitliğine benzeyen bir eşitlik

$$\mathbf{R}_{m+1} \mathbf{a}_m = \begin{bmatrix} P_m \\ \mathbf{0}_m \end{bmatrix} \quad (4.46)$$

elde edilir. P_m , ileri yönde öngörü hata gücü ve $\mathbf{0}_m$ ise, $m \times 1$ boyutlu sıfır vektördür. $\mathbf{R}_{m+1}$ matrisi ile $\mathbf{0}_m$ vektörüne ait $(m+1)$ ve m alt indisleri bunların boyutlarını belirtirken, P_m skaleri ile $\mathbf{a}_m$ vektörüne ait m alt indisi yapılan öngörünün derecesini belirtir.

2. (4.42) eşitliğinin sağ tarafındaki ilk terim için $\mathbf{R}_{m+1}$ matrisinin (2.21) eşitliğine benzeyen parçalanmış hali

$$\mathbf{R}_{m+1} = \begin{bmatrix} \mathbf{R}_m & \mathbf{r}_m^{B*} \\ \mathbf{r}_m^{BT} & r(0) \end{bmatrix} \quad (4.47)$$

kullanılacak olursa,

$$\mathbf{R}_{m+1} \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_m & \mathbf{r}_m^{B*} \\ \mathbf{r}_m^{BT} & r(0) \end{bmatrix} \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_m \mathbf{a}_{m-1} \\ \mathbf{r}_m^{BT} \mathbf{a}_{m-1} \end{bmatrix} \quad (4.48)$$

elde edilir. $\mathbf{R}_m$, $u(n)$, $u(n-1)$, ..., $u(n-m+1)$ tap girişlerine ait $m \times m$ boyutlu ilişki matrisi ve $\mathbf{r}_m^{B*}$ ise, bu tap girişleri ile $u(n-m)$ istenen yanıtı arasındaki $m \times 1$ boyutlu çapraz ilişki vektörüdür.

$(m-1)$. dereceden ileri yönde öngörü hata filtresi için genişletilmiş normal denklemleri, (4.46) eşitliğinde de olduğu gibi, (4.15) eşitliğine benzer şekilde

$$\mathbf{R}_m \mathbf{a}_{m-1} = \begin{bmatrix} P_{m-1} \\ \mathbf{0}_{m-1} \end{bmatrix} \quad (4.49)$$

yazılabilir. Ayrıca,

$$\begin{aligned} \Delta_{m-1} &= \mathbf{r}_m^{BT} \mathbf{a}_{m-1} = [r(-m), \quad r(1-m), \quad \dots, \quad r(-1)] \begin{bmatrix} a_{m-1,0} \\ a_{m-1,1} \\ \vdots \\ a_{m-1,m-1} \end{bmatrix} \\ &= \sum_{k=0}^{m-1} a_{m-1,k} r(k-m) \end{aligned} \quad (4.50)$$

şeklinde bir Δ_{m-1} skaleri tanımlansın. Bu durumda, (4.49) ve (4.50) eşitlikleri (4.48) eşitliğinde yerlerine yazılarak,

$$\mathbf{R}_{m+1} \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} = \begin{bmatrix} P_{m-1} \\ \mathbf{0}_{m-1} \\ \Delta_{m-1} \end{bmatrix} \quad (4.51)$$

eşitliğine ulaşılır. Böylece, (4.42) eşitliğinin sağ tarafındaki ilk terimin $\mathbf{R}_{m+1}$ matrisi ile soldan çarpılması sonucunda (4.51) eşitliği elde edilmiş olur.

3. (4.42) eşitliğinin sağ tarafındaki ikinci terim için $\mathbf{R}_{m+1}$ matrisinin bu defa (2.20) eşitliğine benzeyen bir diğer parçalanmış hali

$$\mathbf{R}_{m+1} = \begin{bmatrix} r(0) & \mathbf{r}_m^H \\ \mathbf{r}_m & \mathbf{R}_m \end{bmatrix} \quad (4.52)$$

kullanılacak olursa,

$$\mathbf{R}_{m+1} \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} = \begin{bmatrix} r(0) & \mathbf{r}_m^H \\ \mathbf{r}_m & \mathbf{R}_m \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_m^H \mathbf{a}_{m-1}^{B*} \\ \mathbf{R}_m \mathbf{a}_{m-1}^{B*} \end{bmatrix} \quad (4.53)$$

elde edilir. $\mathbf{R}_m$, $u(n-1)$, $u(n-2)$, ..., $u(n-m)$ tap girişlerine ait $m \times m$ boyutlu ilişki matrisi ve $\mathbf{r}_m$ ise, bu tap girişleri ile $u(n)$ istenen yanıtı arasındaki $m \times 1$ boyutlu çapraz ilişki vektörüdür. $\mathbf{r}_m^H \mathbf{a}_{m-1}^{B*}$ skaler değeri ise,

$$\begin{aligned} \mathbf{r}_m^H \mathbf{a}_{m-1}^{B*} &= [r(1), r(2), \dots, r(m)] \begin{bmatrix} a_{m-1,m-1}^* \\ a_{m-1,m-2}^* \\ \vdots \\ a_{m-1,0}^* \end{bmatrix} \\ &= \sum_{l=1}^m a_{m-1,m-l}^* r(l) \\ &= \Delta_{m-1}^* \end{aligned} \quad (4.54)$$

olmaktadır. $(m-1)$. dereceden geri yönde öngörü hata filtresi için genişletilmiş normal denklemleri, (4.40) eşitliğine benzer şekilde

$$\mathbf{R}_m \mathbf{a}_{m-1}^{B*} = \begin{bmatrix} \mathbf{0}_{m-1} \\ P_{m-1} \end{bmatrix} \quad (4.55)$$

yazılabilir. (4.54) ve (4.55) eşitlikleri (4.53) eşitliğinde yerlerine yazılarak,

$$\mathbf{R}_{m+1} \begin{bmatrix} 0 \\ \beta^* \\ \mathbf{a}_{m-1} \end{bmatrix} = \begin{bmatrix} \Delta_{m-1}^* \\ \mathbf{0}_{m-1} \\ P_{m-1} \end{bmatrix} \quad (4.56)$$

eşitliğine ulaşılır. Böylece, (4.42) eşitliğinin sağ tarafındaki ikinci terimin $\mathbf{R}_{m+1}$ matrisi ile soldan çarpılması sonucunda (4.56) eşitliği elde edilmiş olur.

4. (4.42) eşitliğinin her iki tarafının $\mathbf{R}_{m+1}$ matrisi ile soldan çarpımında 1., 2. ve 3. aşamalarda elde edilen (4.46), (4.51) ve (4.56) sonuçları kullanıldığında,

$$\begin{bmatrix} P_m \\ \mathbf{0}_m \end{bmatrix} = \begin{bmatrix} P_{m-1} \\ \mathbf{0}_{m-1} \\ \Delta_{m-1} \end{bmatrix} + \Gamma_m \begin{bmatrix} \Delta_{m-1}^* \\ \mathbf{0}_{m-1} \\ P_{m-1} \end{bmatrix} \quad (4.57)$$

eşitliği elde edilir.

(4.57) eşitliğinin sol ve sağ tarafındaki vektörlerin ilk elemanlarıyla,

$$P_m = P_{m-1} + \Gamma_m \Delta_{m-1}^* \quad (4.58)$$

eşitliği ve son elemanlarıyla da,

$$0 = \Delta_{m-1} + \Gamma_m P_{m-1} \quad (4.59)$$

eşitliği yazılabilir. (4.59) eşitliğinden, Γ_m sabiti için

$$\Gamma_m = -\frac{\Delta_{m-1}}{P_{m-1}} \quad (4.60)$$

yazılır. (4.60) eşitliği kullanılarak (4.58) eşitliğinde Δ_{m-1} yok edilirse,

$$P_m = P_{m-1} (1 - |\Gamma_m|^2) \quad (4.61)$$

şeklinde öngörü hata gücüne ait derece güncelleme eşitliği elde edilir.

Öngörü hata filtresinin derecesi (m) arttırılırsa, öngörü hata gücünün bu dereceye karşı gelen değeri (P_m) azalır veya aynı kalır; ama, asla artmaz. (4.4) ve (4.19) eşitliklerinden görülebileceği gibi, P_m asla negatif değer de alamaz. Öyleyse, P_m öngörü hata gücünün alacağı değerler her zaman için

$$0 \leq P_m \leq P_{m-1} \quad , \quad m \geq 1 \quad (4.62)$$

koşulunu sağlamak zorundadır. Yine (4.4) ve (4.19) eşitlikleri yardımıyla, başlangıç durumu olarak kabul edilen $m = 0$ için

$$P_0 = r(0) \quad (4.63)$$

yazılabileceği görülür. Çünkü bu durumda, $f_0(n) = b_0(n) = u(n)$ olmaktadır. (4.61) eşitliğine $m=1$ ile başlayıp her defasında 1 arttırılarak $m=M$ değerine gelindiğinde, M . dereceden öngörü hata filtresine ait öngörü hata gücü için

$$P_M = P_0 \prod_{m=1}^M (1 - |\Gamma_m|^2) \quad (4.64)$$

eşitliği elde edilir.

4.3.1 Δ_{m-1} ve Γ_m parametrelerinin anlamı

Levinson–Durbin yinelemesinin M . dereceden olan bir öngörü hata filtresine uygulanması sonucunda ortaya çıkan Γ_m parametreleri ($1 \leq m \leq M$), *yansımaya katsayıları* (*reflection coefficients*) olarak adlandırılır. (4.61) ve (4.62) eşitlikleri göz önüne alındığında, yansımaya katsayılarının her m değeri için

$$|\Gamma_m| \leq 1 \quad (4.65)$$

koşulunu sağlaması gerekir. Ayrıca (4.43) eşitliğinden, m . dereceden öngörü hata filtresi için Γ_m yansımaya katsayısının filtrenin son tap ağırlığına $a_{m,m}$ eşit olduğu

$$\Gamma_m = a_{m,m} \quad (4.66)$$

görülmür.

Δ_{m-1} parametresi ise, ileri yönde öngörü hatası $f_{m-1}(n)$ ile geciktirilmiş geri yönde öngörü hatası $b_{m-1}(n-1)$ arasındaki çapraz ilişki

$$\Delta_{m-1} = E[b_{m-1}(n-1) f_{m-1}^*(n)] \quad (4.67)$$

olarak açıklanabilir. $f_{m-1}(n)$, $(m-1)$. dereceden ileri yönde öngörü hata filtresinin $u(n)$, $u(n-1)$, ..., $u(n-m+1)$ tap girişlerine karşılık ürettiği çıkış ve $b_{m-1}(n-1)$ ise, $(m-1)$. dereceden geri yönde öngörü hata filtresinin $u(n-1)$, $u(n-2)$, ..., $u(n-m)$ tap girişlerine karşılık ürettiği çıkıştır.

$$f_0(n) = b_0(n) = u(n) \quad (4.68)$$

olduğundan dolayı, (4.67) eşitliğinde $m = 1$ için

$$\Delta_0 = E[b_0(n-1) f_0^*(n)] = E[u(n-1) u^*(n)] = r(-1) \quad (4.69)$$

yazılabilir.

P_{m-1} değeri, $f_{m-1}(n)$ ileri yönde öngörü hatasının karesel ortalama değeri olarak ele alınır ve (4.67) eşitliği (4.60) eşitliğinde yerine yazılırsa, Γ_m için ikinci bir eşitlik

$$\Gamma_m = - \frac{E[b_{m-1}(n-1) f_{m-1}^*(n)]}{E[|f_{m-1}(n)|^2]} \quad (4.70)$$

elde edilir. Bu eşitliğin sağ tarafı, $(-)$ işareti haricinde, *kısmi ilişki katsayısı* (*partial correlation coefficient*, PARCOR) olarak adlandırılır. Bu terminoloji, istatistiksel literatürde yaygın bir şekilde kullanılmaktadır (Box ve Jenkins, 1976). Öyleyse, yansıma katsayısı PARCOR katsayısının negatifi olarak tanımlanabilir.

Yansıma katsayılarının bir başka özelliği ise; aldıkları değerlere bakılarak, kendileri ile ilişkili ileri yönde öngörü hata filtresinin *minimum fazlı* ve geri yönde öngörü hata filtresinin ise *maksimum fazlı* olup olmadığının belirlenebilmesidir. Burada ispatı ile ilgilenilmediğinden, yansıma katsayılarının bu özelliği sadece teorisi verilerek aşağıda iki madde halinde kısaca anlatılmıştır.

1. m . dereceden bir ileri yönde öngörü hata filtresine ait $H_{f,m}(z)$ transfer fonksiyonunun z -düzleminde bulunan birim dairenin üzerinde veya dışında hiçbir sıfırının olmaması, yani tüm sıfırlarının birim daire içerisinde kalması için, bu ileri yönde öngörü hata filtresine ilişkin yansıma katsayılarının $|\Gamma_m| < 1$ koşulunu sağlaması gerekir. Bu koşulu sağlayan filtrelere ise *minimum fazlı* denir. Çünkü bu özellikteki filtreler, belli bir genlik yanıtına karşılık birim dairenin tam üzerinde seçilecek her z değeri için *minimum fazlı yanıt*a sahip olabilmektedirler (Oppenheim ve Schafer, 1975).

2. Yukarıda belirttiğimiz m . dereceden ileri yönde öngörü hata filtresine karşı düşen m . dereceden geri yönde öngörü hata filtresine ait transfer fonksiyonu

$$H_{b,m}(z) = z^{-m} H_{f,m}^*(1/z^*)$$

şeklinde ve $|\Gamma_m| < 1$ koşulunun sağlanması durumunda tüm sıfırları, z -düzleminde bulunan birim dairenin dışında yer alır. Bu tür filtrelere ise *maksimum fazlı* denir (Oppenheim ve Schafer, 1975).

4.4 Kafes Öngörücüler

İleri ve geri yönde öngörü hata filtreleri, Şekil 4.1 ve 4.2’de olduğu gibi, enine filtre yapısı kullanılarak gerçekleştirilebilir. Bunun dışında, her iki filtreyi tek bir yapıda birleştiren ve *kafes öngörücü (lattice predictor)* olarak adlandırılan bir başka filtre yapısı daha vardır. Bir kafes öngörücü, her biri kafes görünümüne sahip birden fazla modülün birbiri ardına kaskad bağlanmasından oluşur. Kafes öngörücüde bulunan modül sayısı, öngörü hata filtresinin derecesine eşittir. Bu nedenle, m . dereceden bir öngörü hata filtresinin kafes öngörücü ile yapılan gerçekleştirilmesinde m adet modül bulunur.

4.4.1 Öngörü hatalarının derece güncelleme yinelemeleri

Kafes öngörücüye ait giriş-çıkış bağıntıları çeşitli yollardan elde edilebilir. Burada bu bağıntıları elde etmek için, Levinson–Durbin yinelemesinden yararlanılacaktır. Öncelikle, tap girişleri $u(n)$, $u(n-1)$, ..., $u(n-m)$ olan m . dereceden ileri yönde

öngörü hata filtresin ait $(m+1) \times 1$ boyutlu $\mathbf{u}_{m+1}(n)$ vektörünün, (2.25) ve (2.27) eşitliklerinde gösterildiği gibi,

$$\mathbf{u}_{m+1}(n) = \begin{bmatrix} \mathbf{u}_m(n) \\ \text{-----} \\ u(n-m) \end{bmatrix} \quad (4.71)$$

şeklinde veya eşdeğeri olan

$$\mathbf{u}_{m+1}(n) = \begin{bmatrix} u(n) \\ \text{-----} \\ \mathbf{u}_m(n-1) \end{bmatrix} \quad (4.72)$$

şeklinde yazılabileceği hatırlanmalıdır. $\mathbf{u}_{m+1}(n)$ vektörünün (4.42) eşitliğinde verilen m . dereceden ileri yönde öngörü hata filtresine ait $(m+1) \times 1$ boyutlu tap ağırlık vektörü $\mathbf{a}_m$ ile iç çarpımı sonucunda ortaya çıkan her bir terim, soldan sağa doğru sırasıyla aşağıda ele alınmıştır:

1. Eşitliğin sol tarafı

$$f_m(n) = \mathbf{a}_m^H \mathbf{u}_{m+1}(n) \quad (4.73)$$

2. Eşitliğin sağ tarafındaki ilk terim için $\mathbf{u}_{m+1}(n)$ vektörünün (4.71) eşitliğiyle verilen açılımı kullanılarak,

$$\begin{aligned} \begin{bmatrix} \mathbf{a}_{m-1}^H & \vdots & 0 \end{bmatrix} \mathbf{u}_{m+1}(n) &= \begin{bmatrix} \mathbf{a}_{m-1}^H & \vdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_m(n) \\ \text{-----} \\ u(n-m) \end{bmatrix} \\ &= \mathbf{a}_{m-1}^H \mathbf{u}_m(n) \\ &= f_{m-1}(n) \end{aligned} \quad (4.74)$$

yazılabilir.

3. Eşitliğin sağ tarafındaki ikinci terim için $\mathbf{u}_{m+1}(n)$ vektörünün (4.72) eşitliğiyle verilen açılımı kullanılarak,

$$\begin{aligned}
\begin{bmatrix} 0 & \vdots & \mathbf{a}_{m-1}^{BT} \end{bmatrix} \mathbf{u}_{m+1}(n) &= \begin{bmatrix} 0 & \vdots & \mathbf{a}_{m-1}^{BT} \end{bmatrix} \begin{bmatrix} u(n) \\ \text{-----} \\ \mathbf{u}_m(n-1) \end{bmatrix} \\
&= \mathbf{a}_{m-1}^{BT} \mathbf{u}_m(n-1) \\
&= b_{m-1}(n-1)
\end{aligned} \tag{4.75}$$

yazılabilir. $b_{m-1}(n-1)$, $(m-1)$. dereceden geri yönde öngörü hata filtresinin çıkışında üretilen $b_{m-1}(n)$ hatasının geciktirilmişidir. (4.73), (4.74) ve (4.75) eşitlikleriyle verilen sonuçlar kullanıldığında, $\mathbf{a}_m^H \mathbf{u}_{m+1}(n)$ iç çarpımı için

$$f_m(n) = f_{m-1}(n) + \Gamma_m^* b_{m-1}(n-1) \tag{4.76}$$

elde edilir.

m . dereceden geri yönde öngörü hata filtresinin de aynı tap girişlerine $u(n)$, $u(n-1)$, ..., $u(n-m)$ sahip olması nedeniyle, $\mathbf{u}_{m+1}(n)$ vektörünün (4.71) ve (4.72)'deki parçalanmış halleri geri yönde öngörü için de aynen geçerlidir. $\mathbf{u}_{m+1}(n)$ vektörünün (4.44) eşitliğinde verilen m . dereceden geri yönde öngörü hata filtresine ait $(m+1) \times 1$ boyutlu tap ağırlık vektörü $\mathbf{a}_m^{B*}$ ile iç çarpımı sonucunda ortaya çıkan her bir terim, soldan sağa doğru sırasıyla aşağıda ele alınmıştır:

1. Eşitliğin sol tarafı

$$b_m(n) = \mathbf{a}_m^{BT} \mathbf{u}_{m+1}(n) \tag{4.77}$$

2. Eşitliğin sağ tarafındaki ilk terim için $\mathbf{u}_{m+1}(n)$ vektörünün (4.72) eşitliğiyle verilen açılımı kullanılarak,

$$\begin{aligned}
\begin{bmatrix} 0 & \vdots & \mathbf{a}_{m-1}^{BT} \end{bmatrix} \mathbf{u}_{m+1}(n) &= \begin{bmatrix} 0 & \vdots & \mathbf{a}_{m-1}^{BT} \end{bmatrix} \begin{bmatrix} u(n) \\ \text{-----} \\ \mathbf{u}_m(n-1) \end{bmatrix} \\
&= \mathbf{a}_{m-1}^{BT} \mathbf{u}_m(n-1) \\
&= b_{m-1}(n-1)
\end{aligned} \tag{4.78}$$

yazılabilir.

3. Eşitliğin sağ tarafındaki ikinci terim için $\mathbf{u}_{m+1}(n)$ vektörünün (4.71) eşitliğiyle verilen açılımı kullanılarak,

$$\begin{aligned} \begin{bmatrix} \mathbf{a}_{m-1}^H & \vdots & 0 \end{bmatrix} \mathbf{u}_{m+1}(n) &= \begin{bmatrix} \mathbf{a}_{m-1}^H & \vdots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_m(n) \\ \vdots \\ u(n-m) \end{bmatrix} \\ &= \mathbf{a}_{m-1}^H \mathbf{u}_m(n) \\ &= f_{m-1}(n) \end{aligned} \quad (4.79)$$

yazılabilir. (4.77), (4.78) ve (4.79) eşitlikleriyle verilen sonuçlar kullanıldığında, $(\mathbf{a}_m^{B*})^H \mathbf{u}_{m+1}(n) = \mathbf{a}_m^{BT} \mathbf{u}_{m+1}(n)$ iç çarpımı için

$$b_m(n) = b_{m-1}(n-1) + \Gamma_m f_{m-1}(n) \quad (4.80)$$

elde edilir.

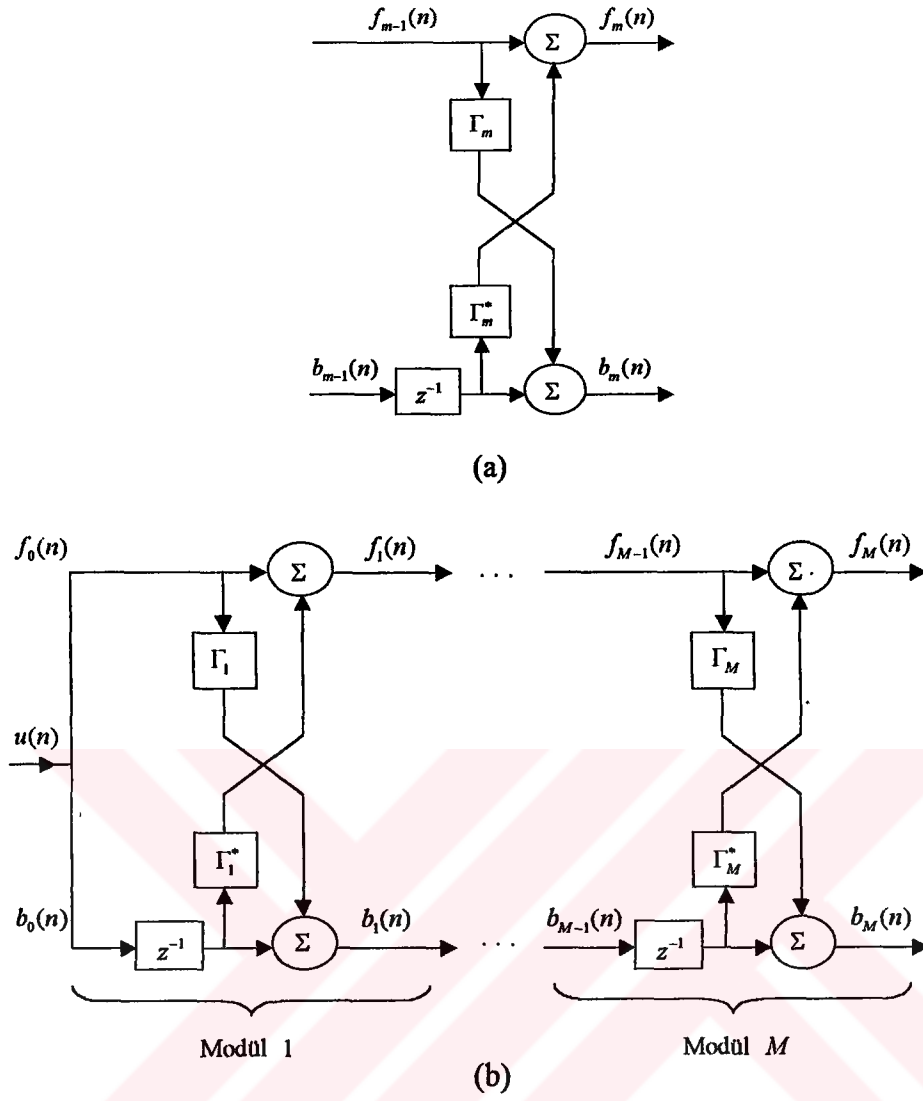
(4.76) ve (4.80) eşitlikleri, kafes öngörücünün m . modülüne ait giriş-çıkış bağıntılarıdır ve *derece güncelleme yinelemeleri* olarak adlandırılır. Bu iki eşitlik, matris formunda da

$$\begin{bmatrix} f_m(n) \\ b_m(n) \end{bmatrix} = \begin{bmatrix} 1 & \Gamma_m^* \\ \Gamma_m & 1 \end{bmatrix} \begin{bmatrix} f_{m-1}(n) \\ b_{m-1}(n-1) \end{bmatrix}, \quad m = 1, 2, \dots, M \quad (4.81)$$

yazılabilir. $b_{m-1}(n)$ geri yönde öngörü hatasına

$$b_{m-1}(n-1) = z^{-1} [b_{m-1}(n)] \quad (4.82)$$

şeklinde z^{-1} birim gecikme operatörü uygulanarak, $b_{m-1}(n-1)$ geri yönde öngörü hatası elde edilir. Diğer bir deyişle, $b_{m-1}(n-1)$ hatası, $b_{m-1}(n)$ geri yönde öngörü hatasının birim geciktirilmişidir. (4.81) ve (4.82) eşitlikleri kullanılarak, kafes öngörücünün m . modülü Şekil 4.3(a)'da verilen işaret akış diyagramıyla gösterilebilir. z^{-1} birim gecikme operatörünü gösteren blok dışında, tüm diyagramın kafes görünümünde olması nedeniyle adına kafes öngörücü denilmektedir.



Şekil 4.3 (a) Kafes öngörücünün m . modülü için işaret akış diyagramı; (b) M . dereceden öngörü hata filtresi için eşdeğer kafes modeli (Haykin,1991)

Her iki şekilden de görülebileceği gibi, Γ_m yansıma katsayısı m . modülün sahip olduğu tek parametredir ve bu parametrenin dışında m . modülü belirleyebilecek başka bir parametre yoktur. Öyleyse, kafes öngörücünün m . modülü Γ_m yansıma katsayısı tarafından tek olarak belirlenir demek yanlış olmaz.

Başlangıç durumu $m = 0$ olduğu için, Levinson–Durbin yinelemesinde (4.68) ile verilen $f_0(n) = b_0(n) = u(n)$ eşitliğine sahip oluruz. $u(n)$, kafes öngörücünün n anındaki girişidir. Böylece, $m=1$ ile başlayıp her defasında filtrenin derecesi 1 arttırıldığında, M . dereceden öngörü hata filtresinin Şekil 4.3(b)’de gösterilen eşdeğer kafes modeline ulaşılmış olur. Bu şekle göre; M modüllü bir kafes filtreyi

oluşturabilmek için, her bir modülünde bir tanesi kullanılmak üzere $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ yansıma katsayılarının bilinmesi gerekir. Kafes filtrenin daha çok tercih edilmesinin nedenleri:

1. Kafes filtre, ileri yönde öngörü hata dizisi $f_1(n), f_2(n), \dots, f_M(n)$ ile bu diziye karşı gelen geri yönde öngörü hata dizisini $b_1(n), b_2(n), \dots, b_M(n)$ birlikte üreten oldukça verimli bir yapıya sahiptir. Örneğin; $b_1(n), b_2(n), \dots, b_M(n)$ hata dizisi enine filtre kullanılarak üretilmek istenirse, toplam M adet ayrı filtre kullanılması gerekir. Çünkü, dizideki her bir hata farklı dereceli bir enine filtre tarafından üretilebilir, yani $b_1(n)$ hatasını üretmek için 1. dereceden, $b_2(n)$ hatası için 2. dereceden ve diğer hataları üretmek için de benzer şekilde farklı dereceden enine filtrelerin kullanılması gerekir. Oysa, M modüllü (veya M . dereceden) tek bir kafes filtre kullanılarak, sadece $b_1(n), b_2(n), \dots, b_M(n)$ hata dizisi değil, aynı zamanda buna karşı düşen $f_1(n), f_2(n), \dots, f_M(n)$ hata dizisi de üretilebilmektedir.
2. Kafes filtre *modüler* yapıdadır. Bu nedenle, öngörü derecesini arttırmak için bir veya istenildiği kadar fazla sayıda modül ilave edilebilir. Yapılan bu ilave, daha önceki hesaplamaları hiçbir şekilde etkilemez.
3. Kafes filtrenin modülleri birbirinden ayrılabilir özelliktedir. Çünkü, geniş anlamda durağan girişler için kafes öngörücü tarafından üretilen geri yönde öngörü hataları birbirine dik (orthogonal) olmaktadır. Bu özellik sayesinde, kafes öngörücünün derecesini azaltmak için istenildiği kadar modül çıkarılabilmektedir. İlave işleminde olduğu gibi, modül çıkarma işlemi de daha önce yapılan hesaplamaları etkilemez.
4. Kafes filtrelerin minimum fazlı olup olmadıklarını belirlemek daha kolaydır. Çünkü, bunu belirlemek için sadece, yansıma katsayılarının $|\Gamma_m| < 1$ koşulunu sağlayıp sağlamadığına bakmak yeterlidir. Sonsuz impuls yanıtı (IIR) bir kafes filtre için $|\Gamma_m| < 1$ koşulunun sağlanıyor olması, bu filtrenin kararlı olduğu anlamına gelir. Bu özellik, yansıma katsayılarının zamanın bir fonksiyonu olarak değiştiği uyarlamalı filtreleme ortamında oldukça faydalıdır. Çünkü, minimum faz koşulu sayesinde filtre katsayıları, filtrenin transfer fonksiyonuna ait tüm kökler birim daire içinde kalacak şekilde her n anı için zamanda güncellenebilmektedir.
5. Diğer filtrelerle karşılaştırıldığında, kafes filtreler parametre kuantalama işleminden dolayı oluşan kötü etkilere karşı daha az hassasiyet göstermektedir.

6. Kafes filtreyi oluşturan her modülün birbirine benzer yapıda olması nedeniyle, kafes filtreler *çok geniş ölçekli entegrasyon* (*very large scale integration*, VLSI) teknolojisinde kullanıma oldukça yatkındırlar.

4.4.2 Kafes filtrelerin ilişki özellikleri

Kafes filtrenin her bir modülünde ürettiği ileri ve geri yönde öngörü hataları arasında ilginç ilişki (correlation) özellikleri bulunmaktadır (Griffiths, 1977; Makhoul, 1978). Bu özelliklerin hepsi temel olarak, Bölüm 3.4'te anlatılan diklik (orthogonallik) kuralının birer sonucudur.

Özellik – 1 İleri yönde öngörü hatası $f_m(n)$ ile tap giriş vektörü $u(n-1)$ birbirine diktir.

$$E[f_m(n)u^*(n-k)] = 0, \quad 1 \leq k \leq m \quad (4.83)$$

Benzer şekilde, geri yönde öngörü hatası $b_m(n)$ ile tap giriş vektörü $u(n)$ birbirine diktir.

$$E[b_m(n)u^*(n-k)] = 0, \quad 0 \leq k \leq m-1 \quad (4.84)$$

Özellik – 2 $f_m(n)$ ile $u(n)$ girişi arasındaki çapraz ilişki, $b_m(n)$ ile $u(n)$ girişinin zamanda ötelenmişi olan $u(n-m)$ girişi arasındaki çapraz ilişkiye eşittir.

$$E[f_m(n)u^*(n)] = E[b_m(n)u^*(n-m)] = P_m \quad (4.85)$$

P_m ise, $E[|f_m(n)|^2] = E[|b_m(n)|^2] = P_m$ eşitliğiyle tanımlanan öngörü hata gücüdür.

Özellik – 3 Geri yönde öngörü hataları birbirine diktir.

$$E[b_m(n)b_i^*(n)] = \begin{cases} P_m, & m = i \\ 0, & m \neq i \end{cases} \quad (4.86)$$

İleri yönde öngörü hataları ise birbirine dik olmayıp, (4.87) eşitliğinde gösterildiği gibi, birbirleriyle ilişkilidirler.

$$E[f_m(n) f_i^*(n)] = P_m, \quad m \geq i \quad (4.87)$$

Özellik – 4 İleri ve geri yönde öngörü hataları arasında

$$E[f_m(n) b_i^*(n)] = \begin{cases} \Gamma_i^* P_m, & m \geq i \\ 0, & m < i \end{cases} \quad (4.88)$$

şeklinde bir çapraz ilişki vardır.

4.5 Gram-Schmidt Dikleştirme Yöntemi Kullanılarak Geri Yönde Öngörü Hata Dizisinin Üretilmesi

Bu bölümde, kafes filtrelerin çalışma prensibi hakkında daha fazla fikir edinebilmek amacıyla, *Gram-Schmidt dikleştirme yöntemi* ele alınacaktır. Burada bu yöntemin ispatına girilmeyecek, sadece teorisi verilip kafes filtrenin çalışma prensibi ile bu yöntem arasındaki benzerlik açıklanacaktır. Bu yöntemle göre:

Birbiriyle ilişkili ve doğrusal bağımsız $(m+1)$ adet $u_0, u_1, \dots, u_m$ rastlantı değişkenleri verildiğinde, her biri bu rastlantı değişkenlerinin

$$\begin{aligned} b_0 &= u_0 \\ b_1 &= a_{1,1} u_0 + u_1 \\ &\vdots \\ b_m &= a_{m,m} u_0 + a_{m,m-1} u_1 + \dots + u_m \end{aligned} \quad (4.89)$$

şeklinde doğrusal toplamından oluşan birbirine dik $(m+1)$ adet $b_0, b_1, \dots, b_m$ rastlantı değişkenlerini oluşturmak her zaman için mümkündür.

Özellik – 3'te verilen (4.86) eşitliği bize şunu anlatır: bir kafes filtre, geniş anlamda durağan bir süreçten elde edilmiş birbiriyle ilişkili $u(n), u(n-1), \dots, u(n-m)$ giriş dizisini birbirine dik $b_0(n), b_1(n), \dots, b_m(n)$ geri yönde öngörü hataları dizisine dönüştürür. Buna göre, Gram-Schmidt dikleştirme yöntemi ile Şekil 4.3(b)'deki kafes filtrenin $b_0(n), b_1(n), \dots, b_M(n)$ geri yönde öngörü hatalarını üretmesi arasında bir benzerlik vardır. Bu benzerliği ispat etmek için $u(n), u(n-1), \dots, u(n-M)$ giriş dizisine Gram-Schmidt yöntemi uygulanırsa, (4.89) eşitliğinden

$$\begin{aligned}
b_0(n) &= u(n) \\
b_1(n) &= a_{1,1} u(n) + u(n-1) \\
&\vdots \\
b_M(n) &= a_{M,M} u(n) + a_{M,M-1} u(n-1) + \dots + u(n-M)
\end{aligned} \tag{4.90}$$

eşitliği yazılır. (4.68), (4.76) ve (4.80) eşitliklerinden de görülebileceği gibi, (4.90) eşitliğinde $a_{1,1} = \Gamma_1$, $a_{2,1} = \Gamma_1 + \Gamma_2 \Gamma_1^*$ ve $a_{2,2} = \Gamma_2$ şeklinde M . modüle kadar devam edip giden bir ilişki söz konusudur. Öyleyse, şu genellemeler yapılabilir:

1. M modüllü bir kafes filtrenin $u(n)$, $u(n-1)$, ..., $u(n-M)$ giriş dizisini kullanarak $b_0(n)$, $b_1(n)$, ..., $b_M(n)$ geri yönde öngörü hata dizisini üretmesi ile, aynı giriş dizisine Gram-Schmidt dikleştirme yönteminin uygulanması birbirine tamamen eşdeğerdir.

2. Gram-Schmidt yöntemine göre;

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

tap giriş vektörünü

$$\mathbf{b}^T(n) = [b_0(n), b_1(n), \dots, b_M(n)]$$

geri yönde öngörü hata vektörüne dönüştürebilmek için,

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ a_{1,1} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{M,M} & a_{M,M-1} & \dots & 1 \end{bmatrix} \tag{4.91}$$

şeklinde $(M+1) \times (M+1)$ boyutlu bir alt üçgensel matris ile soldan çarpmak gerekir.

$$\mathbf{b}(n) = \mathbf{L} \mathbf{u}(n) \tag{4.92}$$

Ana diyagonalindeki tüm elemanlarının 1 olmasından dolayı, $\mathbf{L}$ alt üçgensel matrisinin determinantı daima 1 olur. Diğer bir deyişle, her zaman için tekil olmayan bir matristir ve tersi alınabilir. $\mathbf{L}$ matrisinin her bir satırındaki sıfırdan farklı elemanlar, derecesi o satır numarasının bir eksiğine eşit olan geri yönde öngörü hata filtresine ait tap ağırlıklarından oluşur.

3. $\mathbf{u}(n)$ giriş vektörü ile $\mathbf{b}(n)$ geri yönde öngörü hata vektörü arasında tam anlamıyla *bire-bir* bir karşılık vardır. Bu sayede, bu iki vektörden biri verildiğinde diğeri hiçbir bilgi kaybı olmaksızın kolayca elde edilebilmektedir.

4.5.1 Tap girişlerine ait ilişki matrisinin pozitif belirli olmasının öngörü hata filtresinin minimum fazlı olmasına eşdeğerliği

Bölüm 4.3.1’de, M . dereceden ileri yönde öngörü hata filtresinin minimum fazlı olabilmesi için, bu filtreye ilişkin $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ yansıma katsayılarının $|\Gamma_m| < 1$ ($m = 1, 2, \dots, M$) koşulunu sağlaması gerektiği açıklanmıştı. Bu bölümde ise, tap girişlerine ait $\mathbf{R}$ ilişki matrisinin pozitif belirli olması ile $|\Gamma_m| < 1$ koşulunun aynı anlama geldiği, diğer bir deyişle, bu iki koşuldan birinin sağlanması halinde diğersinin de sağlanmakta olduğu gösterilecektir.

Bu iki koşulun eşdeğerliği, (4.92) eşitliğindeki dönüşüm ifadesi kullanılarak ispatlanacaktır. Geri yönde öngörü hatalarına ait ilişki matrisi $\mathbf{D}$ ile gösterilsin. (4.86) eşitliğinden,

$$\mathbf{D} = E[\mathbf{b}(n)\mathbf{b}^H(n)] = \text{diag}(P_0, P_1, \dots, P_M) \quad (4.93)$$

olacaktır. $\mathbf{D}$ matrisini $\mathbf{R}$ ve $\mathbf{L}$ matrisleri cinsinden

$$\begin{aligned} \mathbf{D} &= E[\mathbf{L}\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{L}^H] \\ &= \mathbf{L} E[\mathbf{u}(n)\mathbf{u}^H(n)] \mathbf{L}^H \\ &= \mathbf{L} \mathbf{R} \mathbf{L}^H \end{aligned} \quad (4.94)$$

yazmak da mümkündür. Bu durumda, matris cebrindeki şu teoremden yararlanılır:

$\mathbf{R}$ matris, $(M+1) \times (M+1)$ boyutlu pozitif belirli bir matris olsun. O zaman $\mathbf{L} \mathbf{R} \mathbf{L}^H$ matris, $(M+1) \times (M+1)$ boyutlu tekil olmayan herhangi bir $\mathbf{L}$ matris için pozitif belirli olacaktır.

$(M+1) \times (M+1)$ boyutlu $\mathbf{D}$ diyagonal matrisinin pozitif belirli olabilmesi için, diyagonal elemanlarının hepsinin

$$P_m > 0 \quad , \quad m = 0, 1, \dots, M \quad (4.95)$$

şeklinde ayrı ayrı pozitif olması gerekir. $P_0 = E[|b_0(n)|^2] = E[|u(n)|^2]$ olduğu için, $P_0 > 0$ koşulu her zaman geçerlidir. (4.61) eşitliği kullanılarak $P_1 = P_0 (1 - |\Gamma_1|^2)$ yazıldığında, $P_1 > 0$ koşulunun sağlanabilmesi için öncelikle $|\Gamma_1| < 1$ koşulunun sağlanması gerektiği görülür. Bu işlem benzer şekilde diğer m değerleri için de gerçekleştirildiğinde, (4.95) eşitliğinde verilen koşulun eşdeğeri olan

$$|\Gamma_m| < 1 \quad , \quad m = 1, 2, \dots, M \quad (4.96)$$

koşulu elde edilir.

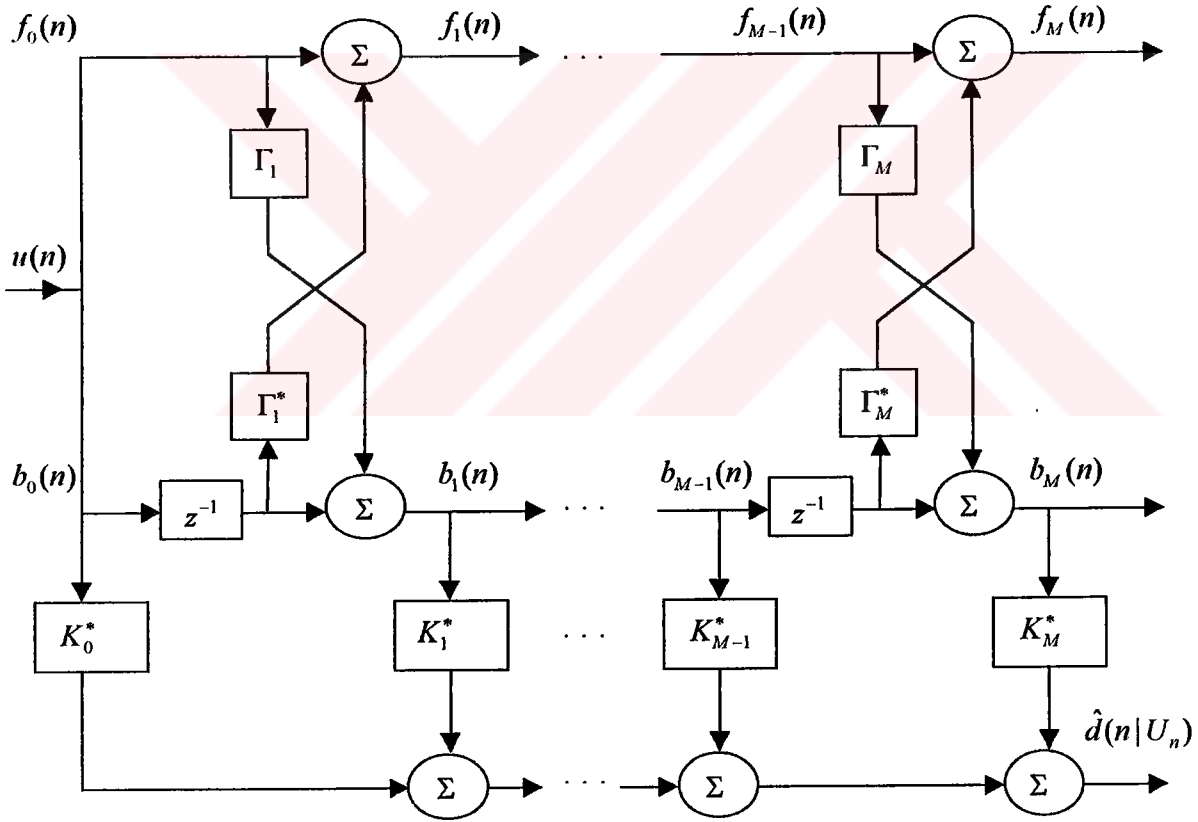
Tüm bunların neticesinde denilebilir ki:

1. $u(n)$ tap giriş vektörüne ait $(M+1) \times (M+1)$ boyutlu $\mathbf{R}$ ilişki matrisi pozitif belirli bir matris ise, $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ yansıma katsayılarının hepsi $|\Gamma_m| < 1$ ($m = 1, 2, \dots, M$) koşulunu sağlar. Bu koşulun sağlanmasına karşılık olarak da, M . dereceden ileri yönde öngörü hata filtresi minimum fazlı olur.
2. Yukarıda söylenenlerin tersi de doğrudur; $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ yansıma katsayılarının hepsinin $|\Gamma_m| < 1$ koşulunu sağlaması durumunda, $u(n)$ tap giriş vektörüne ait $(M+1) \times (M+1)$ boyutlu $\mathbf{R}$ ilişki matrisi pozitif belirli olur.

4.6 Birleşik Süreç Kestirimi

Bu bölümde, karesel ortalama bakımından optimum olan *birleşik süreç kestirimi* (*joint process estimation*) problemini çözmek için, alt sistem olarak kafes öngörücü kullanacaktır. Bölüm 3'te, istenen yanıt olarak adlandırılan $\{d(n)\}$ sürecinin minimum karesel ortalamalı kestiriminin, $\{u(n)\}$ sürecinden elde edilen gözlem değerlerinin doğrudan kullanılmasıyla elde edilebileceği düşünülmüş ve ayrıca $\{d(n)\}$ ile $\{u(n)\}$ süreçlerinin birleşik durağan oldukları kabul edilmişti. Buradaki kestirim probleminde kullanacağımız yaklaşım ise, Bölüm 3'te kullanılan yaklaşıma tek bir fark dışında oldukça benzemektedir. Bu fark ise, istenen yanıtın kestirimini elde etmek için $\{u(n)\}$ sürecine ait gözlem değerlerinin doğrudan değil

de dolaylı yoldan kullanılıyor olmasıdır. Daha açık anlatmak gerekirse, birleşik süreç kestirimi probleminde istenen yanıtın kestirimini elde etmek için $\{u(n)\}$ sürecine ait gözlem değerleri yerine, girişi bu gözlem değerleri ile beslenen bir kafes öngörücü tarafından üretilen geri yönde öngörü hataları kullanılmaktadır. Böyle bir tercih yapılmasının tek nedeni ise, geri yönde öngörü hatalarının birbirine dik olması ve dolayısıyla da, istenen yanıtın kestirimini elde etmek için yapılan hesaplamaların oldukça basitleşmesidir. Birleşik süreç kestirimi için, alt sistem olarak çok modüllü kafes öngörücüyü kullanan bir yapı Şekil 4.4'te gösterilmiştir. Alt sistem (taban) olarak çok modüllü kafes öngörücü kullanılmasaydı, $b_0(n) = u(n)$ olmasının dışında, $b_1(n)$, $b_2(n)$, ..., $b_M(n)$ geri yönde öngörü hatalarının her birini üretebilmek için 1., 2., ..., M . dereceden olmak üzere M . adet enine filtre kullanılması gerekirdi.



Şekil 4.4 Birleşik süreç kestirimi için kafes tabanlı yapı (Haykin,1991)

Bu yapı; ilki, geri yönde öngörü hata dizisi ve ikincisi ise, $\hat{d}(n)$ istenen yanıtının kestirimi olmak üzere iki optimum kestirimi birlikte gerçekleştirdiği için, birleşik süreç kestirici olarak adlandırılır. Şimdi, bu kestirimleri sırasıyla ele alalım:

1. Birleşik süreç kestiricinin girişi, $\Gamma_1, \Gamma_2, \dots, \Gamma_M$ yansıma katsayılarınınca karakterize edilen M modüllü bir kafes öngörücüden oluşur. Kafes öngörücü, öngörme işlemini gerçekleştirerek, birbiriyle ilişkili $u(n), u(n-1), \dots, u(n-M)$ giriş dizisini birbiriyle ilişkisiz $b_0(n), b_1(n), \dots, b_M(n)$ geri yönde öngörü hata dizisine dönüştürür.

2. $K_0, K_1, \dots, K_M$ ağırlıkları ile karakterize edilen *çoklu gerileme filtresi* ise, kafes öngörücünün ürettiği $b_0(n), b_1(n), \dots, b_M(n)$ geri yönde öngörü hata dizisini giriş olarak kullanıp $d(n)$ istenen yanıtının kestirimini üretir. Elde edilen kestirim

$$\hat{d}(n|U_n) = \sum_{i=0}^M K_i^* b_i(n) \quad (4.97)$$

şeklinde tanımlanır. U_n ise, $u(n), u(n-1), \dots, u(n-M)$ girişlerinin oluşturduğu uzayı gösterir. (4.97) eşitliğinin matris formu, kestiricinin $K_0, K_1, \dots, K_M$ gerileme katsayılarını içeren $\mathbf{K}^T = [K_0, K_1, \dots, K_M]$ gerileme katsayı vektörü ile $\mathbf{b}(n)$ geri yönde öngörü hata vektörünün iç çarpımı olarak

$$\hat{d}(n|U_n) = \mathbf{K}^H \mathbf{b}(n) \quad (4.98)$$

yazılır.

Geri yönde öngörü hataları ile istenen yanıt arasındaki çapraz ilişki, $(M+1) \times 1$ boyutlu $\mathbf{s}$ vektör ile gösterilsin. Bu durumda, $\mathbf{s}$ vektör için

$$\mathbf{s} = E[\mathbf{b}(n)d^*(n)] \quad (4.99)$$

eşitliği yazılır. Geri yönde öngörü hatalarına ait ilişki matrisi $\mathbf{D}$ olduğuna göre, (3.22) eşitliğiyle verilen normal denklemleri kullanılarak, *optimum tap ağırlık vektörü* $\mathbf{K}_o$ için

$$\mathbf{D} \mathbf{K}_o = \mathbf{s} \quad (4.100)$$

tanımı yapılabilir. Bu eşitlik $\mathbf{K}_o$ vektörü için çözülürse,

$$\mathbf{K}_o = \mathbf{D}^{-1} \mathbf{s} \quad (4.101)$$

elde edilir.

$\mathbf{D}$ matris, (4.93) ve (4.94) eşitliklerinde de gösterildiği üzere diyagonal bir matristir ve her diyagonal matris gibi

$$\mathbf{D}^k = \text{diag} (P_0^k, P_1^k, \dots, P_M^k) \quad (4.102)$$

eşitliğini sağlayacağı için,

$$\mathbf{D}^{-1} = \text{diag} (P_0^{-1}, P_1^{-1}, \dots, P_M^{-1}) \quad (4.103)$$

olarak yazılabilir. Bu da şunu göstermektedir: Wiener filtre teorisinin enine filtre ile yapılan olağan kestiriminden farklı olarak, birleşik süreç kestirimindeki $\mathbf{K}_o$ tap ağırlık vektörünün hesabı daha basit olmaktadır. Çünkü, $\mathbf{b}(n)$ geri yönde öngörü hata vektörüne ait $\mathbf{D}$ ilişki matrisinin tersini almak, $\mathbf{u}(n)$ giriş vektörüne ait $\mathbf{R}$ ilişki matrisinin tersini almaktan daha kolaydır.

Ana diyagonalinin üstünde kalan tüm elemanları sıfır olan kare matrise alt üçgensel matris (lower triangular matrix), ana diyagonalinin altında kalan tüm elemanları sıfır olan kare matrise ise üst üçgensel matris (upper triangular matrix) denir.

1. Bir alt üçgensel matrisin determinantı, ana diyagonalinde bulunan elemanlarının birbiriyle çarpımına eşittir.
2. Bir alt üçgensel matrisin tersi, bir alt üçgensel matristir.
3. İki alt üçgensel matrisin çarpımı da bir alt üçgensel matristir.

Bu üç özellik, üst üçgensel matrisler için de aynen geçerlidir. Ayrıca, bir alt üçgensel matrisin transpozu bir üst üçgensel matris ve bir üst üçgensel matrisin transpozu ise bir alt üçgensel matris olmaktadır.

Diyagonal matrisler, ana diyagonalinin altında ve üstünde kalan tüm elemanları sıfır olan kare matrislerdir. Buna göre, bir diyagonal matris aynı zamanda hem alt hem de üst üçgensel matris olabilmektedir. Dolayısıyla, yukarıda verilen üç özellik, diyagonal matrisler için de aynen geçerlidir.

5. UYARLAMALI FİLTRELEME

Wiener filtre tasarımı, işlenecek verinin istatistikleri hakkında bir ön bilgiye gerek duyulur. Bu bilgi tam olarak bilinmedikçe, Wiener filtreyi tasarlamak veya tasarlanırsa bile optimum sonuç elde etmek imkansızdır. Çünkü; filtre sadece, girişine gelen verinin istatistiksel karakteristikleri kendisinin tasarımında kullanılan ön bilgiye uyduğu zaman optimum olmaktadır. Bu durumda, *uyarlamalı filtre (adaptive filter)* kullanmak daha uygun bir metottur. Uyarlamalı filtre, girişindeki işaretin istatistikleri hakkında herhangi bir ön bilgiye sahip olmadığı bir ortamda kendi kendini tasarlayarak çalışan bir filtredir. Uyarlamalı filtreye bu özelliği bir *yinelemeli algoritma (recursive algorithm)* kazandırmaktadır. Dolayısıyla, uyarlamalı filtrenin çalışma prensibini yinelemeli algoritma belirlemektedir. Algoritma çalışmaya, içinde bulunduğu ortamdan bağımsız olarak önceden belirlenmiş bir takım *başlangıç koşulları (initial conditions)* ile başlar. Durağan ortamlarda, algoritmanın birbiri ardına gelen iterasyonlar sonunda optimum Wiener çözümüne yakınsadığı görülür. Durağan olmayan ortamlarda ise, algoritma *izleme (tracking)* yeteneğine sahiptir ve bu sayede, giriş verisinin istatistiklerinde zamanla meydana gelen değişimleri (bu değişimlerin yeterince yavaş olması koşulu ile) rahatlıkla izleyebilir.

Uyarlamalı filtrenin parametreleri, yinelemeli algoritma tarafından bir iterasyondan bir sonrakine sürekli olarak güncelleştirilmektedir. Bunun sonucunda da, filtre parametreleri *veri bağımlı* olmaktadır. Bu ise, uyarlamalı filtrenin süperpozisyon ilkesine uymaması bakımından *doğrusal olmayan* bir devre olduğu anlamına gelir. Diğer yandan, filtre çıkışındaki işaretin filtre girişine uygulanan gözlem değerlerinin doğrusal toplamından oluşmasından dolayı, uyarlamalı filtre *doğrusal* bir devre olarak da kabul edilebilmektedir.

Uyarlamalı filtreler için çeşitli yinelemeli algoritmalar geliştirilmiştir. Önemli olan, her biri bir diğerine göre bazı yönlerden üstün olan bu algoritmalar arasından hangisinin yapılacak uygulamaya en uygun olduğunu belirleyebilmektir. Bu nedenle, algoritmaların seçiminde rol oynayan faktörleri açıklamakta fayda vardır:

1. *Yakınsama hızı (rate of convergence)*: Durağan girişler karşısında, algoritmanın karesel ortalama bakımından optimum olan Wiener çözümüne yeterince yakınsaması için gereken iterasyon sayısıdır. Yakınsama hızının yüksek olması, algoritmanın istatistikleri bilinmeyen durağan bir ortama çok çabuk adapte olmasını sağlar.

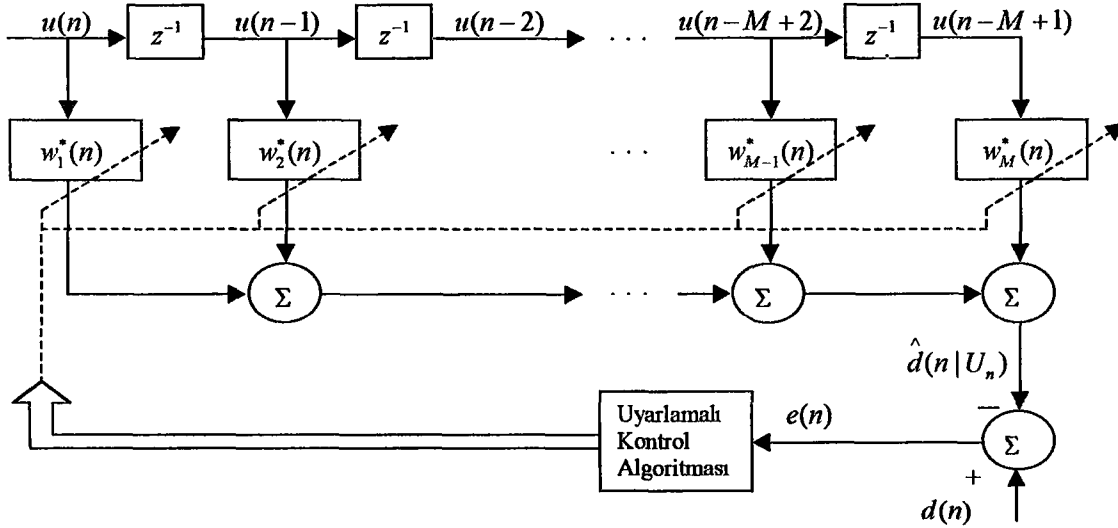
2. *İzleme (tracking)*: Uyarlamalı filtreler durağan olmayan ortamlarda çalıştığı zaman, yinelemeli algoritmanın ortamdaki istatistiksel değişimleri *izlemesi* istenir. Yakınsama hızının yüksek olması, algoritmanın izleme başarısını olumlu yönde etkiler.
3. *Dayanıklılık (robustness)*: Algoritmanın *kötü durumda (ill conditioned)* giriş verisine karşı başarı ile çalışabilmesi anlamına gelir. Verinin *kötü durumda* olmasıyla kastedilen, ilişki matrisinin özdeğer yayılımının büyük olmasıdır.
4. *Hesaplama gereksinimleri*: Algoritmanın tek bir iterasyonu gerçekleştirmek için gerek duyduğu çarpma, bölme, toplama ve çıkarma işlemlerinin sayısı ile, tüm veri ve programı saklayabilmek için gereken bellek miktarı olarak açıklanabilir.
5. *Yapı (structure)*: Yapı ile kastedilen, algoritmadaki bilgi akış yapısıdır. Bilgi akış yapısı, o algoritmanın bir devre halinde uygulanıp uygulanamayacağını belirleyen bir faktördür. Örneğin; bilgi akış yapısı yüksek modülerlik veya paralellik özelliklerine sahip bir algoritma, VLSI teknolojisinin kullanıldığı uygulamalar için oldukça uygundur.

5.1 Uyarlamalı Filtre Yapısı

Uyarlamalı filtre, *uyarlama* ve *filtreleme* olmak üzere iki temel işlemi gerçekleştirir. Uyarlama işleminde, kullanılan yinelemeli algoritmaya bağlı olarak filtrenin tap ağırlıkları otomatik şekilde ayarlanır. Filtreleme işleminde ise; ilk olarak, uyarlama işlemi sonucunda elde edilen tap ağırlıkları kendilerine karşı düşen tap girişleri ile çarpılıp istenen yanıtın kestirimi üretilir ve ardından, üretilen bu kestirim işareti istenen yanıtın çıkarılarak kestirim hatası elde edilir. Uyarlama işleminde tap ağırlıklarının ne şekilde ayarlanması gerektiğini kestirim hatası belirlediği için, bu iki temel işlem birbiriyle karşılıklı ilişki içinde çalışır. Bu nedenle, uyarlama işleminde yapılacak bir hata filtreleme işlemi, filtreleme işleminde yapılacak bir hata ise uyarlama işlemi etkiler.

Uyarlamalı filtre yapısı, Şekil 5.1'deki blok diyagramda da gösterildiği gibi, iki temel bölümden oluşur:

1. n anındaki değerleri $w_1(n)$, $w_2(n)$, ..., $w_M(n)$ olan ayarlanabilir tap ağırlıklarına sahip bir enine filtre,
 2. Bu tap ağırlıklarını uyarlamalı bir yöntemle otomatik olarak ayarlayan bir mekanizma.
- Filtreleme işlemi sırasında, tap girişlerine ilaveten, filtrenin tap ağırlıklarının ayarlanması için referans olarak kullanılan $d(n)$ istenen yanıtının da sağlanması gerekir.



Şekil 5.1 Uyarlamalı enine filtre yapısı (Haykin,1991)

Bölüm 3.1’de olduğu gibi burada da; n anındaki tap girişlerine ait vektör, $\mathbf{u}(n)$ ile ve filtre çıkışında bu girişlere karşı düşen istenen yanıtın kestirimi, $\hat{d}(n|U_n)$ ile gösterilmektedir. U_n ise, $u(n)$, $u(n-1)$, ..., $u(n-M+1)$ tap girişlerinin oluşturduğu M boyutlu uzayı gösterir. İstenen yanıt ile kestirimini birbiriyle karşılaştırarak, $e(n)$ kestirim hatası

$$\begin{aligned} e(n) &= d(n) - \hat{d}(n|U_n) \\ &= d(n) - \mathbf{w}^H(n) \mathbf{u}(n) \end{aligned} \quad (5.1)$$

elde edilir. Tap ağırlık vektörünün ve tap giriş vektörünün açık hali sırasıyla,

$$\mathbf{w}^T(n) = [w_1(n), w_2(n), \dots, w_M(n)] \quad (5.2)$$

$$\mathbf{u}^T(n) = [u(n), u(n-1), \dots, u(n-M+1)] \quad (5.3)$$

şeklinde olmaktadır.

Giriş vektörü $\mathbf{u}(n)$ ile istenen yanıt $d(n)$ birleşik durağan ise, n anındaki karesel ortalama hata $J(n)$ tap ağırlık vektörü $\mathbf{w}(n)$ ’in ikinci dereceden bir fonksiyonu olmaktadır. Böyle bir durum karşısında daha önce elde edilen (3.20) eşitliği kullanılarak,

$$J(n) = \sigma_d^2 - \mathbf{p}^H \mathbf{w}(n) - \mathbf{w}^H(n) \mathbf{p} + \mathbf{w}^H(n) \mathbf{R} \mathbf{w}(n) \quad (5.4)$$

eşitliğini yazmak mümkündür.

Öyleyse, karesel ortalama hatanın tap ağırlık vektörünün elemanlarına olan bağımlılığı, tek bir minimum noktası olan $(M+1)$ boyutlu kase şekilli bir yüzey olarak düşünülebilir. Bu yüzeye, uyarlamalı filtrenin *hata başarıım yüzeyi* denilir. Uyarlama işleminin görevi, bu yüzeyin en altını veya diğer bir deyişle, minimum noktasını sürekli olarak araştırmaktır. Hata başarıım yüzeyinin minimum noktasında tap ağırlık vektörü $\mathbf{w}(n)$, $\mathbf{R}\mathbf{w}_o = \mathbf{p}$ normal denklemleri (3.22) ile tanımlanan $\mathbf{w}_o$ optimum değerini alırken, karesel ortalama hata $J(n)$ ise (3.35) eşitliğinde verilen $J_{\min}$ değerini alır.

Uyarlamalı enine filtrenin durağan bir ortamda çalışması durumunda, hata başarıım yüzeyinin şekli ve yönü sabittir. Bu durumda tek problem, uyarlamalı filtreyi bu yüzeyin minimum noktasında veya bu noktaya çok yakın bir noktada çalışacak şekilde tasarlamaktır. Uyarlamalı enine filtrenin durağan olmayan bir ortamda çalışması durumunda ise, hata başarıım yüzeyinin şekli ve yönü değişkendir; dolayısıyla, minimum noktası sürekli olarak hareket eder. Bu nedenle, girişlerin durağan olmaması durumunda uyarlamalı filtre, yüzeyin minimum noktasını araştırıp bulma görevine ilaveten, bu noktayı bulduktan sonra sürekli olarak izleme (tracking) görevini de üstlenir.

5.2 Yinelemeli En Küçük Kareler Kestirimi

Uyarlamalı filtreleme probleminin tek bir çözümü yoktur. Tek bir çözüm yerine, her biri kendine özgü özellikleri olan çeşitli yinelemeli algoritmalar vardır. Bu nedenle, uyarlamalı filtreleme gerçekleştirecek bir kişinin öncelikle, değişik yinelemeli algoritmaların ne gibi yetenek ve kısıtlamalara sahip olduğunu iyice anlaması ve bundan sonra, bu algoritmalar arasından yapacağı uygulamaya en uygun olanını belirlemesi gerekir. Uyarlamalı filtrelerin çalışması için gerekli yinelemeli algoritmaları elde etmek üzere kullanılan üç farklı metot vardır:

1. Wiener filtre teorisi
2. Kalman filtre teorisi
3. En küçük kareler metodu

Bu üç metot temel olarak kullanılıp, bunlara dayalı bir çok yinelemeli algoritma geliştirilmiştir. Wiener ve Kalman filtre teorileri istatistiksel kavramlar üzerine kurulu olup, ensemble ortalamasına dayalıdır. Klasik en küçük kareler metodunun kullanıldığı üçüncü

metot ise, zaman ortalamasına dayanması ve baştan sona tüm formüllerinin deterministik olması bakımından ilk iki metottan farklıdır. Bu bölümde, uyarlamalı enine filtrenin tasarımı için gerekli yinelemeli algoritma, en küçük kareler metodu kullanılarak elde edilecektir. Sonuçta elde edilen algoritmaya, *yinelemeli en küçük kareler algoritması (recursive least squares algorithm)* veya kısaca *RLS algoritması* denir. RLS algoritmasında, uyarlamalı enine filtrenin tap ağırlık vektörünün $(n-1)$ anındaki en küçük kareler kestirimi kullanılıp, gelen yeni veriye bağlı olarak bu vektörün n anındaki güncelleştirilmiş kestirimi hesaplanmaktadır.

RLS algoritması, Kalman filtre teorisinin deterministik karşılığı olarak düşünülebilir. Ensemble ortalamasına dayalı Kalman filtre teorisinde, karesel ortalama anlamda optimum olan bir enine filtre elde edilir. RLS algoritmasında ise, gelen her yeni giriş veri kümesi için farklı bir filtre oluşturulması nedeniyle, veri uyarlamalı bir enine filtre elde edilmektedir.

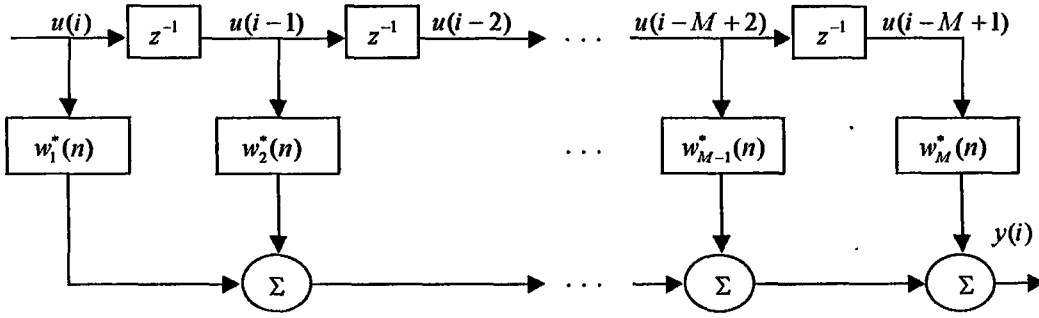
RLS algoritmasının elde edilmesine, en küçük kareler metoduna ait bazı temel ifadelerin ön bilgi olarak verilmesiyle başlanacak ve sonra da, matris cebirinde *matris tersi alma yasası (matrix inversion lemma)* olarak adlandırılan bir ifadeden yararlanılarak devam edilecektir. RLS algoritmasının önemli bir özelliği, algoritmanın içinde bulunduğu andan başlangıç anına kadar giriş verisinde taşınan tüm bilgiyi kullanıyor olmasıdır. Yakınsama hızı, *en küçük karesel ortalama (least mean square, LMS) algoritmasının* yakınsama hızından daha hızlıdır. Ancak, yakınsama hızındaki bu artış, algoritmanın hesaplama karmaşıklığında büyük bir artışa göz yumularak sağlanmaktadır.

5.2.1 Bazı ön bilgiler

En küçük kareler metodunun (RLS algoritması gibi) *yinelemeli* türden tüm uygulamalarında, hesaplamalara önceden belirlenmiş *başlangıç koşulları (initial conditions)* ile başlanır ve eski kestirimleri *güncellemek* için yeni veri örneklerinde taşınan bilgi kullanılır. Bu nedenle, gözlemlenen verinin uzunluğu da değişken olur. Bu durumda, n gözlemlenen verinin değişken olan uzunluğunu göstermek üzere, minimum yapılması gereken *başarı göstergesi* $\xi(n)$ olarak gösterilebilir. $\xi(n)$ başarı göstergesi ifadesinde *ağırlık çarpanı (weighting factor)* veya *unutma çarpanı (forgetting factor)* olarak adlandırılan bir çarpan kullanmak oldukça yaygın olduğu için, $\xi(n)$ başarı göstergesi ifadesi

$$\xi(n) = \sum_{i=1}^n \beta(n, i) |e(i)|^2 \quad (5.5)$$

şeklinde yazılabilir. Burada $e(i)$, istenen yanıt $d(i)$ ile enine filtrenin i anındaki çıkışı $y(i)$ arasındaki farktır. Enine filtre Şekil 5.2'de gösterilmektedir.



Şekil 5.2 Enine filtre (Haykin, 1991)

i anındaki tap girişleri $u(i)$, $u(i-1)$, ..., $u(i-M+1)$ olduğuna göre, enine filtrenin i anındaki tap giriş vektörü $\mathbf{u}(i)$ ve n anındaki tap ağırlık vektörü $\mathbf{w}(n)$

$$\mathbf{u}^T(i) = [u(i), u(i-1), \dots, u(i-M+1)] \quad (5.6)$$

$$\mathbf{w}^T(n) = [w_1(n), w_2(n), \dots, w_M(n)] \quad (5.7)$$

şeklinde olacaktır. Bu durumda $e(i)$,

$$\begin{aligned} e(i) &= d(i) - y(i) \\ &= d(i) - \mathbf{w}^H(n) \mathbf{u}(i) \end{aligned} \quad (5.8)$$

olarak yazılabilir. Tap ağırlıkları, $1 \leq i \leq n$ gözlem aralığı boyunca sabit kalmaktadır.

(5.5) eşitliğindeki ağırlık çarpanı $\beta(n, i)$,

$$0 < \beta(n, i) \leq 1, \quad i = 1, 2, \dots, n \quad (5.9)$$

özellğine sahiptir. Filtrenin durağan olmayan bir ortamda çalışması durumunda, gözlemlenen verinin istatistiksel değişimlerini takip edebilmek için, uzak geçmişteki verinin

unutulması gerekir. Ağırlık çarpanı, bu unutmayı sağlamak amacıyla kullanılmaktadır. Bu nedenle, diğer bir adı da unutma çarpanıdır. Ağırlık çarpanının en çok kullanılan biçimi,

$$\beta(n, i) = \lambda^{n-i}, \quad i = 1, 2, \dots, n \quad (5.10)$$

şeklinde tanımlanan *üstel ağırlık çarpanı* (*exponential weighting factor*) olmaktadır. λ , 1'e yakın ama 1'den küçük pozitif değerli bir sabittir. $\lambda = 1$ olduğu zaman, en küçük kareler metodunun sıradan hali elde edilir. $1/(1-\lambda)$ değeri, algoritma belleğinin yaklaşık bir ölçümüdür. $\lambda = 1$ özel durumunda, algoritma sonsuz belleğe (infinite memory) sahiptir. RLS algoritması için $(1-\lambda)$ değerinin görevi, LMS algoritmasındaki μ adım uzunluğu (step-size) parametresininkine benzer. Bu değere bakılarak, tap ağırlıklarının yavaş mı yoksa hızlı mı uyarlandığı söylenebilir. Öyleyse, *üstel ağırlıklı en küçük kareler metodunda* (*the method of exponentially weighted least squares*) minimum yapılması gereken başarı göstergesi ifadesi

$$\xi(n) = \sum_{i=1}^n \lambda^{n-i} |e(i)|^2 \quad (5.11)$$

olmaktadır. (5.11) eşitliğindeki $\xi(n)$ başarı göstergesine minimum değerini aldirtan tap ağırlık vektörünün optimum değeri $\hat{\mathbf{w}}(n)$,

$$\Phi(n) \hat{\mathbf{w}}(n) = \theta(n) \quad (5.12)$$

normal denklemleri ile tanımlanır. Burada $\Phi(n)$, $M \times M$ boyutlu ilişki matrisidir ve

$$\Phi(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}(i) \mathbf{u}^H(i) \quad (5.13)$$

şeklinde tanımlanmaktadır. $\theta(n)$ ise, enine filtrenin tap girişleri ile istenen yanıt arasındaki $M \times 1$ boyutlu çapraz ilişki vektörüdür ve

$$\theta(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}(i) d^*(i) \quad (5.14)$$

şeklinde tanımlanmaktadır.

(5.13) eşitliğinin sağ tarafındaki toplamda $i = n$ için elde edilen terim bu toplamın dışına çıkarılır ise,

$$\Phi(n) = \lambda \left[\sum_{i=1}^{n-1} \lambda^{n-1-i} \mathbf{u}(i) \mathbf{u}^H(i) \right] + \mathbf{u}(n) \mathbf{u}^H(n) \quad (5.15)$$

eşitliğine ulaşılır. Tanım gereği, köşeli parantez içindeki ifade $\Phi(n-1)$ ilişki matrisine eşit olmaktadır. Öyleyse (5.15) eşitliği,

$$\Phi(n) = \lambda \Phi(n-1) + \mathbf{u}(n) \mathbf{u}^H(n) \quad (5.16)$$

şeklinde yazılabilir. Böylelikle, tap girişlerinin deterministik ilişki matrisini güncellemek için kullanılabilecek bir yineleme elde edilmiş olur. $\mathbf{u}(n) \mathbf{u}^H(n)$ çarpımının düzeltme terimi olarak görev yaptığı bu yineleme, ilişki matrisinin $\Phi(n-1)$ eski değerini güncelleyerek $\Phi(n)$ yeni değerini üretmektedir.

Benzer işlemler (5.14) eşitliği için de yapıldığında, tap girişleri ile istenen yanıt arasındaki $\theta(n)$ deterministik çapraz ilişki vektörünü güncellemek için kullanılabilecek

$$\theta(n) = \lambda \theta(n-1) + \mathbf{u}(n) d^*(n) \quad (5.17)$$

yinelemesi elde edilir. Bu yineleme ise, $\theta(n-1)$ eski değerini güncellemektedir.

$\mathbf{w}(n)$ tap ağırlık vektörünün en küçük kareler kestirimi olan $\hat{\mathbf{w}}(n)$ 'in hesaplanması için, $\Phi(n)$ ilişki matrisinin tersinin bulunması gerekir. Pratikte ve özellikle de tap ağırlıklarının sayısı M çok büyük olduğu zaman, $\Phi(n)$ matrisinin tersini bulmak kaçınılmazı gereken zaman alıcı bir işlemdir. Ayrıca, $\hat{\mathbf{w}}(n)$ 'in (5.16) ve (5.17) eşitliklerindeki gibi, $n = 1, 2, \dots, \infty$ için yinelemeli olarak hesaplanabilmesi istenir. İlki, $\Phi(n)$ ilişki matrisinin tersini bulmak ve ikincisi ise, $\hat{\mathbf{w}}(n)$ için yinelemeli bir ifade elde etmek şeklinde yukarıda açıklanmaya çalışılan bu iki amacın her ikisi de, matris cebrinde *matris tersi alma yasası* olarak bilinen bir temel ifade kullanılarak gerçekleştirilebilir. Yalnız, başlangıç koşullarının $\Phi(n)$ ilişki matrisinin tekil olmayan bir matris olmasını sağlayacak şekilde seçilmiş olduğu kabul edilecektir.

5.2.2 Matris tersi alma yasası

A ve B , $M \times M$; D , $N \times N$ ve C ise, $M \times N$ boyutlu dört matris olsun. Bu matrislerden A , B ve D matrisleri pozitif belirli kabul edilsin. Bu koşullar altında, bu dört matris için

$$A = B^{-1} + C D^{-1} C^H \quad (5.18)$$

eşitliğini yazmak mümkündür. Matris tersi alma yasasına göre, bu şekilde tanımlanan bir A matrisinin tersi

$$A^{-1} = B - B C (D + C^H B C)^{-1} C^H B \quad (5.19)$$

eşitliği ile bulunur. Ters alınamayan bir kare matrisin tersi ile çarpımı I birim matrisine eşit olacağı için, (5.18) ve (5.19) eşitliklerini çarparak yasanın doğruluğu ispatlanabilir. Bir sonraki bölümde, $w(n)$ tap ağırlık vektörünün en küçük kareler kestirimi $\hat{w}(n)$ 'in hesaplanması için gerekli yinelemeli eşitliğin elde edilmesinde, matris tersi alma yasasından nasıl yararlanıldığı gösterilecektir.

5.2.3 Üstel ağırlıklı yinelemeli en küçük kareler algoritması

Tersinin mevcut olması için $\Phi(n)$ ilişki matrisinin pozitif belirli, diğer bir deyişle, tekil olmayan bir matris olması gerekir. Ancak bu şart sağlandığı takdirde, (5.16) yinelemeli eşitliğine matris tersi alma yasası uygulanabilir. Bu şartın sağlandığı varsayılın; bu durumda önce, (5.16) eşitliğini (5.18) eşitliğine benzetebilmek için

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

tanımlamaları yapılır ve sonra, bu tanımlar (5.19) eşitliğinde yerlerine yazılarak $\Phi(n)$ ilişki matrisinin tersi için

$$\Phi^{-1}(n) = \lambda^{-1} \Phi^{-1}(n-1) - \frac{\lambda^{-2} \Phi^{-1}(n-1) u(n) u^H(n) \Phi^{-1}(n-1)}{1 + \lambda^{-1} u^H(n) \Phi^{-1}(n-1) u(n)} \quad (5.20)$$

yinelemeli eşitliği elde edilir. Hesaplamalarda kolaylık sağlaması için,

$$\mathbf{P}(n) = \Phi^{-1}(n) \quad (5.21)$$

$$\mathbf{k}(n) = \frac{\lambda^{-1} \mathbf{P}(n-1) \mathbf{u}(n)}{1 + \lambda^{-1} \mathbf{u}^H(n) \mathbf{P}(n-1) \mathbf{u}(n)} \quad (5.22)$$

şeklinde alınsın. Bu iki eşitlik kullanılarak $\Phi(n)$ ilişki matrisinin tersi için elde edilen (5.20) yinelemeli eşitliği,

$$\mathbf{P}(n) = \lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}^H(n) \mathbf{P}(n-1) \quad (5.23)$$

şeklinde daha kısa yazılabilir. $\mathbf{P}(n)$, $M \times M$ boyutlu bir matris ve $\mathbf{k}(n)$ ise, $M \times 1$ boyutlu bir vektördür. $\mathbf{k}(n)$, *kazanç vektörü* (gain vector) olarak adlandırılmaktadır. $\mathbf{k}(n)$ için yazılan (5.22) eşitliği

$$\begin{aligned} \mathbf{k}(n) &= \lambda^{-1} \mathbf{P}(n-1) \mathbf{u}(n) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}^H(n) \mathbf{P}(n-1) \mathbf{u}(n) \\ &= [\lambda^{-1} \mathbf{P}(n-1) - \lambda^{-1} \mathbf{k}(n) \mathbf{u}^H(n) \mathbf{P}(n-1)] \mathbf{u}(n) \end{aligned} \quad (5.24)$$

şeklinde yeniden yazılırsa, köşeli parantez içinde kalan ifadenin $\mathbf{P}(n)$ 'e eşit olduğu görülür. Öyleyse $\mathbf{k}(n)$ vektörü,

$$\mathbf{k}(n) = \mathbf{P}(n) \mathbf{u}(n) \quad (5.25)$$

şeklinde basitçe yazılabilir. Ayrıca (5.21) eşitliği kullanılarak,

$$\mathbf{k}(n) = \Phi^{-1}(n) \mathbf{u}(n) \quad (5.26)$$

şeklinde de yazılabilir.

5.2.3.1 Tap ağırlık vektörünün zaman güncellemesi

Bu bölümde, $\mathbf{w}(n)$ tap ağırlık vektörünün en küçük kareler kestirimi $\hat{\mathbf{w}}(n)$ 'i zamanda güncelleştirecek yinelemeli bir eşitlik elde edilemeye çalışılacaktır. Bu amaçla ilk önce, (5.17) ve (5.21) eşitlikleri (5.12) eşitliğinde kullanılarak tap ağırlık vektörünün n anındaki en küçük kareler kestirimi $\hat{\mathbf{w}}(n)$ için

$$\begin{aligned}
\hat{\mathbf{w}}(n) &= \Phi^{-1}(n) \boldsymbol{\theta}(n) \\
&= \mathbf{P}(n) \boldsymbol{\theta}(n) \\
&= \lambda \mathbf{P}(n) \boldsymbol{\theta}(n-1) + \mathbf{P}(n) \mathbf{u}(n) d^*(n)
\end{aligned} \tag{5.27}$$

eşitliği elde edilir ve sonra, (5.23) eşitliği sadece (5.27) eşitliğinin son satırındaki ilk $\mathbf{P}(n)$ için yerine yazılarak,

$$\begin{aligned}
\hat{\mathbf{w}}(n) &= \mathbf{P}(n-1) \boldsymbol{\theta}(n-1) - \mathbf{k}(n) \mathbf{u}^H(n) \mathbf{P}(n-1) \boldsymbol{\theta}(n-1) + \mathbf{P}(n) \mathbf{u}(n) d^*(n) \\
&= \Phi^{-1}(n-1) \boldsymbol{\theta}(n-1) - \mathbf{k}(n) \mathbf{u}^H(n) \Phi^{-1}(n-1) \boldsymbol{\theta}(n-1) + \mathbf{P}(n) \mathbf{u}(n) d^*(n) \\
&= \hat{\mathbf{w}}(n-1) - \mathbf{k}(n) \mathbf{u}^H(n) \hat{\mathbf{w}}(n-1) + \mathbf{P}(n) \mathbf{u}(n) d^*(n)
\end{aligned} \tag{5.28}$$

eşitliğine ulaşılır. Son olarak, (5.25) eşitliği (5.28) eşitliğinde yerine yazıldığında, $\hat{\mathbf{w}}(n)$ 'in zaman güncellemesi için elde edilmeye çalışılan yinelemeli eşitlik

$$\begin{aligned}
\hat{\mathbf{w}}(n) &= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n) [d^*(n) - \mathbf{u}^H(n) \hat{\mathbf{w}}(n-1)] \\
&= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n) \alpha^*(n)
\end{aligned} \tag{5.29}$$

bulunmuş olur. Burada $\alpha(n)$,

$$\begin{aligned}
\alpha(n) &= d(n) - \mathbf{u}^T(n) \hat{\mathbf{w}}^*(n-1) \\
&= d(n) - \hat{\mathbf{w}}^H(n-1) \mathbf{u}(n)
\end{aligned} \tag{5.30}$$

eşitliği ile tanımlı *yenileme (innovation)* değeridir. $\hat{\mathbf{w}}(n-1)$, tap ağırlık vektörünün $(n-1)$ anında yapılmış en küçük kareler kestirimidir ve n anında yapılan $\hat{\mathbf{w}}(n)$ kestirimine göre *eski* değerdir. Öyleyse $\hat{\mathbf{w}}^H(n-1) \mathbf{u}(n)$ iç çarpımı, $d(n)$ istenen yanıtının $\hat{\mathbf{w}}(n-1)$ eski kestirimine dayalı bir kestirimidir. Bu durumda $\alpha(n)$, *önceki kestirim hatası (priori estimation error)* olarak da adlandırılabilir.

$$e(n) = d(n) - \hat{\mathbf{w}}^H(n) \mathbf{u}(n) \tag{5.31}$$

eşitliği ile tanımlanan *sonraki kestirim hatası (posteriori estimation error)* $e(n)$, tap ağırlık vektörünün *eski* kestirim değeri $\hat{\mathbf{w}}(n-1)$ yerine *şimdiki* kestirim değeri $\hat{\mathbf{w}}(n)$ 'i içermesi bakımından, önceki kestirim hatası $\alpha(n)$ 'den farklıdır. Bu fark nedeniyle de, $\alpha(n)$ önceki ve $e(n)$ ise sonraki kestirim hatası olarak adlandırılır. Aslında $\alpha(n)$, $e(n)$ 'in tap ağırlık vektörü

güncelleştirilmeden önceki *kesin olmayan (tentative)* değeridir. Diğer yandan, tap ağırlık vektörü için (5.29) yinelemeli eşitliğinin elde edildiği en küçük kareler optimizasyonunda, $\xi(n)$ başarı göstergesinin $\alpha(n)$ 'e göre değil, $e(n)$ 'e göre minimum yapıldığı hatırlanmalıdır.

(5.22), (5.30), (5.29) ve (5.23) eşitlikleri verilen bu sıraya göre hep birlikte RLS algoritmasını oluştururlar. (5.30) eşitliği RLS algoritmasının filtreleme işlemini, (5.29) eşitliği ise uyarlama işlemini tanımlar. Geriye kalan (5.22) ve (5.23) eşitlikleri ise, $\mathbf{k}(n)$ kazanç vektörünü günceller. (5.29) eşitliğine göre tap ağırlık vektörü, eski değeri $\alpha^*(n)$ 'in $\mathbf{k}(n)$ katı kadar artırılarak güncelleştirilmektedir. Bu nedenle de $\mathbf{k}(n)$ vektörüne, kazanç vektörü denilmektedir. RLS algoritmasının önemli bir özelliği, $\Phi(n)$ ilişki matrisinin tersinin direkt olarak alınmayıp, bu işlemin her bir adımda basit bir skaler bölme ile gerçekleştirilmesidir.

5.2.3.2 Başlangıç koşulları

Bölüm 5.2.1'in sonunda da belirtildiği gibi, RLS algoritmasının kullanılabilmesi için, başlangıç koşullarının $\Phi(n)$ ilişki matrisinin tekil olmayan bir matris olmasını sağlayacak şekilde seçilmiş olması gerekir. Bu nedenle, (5.23) yinelemesinin başlangıç değeri $\mathbf{P}(0)$, $\Phi(n)$ matrisini tekil olmayan bir matris yapacak şekilde seçilir. Bunun için, $\Phi(n)$ ilişki matrisine ait (5.13) eşitliğinde

$$\Phi(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}(i) \mathbf{u}^H(i) + \delta \lambda^n \mathbf{I} \quad (5.32)$$

şeklinde önemsiz sayılacak küçük bir değişiklik yapılır. $\mathbf{I}$, $M \times M$ boyutlu birim matris ve δ ise, küçük bir pozitif sabit sayıdır. Yapılan bu küçük değişiklik, sadece başlangıç değerini etkiler; algoritmanın yinelemelerinde herhangi bir bozulmaya neden olmaz. Veri uzunluğu n büyük değerler aldığı zaman, δ sabitinin seçimi genellikle önemsizleşir. $\mathbf{P}(n) = \Phi^{-1}(n)$

Böylelikle geriye sadece, tap ağırlık vektörü $\hat{\mathbf{w}}(n)$ için bir başlangıç koşulunun belirlenmesi kalır. Bunun için ise, artık alışlagelmiş bir hal alan $M \times 1$ boyutlu sıfır vektör

$$\hat{\mathbf{w}}(0) = \mathbf{0} \quad (5.34)$$

tercih edilmektedir.

RLS algoritmasının (5.33) ve (5.34) eşitlikleriyle koşullandırılmasına, *hafif zorlanmış koşullandırma* (*soft-constrained initialization*) denir. Çünkü, bu koşullandırmada seçilmesi gereken tek bir parametre vardır: δ sabiti. RLS algoritması ile yapılan çalışmalar sonucunda, δ pozitif sabitinin $0.01 \sigma_u^2$ değerine eşit olacak kadar küçük seçilmesi önerilmektedir. σ_u^2 , $u(n)$ veri örneğinin değişintisidir. Büyük veri uzunlukları için δ sabitinin seçimi önemsizdir.

5.2.4 Hata karelerinin ağırlıklı toplamını güncelleyen yineleme

Tap ağırlık vektörü $\mathbf{w}(n)$ en küçük kareler kestirimi $\hat{\mathbf{w}}(n)$ 'e eşit olduğu zaman, hata karelerinin ağırlıklı toplamı $\xi(n)$, $\xi_{\min}(n)$ ile gösterilen minimum değerini alır. $\xi_{\min}(n)$ değerini hesaplamak için

$$\xi_{\min}(n) = \xi_d(n) - \mathbf{\theta}^H(n) \hat{\mathbf{w}}(n) \quad (5.35)$$

eşitliği kullanılır. Burada $\xi_d(n)$, istenen yanıt karelerinin

$$\begin{aligned} \xi_d(n) &= \sum_{i=1}^n \lambda^{n-i} |d(i)|^2 \\ &= \lambda \xi_d(n-1) + |d(n)|^2 \end{aligned} \quad (5.36)$$

şeklinde ağırlıklı toplamından oluşmaktadır. (5.17), (5.29) ve (5.36) eşitlikleri (5.35) eşitliğinde yerlerine yazıldığında,

$$\begin{aligned} \xi_{\min}(n) &= \lambda [\xi_d(n-1) - \mathbf{\theta}^H(n-1) \hat{\mathbf{w}}(n-1)] \\ &\quad + d(n) [d^*(n) - \mathbf{u}^H(n) \hat{\mathbf{w}}(n-1)] - \mathbf{\theta}^H(n) \mathbf{k}(n) \alpha^*(n) \end{aligned} \quad (5.37)$$

elde edilir. Tanım gereği, (5.37) eşitliğinin sağ tarafındaki ilk köşeli parantez içindeki ifade

$\xi_{\min}(n-1)$ değerine ve ikinci köşeli parantez içindeki ifade ise $\alpha^*(n)$ değerine eşittir. (5.37) eşitliğinin sonuncu terimindeki $\theta^H(n)\mathbf{k}(n)$ iç çarpımını açıklamak için $\mathbf{k}(n)$ kazanç vektörüne ait (5.26) eşitliği kullanılırsa,

$$\begin{aligned}\theta^H(n)\mathbf{k}(n) &= \theta^H(n)\Phi^{-1}(n)\mathbf{u}(n) \\ &= [\Phi^{-1}(n)\theta(n)]^H \mathbf{u}(n) \\ &= \hat{\mathbf{w}}^H(n)\mathbf{u}(n)\end{aligned}$$

olduğu görülür. İkinci satırın yazılmasında $\Phi(n)$ ilişki matrisinin Hermitian özelliğinden $\Phi^H(n) = \Phi(n)$, üçüncü satırın yazılmasında ise (5.12) eşitliğinden yararlanılmıştır. Bu durumda (5.37) eşitliği,

$$\begin{aligned}\xi_{\min}(n) &= \lambda \xi_{\min}(n-1) + d(n)\alpha^*(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)\alpha^*(n) \\ &= \lambda \xi_{\min}(n-1) + \alpha^*(n)[d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)] \\ &= \lambda \xi_{\min}(n-1) + \alpha^*(n)e(n)\end{aligned}\quad (5.38)$$

şeklinde basitleştirilebilir. Böylece, hata karelerinin ağırlıklı toplamını güncellemek için aranan yineleme elde edilmiş olur. (5.11) eşitliğine bakıldığında, $\xi(n)$ 'in her zaman için gerçel değerler alacağı görülür. Öyleyse, $\xi_{\min}(n)$ ve dolayısıyla $\xi_{\min}(n-1)$ ifadeleri de gerçel değerli olmak zorundadır. Bu durumda, (5.38) yinelemesinde düzeltme terimi olarak görev yapan $\alpha^*(n)e(n)$ çarpımının her zaman için gerçel değerli olduğu ve bu nedenle de,

$$\alpha^*(n)e(n) = \alpha(n)e^*(n) \quad (5.39)$$

eşitliğinin yazılabileceği ortaya çıkar.

5.2.5 RLS algoritmasının yakınsama analizi

Bu bölümde, RLS algoritmasının yakınsaması

1. $\hat{\mathbf{w}}(n)$ kestiriminin ortalama anlamda (in the mean) yakınsaması
2. $\hat{\mathbf{w}}(n)$ kestiriminin karesel ortalama anlamda (in the mean square) yakınsaması
3. $\alpha(n)$ önceki kestirim hatasının karesel ortalama değerinin yakınsaması

şeklinde belirtilen bu üç yakınsama problemi ele alınarak incelenecektir. Algoritmanın

yakınsama analize geçilmeden önce, $\mathbf{u}(n)$ tap giriş vektörü ile $d(n)$ istenen yanıtı arasında *çoklu doğrusal gerileme modeli (multiple linear regression model)* olarak adlandırılan ve

$$d(n) = e_o(n) + \mathbf{w}_o^H \mathbf{u}(n) \quad (5.40)$$

şeklinde tanımlanan bir eşitlik olduğu kabul edilecektir. Bu eşitlikte $\mathbf{w}_o$, $M \times 1$ boyutlu *gerileme parametre vektörünü (regression parameter vector)* ve $e_o(n)$ ise, *ölçüm hatasını (measurement error)* göstermektedir. Ölçüm hatası süreci $\{e_o(n)\}$, sıfır ortalamalı ve σ^2 değişintili beyaz gürültüdür. Parametre vektörü $\mathbf{w}_o$, sabittir. İkinci kabul olarak, $\lambda = 1$ seçilir. Bu seçim, uyarlamalı enine filtrenin durağan bir ortamda çalıştığını söylemeye eşdeğerdir. Durağan bir ortam için en iyi kalıcı hal sonuçları, tap ağırlıkları yavaş yavaş uyarlandığı zaman elde edilir; bu ise, $\lambda = 1$ seçimine karşı düşen bir koşuldur.

5.2.5.1 Tap ağırlık vektörünün ortalama anlamda yakınsaması

(5.32) eşitliğinde yapıldığı gibi, $\Phi(n)$ zaman ortalamalı ilişki matrisinin formülünde küçük değerli $\delta \mathbf{I}$ terimi ile yapılacak değişiklik, RLS algoritması tarafından üretilen $\hat{\mathbf{w}}(n)$ kestirimine bir *sapma (bias)* getirir. Başka bir deyişle, yapılan bu değişiklik sonucunda $\hat{\mathbf{w}}(n)$ kestiriminin ortalama değeri, $E[\hat{\mathbf{w}}(n)] = \mathbf{w}_o$ yerine

$$E[\hat{\mathbf{w}}(n)] = \mathbf{w}_o + \mathbf{b}(n) \quad (5.41)$$

haline gelir. $\mathbf{w}_o$, çoklu gerileme modelinin parametre vektörünü ve $\mathbf{b}(n)$ ise, sapmayı gösterir. Sapmanın kendisi,

$$\mathbf{b}(n) = -\delta \lambda^n \Phi^{-1}(n) \mathbf{w}_o, \quad n \geq 1 \quad (5.42)$$

eşitliği ile tanımlanır; fakat, başlangıçta yapılan ikinci kabulde $\lambda = 1$ seçildiği için,

$$\mathbf{b}(n) = -\delta \Phi^{-1}(n) \mathbf{w}_o \quad (5.43)$$

haline gelir. Zaten, $\Phi(n)$ matrisine eklenecek küçük değerli terimin (5.32) eşitliğindeki $\delta \lambda^n \mathbf{I}$ yerine $\delta \mathbf{I}$ olmasının nedeni de, $\lambda = 1$ seçilmesidir. Dolayısıyla $\Phi(n)$ matrisi, $\Phi(0) = \delta \mathbf{I}$ başlangıç değeri ile koşullandırılmış olur.

Tap girişlerinin zaman ortalamalı ilişki matrisi $\Phi(n)$, $\lambda = 1$ için

$$\Phi(n) = \sum_{i=1}^n \mathbf{u}(i) \mathbf{u}^H(i) \quad (5.44)$$

haline gelir. Bir rasgele sürecin ergodik olabilmesi için, öncelikle o rasgele sürecin durağan olması zorunludur. Öyleyse, her ergodik süreç aynı zamanda durağandır. Buna karşılık, durağan her rasgele süreç ergodik olmayabilir. Bir rasgele süreç ergodik ise, o rasgele sürecin ensemble ortalamalı ifadelerinin yerine bu ifadelerle karşı düşen zaman ortalamalı ifadeler yaklaşık olarak eşit alınabilir. Örneğin; $\mathbf{u}(n)$ tap giriş vektörüyle sunulan $\{\mathbf{u}(n)\}$ rasgele süreci, özilişki matrisi bakımından ergodik olsun. Bu durumda, $\mathbf{u}(n)$ vektörünün $\mathbf{R}$ ilişki matrisi için ensemble ortalamalı $E[\mathbf{u}(n) \mathbf{u}^H(n)]$ ifadesi yerine bu ifadeye karşı düşen

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \mathbf{u}(i) \mathbf{u}^H(i) \quad (5.45)$$

şeklindeki zaman ortalamalı ifade yaklaşık olarak eşit alınabilir. (5.45) eşitliği, n veri uzunluğunun çok büyük olması durumunda

$$\mathbf{R} \cong \frac{1}{n} \sum_{i=1}^n \mathbf{u}(i) \mathbf{u}^H(i) \quad (5.46)$$

şeklinde yazılabilir. Yukarıda anlatılanlara göre; $\mathbf{u}(n)$ tap giriş vektörüyle sunulan $\{\mathbf{u}(n)\}$ rasgele süreci ergodik olduğu zaman, $\mathbf{u}(n)$ tap giriş vektörünün $M \times M$ boyutlu ensemble ortalamalı ilişki matrisi $\mathbf{R}$ ile zaman ortalamalı ilişki matrisi $\Phi(n)$ arasında ($\lambda = 1$ için)

$$\mathbf{R} \cong \frac{1}{n} \Phi(n), \quad \text{büyük } n \text{ değeri için} \quad (5.47)$$

eşitliğinin yazılabileceği ortaya çıkar. (5.47) eşitliği (5.43) eşitliğinde kullanıldığında, $\mathbf{b}(n)$ sapma değeri için

$$\mathbf{b}(n) \cong -\frac{\delta \mathbf{R}^{-1} \mathbf{w}_o}{n}, \quad \text{büyük } n \text{ değeri için} \quad (5.48)$$

sonucu elde edilir. Bu sonuca göre, iterasyon sayısı (veya veri uzunluğu) n sonsuza yaklaştıkça $\mathbf{b}(n)$ sapma değeri de sıfıra yaklaşmaktadır.

Buraya kadar anlatılanlardan şu sonuca ulaşılır: RLS algoritması, $\mathbf{w}_o$ gerileme parametre vektörünün asimptotik olarak sapmasız (unbiased, $\mathbf{b}(n) = 0$) bir kestirimini üretir. Diğer bir deyişle, RLS algoritması ortalama anlamda (in the mean) yakınsak bir algoritmadır.

5.2.5.2 Tap ağırlık vektörünün karesel ortalama anlamda yakınsaması

$\varepsilon(n)$, RLS algoritması tarafından üretilen $\hat{\mathbf{w}}(n)$ kestirimi ile çoklu doğrusal gerileme modeline ait $\mathbf{w}_o$ parametre vektörü arasındaki fark olarak tanımlanan

$$\varepsilon(n) = \hat{\mathbf{w}}(n) - \mathbf{w}_o \quad (5.49)$$

ağırlık hata vektörü (weight-error vector) olsun. $\{e_o(n)\}$ ölçüm hatası sürecinin değişintisi σ^2 ile gösterilmek üzere, ağırlık hata vektörünün ilişki matrisi

$$\begin{aligned} \mathbf{K}(n) &= E[\varepsilon(n) \varepsilon^H(n)] \\ &= \sigma^2 \Phi^{-1}(n) \end{aligned} \quad (5.50)$$

eşitliği ile verilir. Bu bölümde incelenecek problem ise, iterasyon sayısı n sonsuza yaklaştıkça $\mathbf{K}(n)$ ağırlık hata ilişki matrisinin de sıfıra yaklaştığının gösterilmesidir. (5.47) eşitliği göz önüne alındığında, $\mathbf{K}(n)$ için

$$\mathbf{K}(n) \cong \frac{\sigma^2}{n} \mathbf{R}^{-1}, \quad \text{büyük } n \text{ değeri için} \quad (5.51)$$

yaklaşıklığı kullanılabilir. Her iki tarafın normu alınıp $\|\mathbf{R}^{-1}\|$ için (2.62) eşitliğinden yararlanılırsa, $\mathbf{K}(n)$ matrisinin normu

$$\|\mathbf{K}(n)\| \cong \frac{\sigma^2}{n \lambda_{\min}} \quad \text{büyük } n \text{ değeri için} \quad (5.52)$$

şeklinde basitçe ifade edilebilir. Büyük n değerleri için elde edilen (5.52) eşitliğinden iki önemli sonuç çıkarılır:

1. Ağırlık hata ilişki matrisinin normu, en küçük özdeğerin tersi tarafından artırılır. Dolayısıyla, RLS algoritmasının özdeğer yayılımına olan hassasiyeti belirlenirken ilk olarak, en küçük özdeğerin tersinin aldığı değere bakılır. Bu değer büyük ise, yani giriş

verisi kötü durumda ise, en küçük kareler kestirimi de kötü bir yakınsama özelliği sergiler.

2. Ağırlık hata ilişki matrisinin normu, zamanla hemen hemen doğrusal bir şekilde (almost linearly with time) azalır. Bu da gösterir ki; RLS algoritmasının ürettiği $\hat{w}(n)$ kestirimi norm bakımından (in the norm), zamanla hemen hemen doğrusal bir şekilde w_0 parametre vektörüne yakınsamaktadır.

5.2.5.3 RLS algoritmasının karesel ortalama anlamda yakınsaması

RLS algoritmasında, önceki kestirim hatası $\alpha(n)$ ve sonraki kestirim hatası $e(n)$ olmak üzere iki hata değeri vardır. Bölüm 5.2.3.2’de verilen başlangıç koşulları göz önüne alındığında, bu iki hatanın karesel ortalama değerlerinin n zamanı ile birbirlerinden farklı biçimde değiştikleri görülür. $\alpha(n)$ ’in karesel ortalama değeri, $n = 1$ anında $d(n)$ istenen yanıtının karesel ortalama değerine eşit olacak kadar büyük bir değer alır ve sonra, n arttıkça giderek azalır. $e(n)$ ’in karesel ortalama değeri ise, $n = 1$ anında küçük bir değer alır ve sonra, n arttıkça giderek artar. Kestirim hatasının karesel ortalama değerinin n iterasyon sayısına (diğer bir deyişle, zamana) göre çizimine, *öğrenme eğrisi* (*learning curve*) denir. RLS algoritması için de bir öğrenme eğrisi çizilmek istenir ise, bu algoritmanın ürettiği iki ayrı kestirim hatasından hangisinin öğrenme eğrisinin çiziminde kullanılacağına karar verilmesi gerekir. Burada, RLS algoritmasını temel bir algoritma olan LMS algoritması ile öğrenme eğrileri açısından (grafiksel olarak) karşılaştırabilmek amacıyla, $e(n)$ yerine $\alpha(n)$ hatası tercih edilecektir. Çünkü; RLS algoritması için $\alpha(n)$ hatası kullanılarak elde edilen öğrenme eğrisi, LMS algoritmasının öğrenme eğrisi ile aynı şekle sahiptir ve bu benzerlik sayesinde de, karşılaştırma yapma imkanı doğmaktadır. $e(n)$ hatası kullanılarak elde edilen öğrenme eğrisi farklı şekle sahip olduğu için, karşılaştırma yapmak zorlaşır.

Öğrenme eğrisinin, kestirim hatasının karesel ortalama değerinin zamanla değişimini gösteren bir eğri olması ve RLS algoritmasının öğrenme eğrisi için $\alpha(n)$ kestirim hatasının tercih edilmesi, RLS algoritmasının karesel ortalama anlamda yakınsama analizinin $\alpha(n)$ kestirim hatasına dayalı olarak yapılmasını zorunlu kılar.

$d(n)$ istenen yanıtını ortadan kaldırmak amacıyla (5.40) eşitliği (5.30) eşitliğinde yerine konulduğunda, önceki kestirim hatası $\alpha(n)$

$$\begin{aligned}\alpha(n) &= e_o(n) - [\hat{\mathbf{w}}(n-1) - \mathbf{w}_o]^H \mathbf{u}(n) \\ &= e_o(n) - \boldsymbol{\varepsilon}^H(n-1) \mathbf{u}(n)\end{aligned}\quad (5.53)$$

olur. $\boldsymbol{\varepsilon}(n-1)$, $(n-1)$ anındaki ağırlık hata vektörüdür. Wiener filtrelerde, filtrenin başarı göstergesi olarak karesel ortalama hata diye adlandırılan $J(\mathbf{w})$ değeri kullanılmıştı. RLS algoritması için de *istatistiksel başarı göstergesi* olarak, *karesel ortalama hata* kullanılacaktır. Tek fark, burada karesel ortalaması alınacak kestirim hatasının $\alpha(n)$ olması ve de bu değer, $J(\mathbf{w})$ yerine $J'(n)$ ile gösterilmesidir. Öyleyse, RLS algoritmasının istatistiksel başarı göstergesi olarak kullanılacak karesel ortalama hata

$$J'(n) = E[|\alpha(n)|^2] \quad (5.54)$$

şeklinde tanımlanır. Üs işareti, önceki kestirim hatası $\alpha(n)$ ile sonraki kestirim hatası $e(n)$ 'in karesel ortalama değerlerinin birbirine karıştırılmaması için kullanılmaktadır. (5.53) eşitliği (5.54) eşitliğinde yerine konulduğunda,

$$\begin{aligned}J'(n) &= E[|e_o(n)|^2] + E[\mathbf{u}^H(n) \boldsymbol{\varepsilon}(n-1) \boldsymbol{\varepsilon}^H(n-1) \mathbf{u}(n)] \\ &\quad - E[\boldsymbol{\varepsilon}^H(n-1) \mathbf{u}(n) e_o^*(n)] - E[e_o(n) \mathbf{u}^H(n) \boldsymbol{\varepsilon}(n-1)]\end{aligned}\quad (5.55)$$

elde edilir. Başlangıçta, ölçüm hatası $e_o(n)$ sıfır ortalamalı ve σ^2 değişintili kabul edildiği için, (5.55) eşitliğinin sağ tarafındaki ilk beklendik değer σ^2 değişintisine eşit olacaktır. Geriye kalan üç beklendik değer için ise şunlar söylenebilir:

1. $\hat{\mathbf{w}}(n-1)$ kestirimi ve dolayısıyla $\boldsymbol{\varepsilon}(n-1)$ ağırlık hata vektörü, $\mathbf{u}(n)$ tap giriş vektöründen istatistiksel olarak bağımsızdır. Ayrıca, $\mathbf{u}(n)$ tap giriş vektörünün sıfır ortalamalı durağan bir rasgele süreçten gözlemlenen değerlerden oluştuğu kabul edilsin. Bu durumda ikinci beklendik değer, matris cebirinde kullanılan eşitliklerin de yardımıyla,

$$\begin{aligned}E[\mathbf{u}^H(n) \boldsymbol{\varepsilon}(n-1) \boldsymbol{\varepsilon}^H(n-1) \mathbf{u}(n)] &= E[\text{tr}\{\mathbf{u}^H(n) \boldsymbol{\varepsilon}(n-1) \boldsymbol{\varepsilon}^H(n-1) \mathbf{u}(n)\}] \\ &= E[\text{tr}\{\mathbf{u}^H(n) \mathbf{u}(n) \boldsymbol{\varepsilon}(n-1) \boldsymbol{\varepsilon}^H(n-1)\}] \\ &= \text{tr}\{E[\mathbf{u}^H(n) \mathbf{u}(n) \boldsymbol{\varepsilon}(n-1) \boldsymbol{\varepsilon}^H(n-1)]\} \\ &= \text{tr}\{E[\mathbf{u}^H(n) \mathbf{u}(n)] E[\boldsymbol{\varepsilon}(n-1) \boldsymbol{\varepsilon}^H(n-1)]\} \\ &= \text{tr}[\mathbf{R} \mathbf{K}(n-1)]\end{aligned}\quad (5.56)$$

şeklinde yazılabilir. Beklendik değer $E[\cdot]$ ve iz alma $\text{tr}[\cdot]$ işlemlerinin her ikisi de

doğrusal olduğu için, işlem sırası bakımından birbirleriyle yer değiştirebilirler. (5.56) eşitliğinin ikinci satırından üçüncü satırına geçerken bu özellikten yararlanılmıştır. Ayrıca, bir skalerin izi kendisine eşittir. Son satır, ensemble ortalamalı ilişki matrisi $\mathbf{R}$ ve ağırlık hata ilişki matrisi $\mathbf{K}(n)$ için yapılan tanımlar kullanılarak yazılmıştır.

2. (5.40) eşitliğine bakıldığında, ölçüm hatası $e_o(n)$ 'in $\mathbf{u}(n)$ tap giriş vektörüne bağlı olduğu görülür. Bu nedenle $\boldsymbol{\varepsilon}(n-1)$ ağırlık hata vektörü, $\mathbf{u}(n)$ vektörünün yanı sıra $e_o(n)$ ölçüm hatasından da istatistiksel bağımsız olmuş olur. Öyleyse, üçüncü beklendik değer yerine

$$E[\boldsymbol{\varepsilon}^H(n-1)\mathbf{u}(n)e_o^*(n)] = E[\boldsymbol{\varepsilon}^H(n-1)] E[\mathbf{u}(n)e_o^*(n)]$$

eşitliği kullanılabilir. Bölüm 3.4'te anlatılan diklik kuralına göre, $\mathbf{u}(n)$ tap giriş vektörünün her bir elemanı $e_o(n)$ ölçüm (optimum kestirim) hatasına diktir, yani aralarında (3.29) eşitliği geçerlidir. (3.29) eşitliği yukarıdaki eşitlikte yerine yazıldığında, üçüncü beklendik değer için

$$E[\boldsymbol{\varepsilon}^H(n-1)\mathbf{u}(n)e_o^*(n)] = 0 \quad (5.57)$$

sonucu elde edilir.

3. Dördüncü ve son beklendik değer, üçüncü beklendik değer kompleks eşleniğine eşittir. Öyleyse, dördüncü beklendik değer

$$E[e_o(n)\mathbf{u}^H(n)\boldsymbol{\varepsilon}(n-1)] = 0 \quad (5.58)$$

olacaktır.

(5.56), (5.57) ve (5.58) sonuçları (5.55) eşitliğinde yerlerine yazılarak, RLS algoritmasının karesel ortalama hata değeri için

$$J'(n) = \sigma^2 + \text{tr}[\mathbf{R}\mathbf{K}(n-1)] \quad (5.59)$$

şeklinde basit bir formül elde edilir.

Artık sıra, RLS algoritmasının karesel ortalama anlamda yakınsama analizini yapmaya, diğer bir deyişle, büyük n değerleri için (5.59) eşitliğinin yakınsadığı değeri bulmaya

gelmiştir. Bunun için, (5.51) eşitliğinde n yerine $(n-1)$ alınır ve elde edilen $K(n-1)$ ifadesi (5.59) eşitliğinde yerine yazılır. Böylece, $J'(n)$ 'in büyük n değerleri için

$$\begin{aligned} J'(n) &\cong \sigma^2 + \frac{\sigma^2}{n-1} \text{tr}[\mathbf{R} \mathbf{R}^{-1}] \\ &\cong \sigma^2 + \frac{\sigma^2}{n-1} \text{tr}[\mathbf{I}] \end{aligned} \quad (5.60)$$

değerine yakınsadığı bulunur. $\mathbf{R}$ matrisinden dolayı, $\mathbf{I}$ birim matrisi de $M \times M$ boyutludur. Bu nedenle, $\text{tr}[\mathbf{I}] = M$ olur. M , enine filtrenin tap ağırlıklarının sayısıdır. Ayrıca, büyük n değerleri ile çalışıldığı için, $(n-1)$ yerine n alınabilir. Sonuç olarak, RLS algoritması tarafından üretilen *kareseel ortalama hata* için

$$J'(n) \cong \sigma^2 + \frac{M \sigma^2}{n} \quad (5.61)$$

basit formülü elde edilir. (5.61) eşitliğinden şu sonuçlar çıkarılır:

1. RLS algoritması, yaklaşık $n \cong 2M$ iterasyon sonunda kareseel ortalama anlamda yakınsar. Öyleyse, RLS algoritmasının yakınsama hızı LMS algoritmasının yakınsama hızından daha büyüktür. M , enine filtrenin ayarlanabilir katsayılarının (ayarlanabilir tap ağırlıklarının) adedidir.
2. RLS algoritmasının yakınsama hızı, LMS algoritmasından farklı olarak, $u(n)$ tap giriş vektörüne ait $\mathbf{R}$ ilişki matrisinin özdeğer yayılımında meydana gelen değişimlere karşı duyarsızdır.
3. İterasyon sayısı n sonsuza yaklaştıkça, kareseel ortalama hata da son değerine ulaşır. Kareseel ortalama hatanın son değeri, (5.61) eşitliğinden de görüldüğü gibi, ölçüm hatası $e_o(n)$ 'in değişimisine σ^2 eşittir. *Minimum kareseel ortalama hata* da σ^2 değerine eşit olduğuna göre, RLS algoritması durağan bir ortamda çalıştığı zaman, teorik olarak sıfır *artık kareseel ortalama hata* (*excess mean-squared error*) üretiyor demektir.

Yukarıda açıklanan sonuçlar için şu vurgulamaların yapılmasında fayda vardır:

1. RLS algoritmasının LMS algoritmasına karşı yakınsama hızında elde ettiği üstünlük sadece, $e_o(n)$ ölçüm hatası $d(n)$ istenen yanıtından daha küçük olduğu zaman, diğer bir deyişle, işaret-gürültü oranı (signal-to-noise ratio, SNR) yüksek olduğu zaman geçerlidir.

2. RLS algoritmasının sıfır artık karesel ortalama hata ile çalışabilmesi, üstel ağırlık çarpanının $\lambda = 1$ olmasını gerektirir. Bu ise, algoritmanın sonsuz bellekle çalışması demektir.

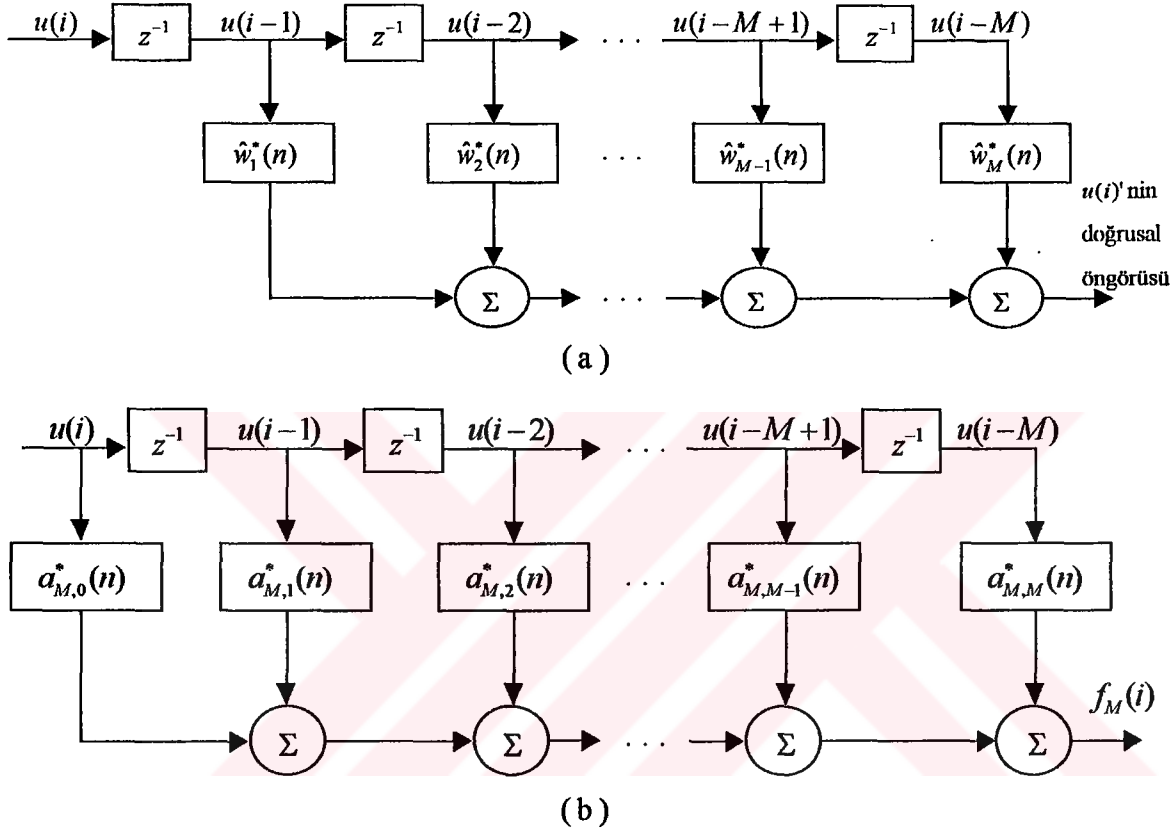
5.3 Uyarlamalı Doğrusal Öngörü

Bölüm 5.2’de, yinelemeli en küçük kareler problemini çözmek için, uyarlamalı filtre yapısı olarak enine filtrenin kullanıldığı *standart* yaklaşım ele alınmış ve RLS algoritması elde edilmiştir. RLS algoritmasının yeteneği, LMS algoritmasına göre daha yüksek yakınsama hızına sahip olmasıdır. Buna karşılık, LMS algoritmasında her bir iterasyon başına gerçekleştirilmesi gereken işlem sayısı (hesaplama gereksinimleri) M ile tanımlanırken, bu sayı RLS algoritması için M^2 olmaktadır. Diğer bir deyişle, RLS algoritmasının yakınsama hızı ile beraber hesaplama karmaşıklığı da yüksek olmaktadır. Özellikle, (5.22) eşitliğiyle verilen $k(n)$ kazanç vektörünü hesaplamak için her bir iterasyonda $M \times M$ boyutlu bir matrisin güncelleştirilip belleğe depo edilmesi gerektiği göz önüne alınırsa, hesaplama karmaşıklığının ne derece yüksek olduğu kolayca anlaşılabilir. Öyleyse RLS algoritması, yüksek yakınsama hızı *yeteneğine* karşılık, yüksek hesaplama karmaşıklığı *kısıtlamasına* sahip bir algoritmadır.

Yüksek hesaplama karmaşıklığı, RLS algoritması ile pratik uygulamalar yapılmasını zorlaştırmaktadır. Bu kısıtlamanın üstesinden gelebilmek için pek çok kişi tarafından, algoritmanın hesaplama yöntemini iyileştirici çeşitli değişiklikler sunulmuştur. Yapılan bu iyileştirici değişiklikler sayesinde, seri haline getirilmiş giriş verisinin *öteleme özelliğini* (*shifting property*) kullanan *hızlı algoritmalar* (*fast algorithms*) elde edilmiştir. Eğer bir algoritmanın hesaplama karmaşıklığı, ayarlanabilir katsayılarının sayısı ile *doğrusal olarak* artıyorsa, o algoritmaya hızlı algoritma denir. Öyleyse hızlı bir algoritma, hesaplama gereksinimleri açısından LMS algoritmasına benzer. Çünkü, her bir iterasyon başına gerçekleştirilmesi gereken işlem sayısı M^2 ile değil, LMS algoritmasındaki gibi M ile değişmektedir. Sonuç olarak; RLS algoritmasının geliştirilmesiyle ortaya çıkan hızlı algoritmalar, RLS algoritmasının eriştiği yüksek yakınsama hızına LMS algoritmasının sahip olduğu düşük hesaplama karmaşıklığı ile ulaşan algoritmalarlardır.

Hızlı algoritmalarından herhangi birinin anlatımına geçmeden önce, bu algoritmaların anlaşılmasına temel teşkil etmesi amacıyla, en küçük kareler anlamında (in the least squares sense) optimize edilen *ileri ve geri yönde uyarlamalı doğrusal öngörücüler* ele alınacaktır.

5.3.1 İleri yönde uyarlamalı doğrusal öngörü



Şekil 5.3 (a) M . dereceden ileri yönde öngörücü; (b) buna karşı düşen öngörü hata filtresi (Haykin, 1991)

Şekil 5.3(a)'da gösterilen M . dereceden ileri yönde doğrusal öngörücü ele alınsın ve tap ağırlık vektörünün $\hat{\mathbf{w}}(n)$, $1 \leq i \leq n$ gözlem aralığı boyunca en küçük kareler anlamında optimize edildiği kabul edilsin. $f_M(i)$, *ileri yönde öngörü hatası* olup, istenen yanıt $u(i)$ ile öngörücünün $\mathbf{u}_M(i-1)$ tap giriş vektörüne karşılık i anında ürettiği cevap arasındaki fark olarak tanımlanır. Bu durumda, $\mathbf{u}_M(i-1)$ tap giriş vektörünün açık hali

$$\mathbf{u}_M^T(i-1) = [u(i-1), u(i-2), \dots, u(i-M)] \quad (5.62)$$

olmak üzere,

$$f_M(i) = u(i) - \hat{\mathbf{w}}^H(n) \mathbf{u}_M(i-1) , \quad 1 \leq i \leq n \quad (5.63)$$

olacaktır. Bu eşitlikten de görüldüğü gibi, $f_M(i)$ hesaplanırken, ileri yönde öngörücüye ait tap ağırlık vektörünün *şimdiki* değeri $\hat{\mathbf{w}}(n)$ kullanılmaktadır. Bu nedenle $f_M(i)$, *sonraki* (*posteriori*) öngörü hatasıdır.

İleri yönde uyarlamalı doğrusal öngörü problemi, Şekil 5.3(a)'daki öngörücünün yanı sıra, Şekil 5.3(b)'de gösterilen öngörü hata filtresi ile de tanımlanabilir. $\mathbf{a}_M(n)$, M . dereceden öngörü hata filtresine ait $(M+1) \times 1$ boyutlu tap ağırlık vektörünü gösterebilir. Bu vektör ile Şekil 5.3(a)'daki öngörücüye ait $\hat{\mathbf{w}}(n)$ tap ağırlık vektörü arasında

$$\mathbf{a}_M(n) = \begin{bmatrix} 1 \\ -\hat{\mathbf{w}}(n) \end{bmatrix} \quad (5.64)$$

şeklinde bir ilişki vardır. İlk elemanını, istenen yanıt $u(i)$ 'nin ve geriye kalan M adet elemanını ise, tap giriş vektörü $\mathbf{u}_M(i-1)$ 'in oluşturduğu

$$\mathbf{u}_{M+1}(i) = \begin{bmatrix} u(i) \\ \mathbf{u}_M(i-1) \end{bmatrix} \quad (5.65)$$

$(M+1) \times 1$ boyutlu $\mathbf{u}_{M+1}(i)$ vektörü tanımlansın. Bu durumda, ileri yönde sonraki öngörü hatası (the forward a posteriori prediction error) $f_M(i)$ için yazılan (5.63) eşitliği,

$$f_M(i) = \mathbf{a}_M^H(n) \mathbf{u}_{M+1}(i) , \quad 1 \leq i \leq n \quad (5.66)$$

haline gelir.

Tap ağırlık vektörü $\hat{\mathbf{w}}(n)$, $1 \leq i \leq n$ aralığında ileri yönde öngörü hata karelerinin ağırlıklı toplamı (the sum of weighted forward prediction error squares) minimum yapılarak elde edilen çözümdür. Öyleyse, ileri yönde doğrusal öngörüye ait genişletilmiş normal denklemleri cinsinden en küçük kareler çözümü,

$$\Phi_{M+1}(n) \mathbf{a}_M(n) = \begin{bmatrix} F_M(n) \\ \mathbf{0} \end{bmatrix} \quad (5.67)$$

şeklinde ifade edilebilir. $\mathbf{0}$, $M \times 1$ boyutlu sıfır vektördür. $F_M(n)$ skaleri ise, *ileri yönde sonraki öngörü hata karelerinin ağırlıklı toplamının (the sum of weighted forward a posteriori prediction error squares) aldığı minimum* değer olup,

$$F_M(n) = \sum_{i=1}^n \lambda^{n-i} |f_M(i)|^2 \quad (5.68)$$

eşitliği ile tanımlanır. $(M+1) \times (M+1)$ boyutlu $\Phi_{M+1}(n)$ matris, öngörü hata filtresinin tap giriş vektörü $\mathbf{u}_{M+1}(i)$ 'ye ait deterministik ilişki matrisidir ve (5.13) eşitliğinin yardımıyla,

$$\Phi_{M+1}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_{M+1}(i) \mathbf{u}_{M+1}^H(i) \quad (5.69)$$

olarak yazılır. $\Phi_{M+1}(n)$ matrisi, $\mathbf{u}_{M+1}(i)$ vektörünün (5.65) eşitliğiyle verilen parçalanmış haline dayanılarak,

$$\Phi_{M+1}(n) = \begin{bmatrix} U_1(n) & \boldsymbol{\theta}_1^H(n) \\ \boldsymbol{\theta}_1(n) & \Phi_M(n-1) \end{bmatrix} \quad (5.70)$$

şeklinde parçalanabilir. $U_1(n)$, ileri yönde öngörüye ait istenen yanıt karelerinin

$$U_1(n) = \sum_{i=1}^n \lambda^{n-i} |u(i)|^2 \quad (5.71)$$

şeklinde ağırlıklı toplamından oluşmaktadır. $\boldsymbol{\theta}_1(n)$, istenen yanıt $u(i)$ ile ileri yönde öngörücüye ait $\mathbf{u}_M(i-1)$ tap giriş vektörü arasındaki $M \times 1$ boyutlu çapraz ilişki vektörüdür ve (5.14) eşitliğinin yardımıyla,

$$\boldsymbol{\theta}_1(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_M(i-1) u^*(i) \quad (5.72)$$

şeklinde yazılır. $\Phi_M(n-1)$ ise, $\mathbf{u}_M(i-1)$ tap giriş vektörünün $M \times M$ boyutlu deterministik ilişki matrisidir ve

$$\begin{aligned}
\Phi_M(n-1) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_M(i-1) \mathbf{u}_M^H(i-1) \\
&= \sum_{i=1}^{n-1} \lambda^{(n-1)-i} \mathbf{u}_M(i) \mathbf{u}_M^H(i)
\end{aligned} \tag{5.73}$$

olarak ifade edilir. (5.73) eşitliği elde edilirken: önce, (5.13) eşitliğinde $\mathbf{u}(i) \mathbf{u}^H(i)$ çarpımı $\mathbf{u}(i-1) \mathbf{u}^H(i-1)$ çarpımı ile yer değiştirilir; çünkü tap giriş vektörü $\mathbf{u}(i)$ değil, $\mathbf{u}(i-1)$ olmaktadır. Sonra, $(i-1)$ yerine i yazılır; bu durumda genel toplamın sınırları, $i = 1$ değerinden $i = 0$ değerine ve $i = n$ değerinden de $i = n-1$ değerine geçecektir. Ancak, Bölüm 5.2'den beri en küçük kareler metodu için ($n = 1, 2, \dots, \infty$ olmak üzere) gözlem aralığının $1 \leq i \leq n$ şeklinde seçilmesi ve $i = 1$ anından önceki giriş verisinin sıfır kabul edilmesi nedeniyle, yeni alt sınırın $i = 0$ yerine $i = 1$ alınmasında herhangi bir sakınca yoktur. Buna karşılık, yeni üst sınır $i = n-1$ aynen kullanılır, değiştirilemez. Sonuçta, (5.73) eşitliğinin son satırı elde edilir. Bu son satır (5.13) eşitliği ile karşılaştırıldığında, $(n-1)$ terimlerinden dolayı, sol tarafının $\Phi_M(n)$ yerine $\Phi_M(n-1)$ olması gerektiği görülür.

Bölüm 5.2'de, uyarlamalı doğrusal kestirim problemini çözmek için en küçük kareler metodu kullanılmış ve sonucunda, RLS algoritmasının genel hali elde edilmişti. Bölüm 5.3.1 ve 5.3.2'de ise, Bölüm 5.2'de ele alınan problemin iki temel formu olan ileri ve geri yönde uyarlamalı doğrusal öngörü problemleri ele alınmakta ve bu iki problem de, en küçük kareler metodu kullanılarak çözülmeye çalışılmaktadır. Bu iki problemin çözümünde elde edilen bağıntılar, hem birbirlerinden ve hem de genel RLS algoritması bağıntılarından farklı olmaktadır. Aynı durumla daha önce Wiener filtre teorisinde karşılaşmış, ortaya çıkan farklılıklar ve benzerlikler Çizelge 4.1 ile gösterilmişti. Buna benzer bir karşılaştırma, Çizelge 5.1'de RLS algoritması için yapılmıştır. Bu çizelge sayesinde, Bölüm 5.2.3 ve 5.2.4'te elde edilen RLS algoritması ifadelerinde gerekli değişiklikler yapılarak, ileri ve geri yönde doğrusal öngörü problemleri için aranılan yinelemeleri yazmak oldukça kolaylaşır.

Çizelge 5.1 Doğrusal kestirim ile ileri ve geri yönde doğrusal öngörülerin RLS algoritması için birbirlerine karşı düşen değişkenlerinin özeti

	Doğrusal kestirim için RLS algoritmasının genel hali	M . dereceden ileri yönde doğrusal öngörü için RLS algoritması	M . dereceden geri yönde doğrusal öngörü için RLS algoritması
Değişkenler			
Tap giriş vektörü	$\mathbf{u}(i)$	$\mathbf{u}_M(i-1)$	$\mathbf{u}_M(i)$
İstenen yanıt	$d(i)$	$u(i)$	$u(i-M)$
Tap ağırlık vektörü	$\hat{\mathbf{w}}(n)$	$\hat{\mathbf{w}}(n)$	$\mathbf{g}(n)$
Sonraki kestirim hatası	$e(n)$	$f_M(n)$	$b_M(n)$
Önceki kestirim hatası	$\alpha(n)$	$\eta_M(n)$	$\psi_M(n)$
Kazanç vektörü	$\mathbf{k}(n)$	$\mathbf{k}_M(n-1)$	$\mathbf{k}_M(n)$
Tap girişlerinin ilişki matrisi	$\Phi(n)$	$\Phi_M(n-1)$	$\Phi_M(n)$
Tap girişleri ile istenen yanıt arasındaki çapraz ilişki vektörü	$\mathbf{0}(n)$	$\mathbf{0}_1(n)$	$\mathbf{0}_2(n)$
Hata karelerinin ağırlıklı toplamının minimum değeri	$\xi_{\min}(n)$	$F_M(n)$	$B_M(n)$

Tap giriş vektörünün $\mathbf{u}(i)$ yerine $\mathbf{u}_M(i-1)$ olması nedeniyle, Şekil 5.3(a)'da gösterilen M . dereceden ileri yönde doğrusal öngörücü için aranılan yinelemeler; (5.26), (5.29) ve (5.30) yinelemelerinde $\mathbf{u}(n)$ yerine $\mathbf{u}_M(n-1)$, $\mathbf{k}(n)$ yerine $\mathbf{k}_M(n-1)$ ve $\Phi(n)$ yerine de $\Phi_M(n-1)$ kullanılarak elde edilebilir. Bu durumda, ileri yönde öngörücünün tap ağırlık vektörünü güncelleyen yineleme

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}_M(n-1) \eta_M^*(n) \quad (5.74)$$

şeklinde olur.

$\eta_M(n)$, *ileri yönde önceki öngörü hatası (the forward a priori prediction error)* veya diğer bir ifadeyle, $f_M(n)$ 'in tap ağırlık vektörü güncelleştirilmeden önceki kesin olmayan (tentative) değeridir. $\eta_M(n)$ için

$$\eta_M(n) = u(n) - \hat{\mathbf{w}}^H(n-1) \mathbf{u}_M(n-1) \quad (5.75)$$

şeklinde $\alpha(n)$ ifadesine benzer bir eşitlik yazılabileceği gibi, bu eşitliğin (5.64) ve (5.65) eşitliklerinden yararlanılarak elde edilen eşdeğeri de

$$\begin{aligned} \eta_M(n) &= \begin{bmatrix} 1, & -\hat{\mathbf{w}}^H(n-1) \end{bmatrix} \begin{bmatrix} u(n) \\ \mathbf{u}_M(n-1) \end{bmatrix} \\ &= \mathbf{a}_M^H(n-1) \mathbf{u}_{M+1}(n) \end{aligned} \quad (5.76)$$

yazılabilir. M . dereceden ileri yönde doğrusal öngörücüye ait kazanç vektörü

$$\mathbf{k}_M(n-1) = \Phi_M^{-1}(n-1) \mathbf{u}_M(n-1) \quad (5.77)$$

olmaktadır. (5.74) eşitliği (5.64) eşitliğinde kullanılarak, Şekil 5.3(b)'de gösterilen M . dereceden ileri yönde öngörü hata filtresinin tap ağırlık vektörünü güncelleyen yineleme

$$\mathbf{a}_M(n) = \mathbf{a}_M(n-1) - \begin{bmatrix} 0 \\ \mathbf{k}_M(n-1) \end{bmatrix} \eta_M^*(n) \quad (5.78)$$

elde edilir. Son olarak, (5.38) eşitliğinden faydalanıp, ileri yönde sonraki öngörü hata karelerinin ağırlıklı toplamının aldığı minimum değeri güncellemek için kullanılan

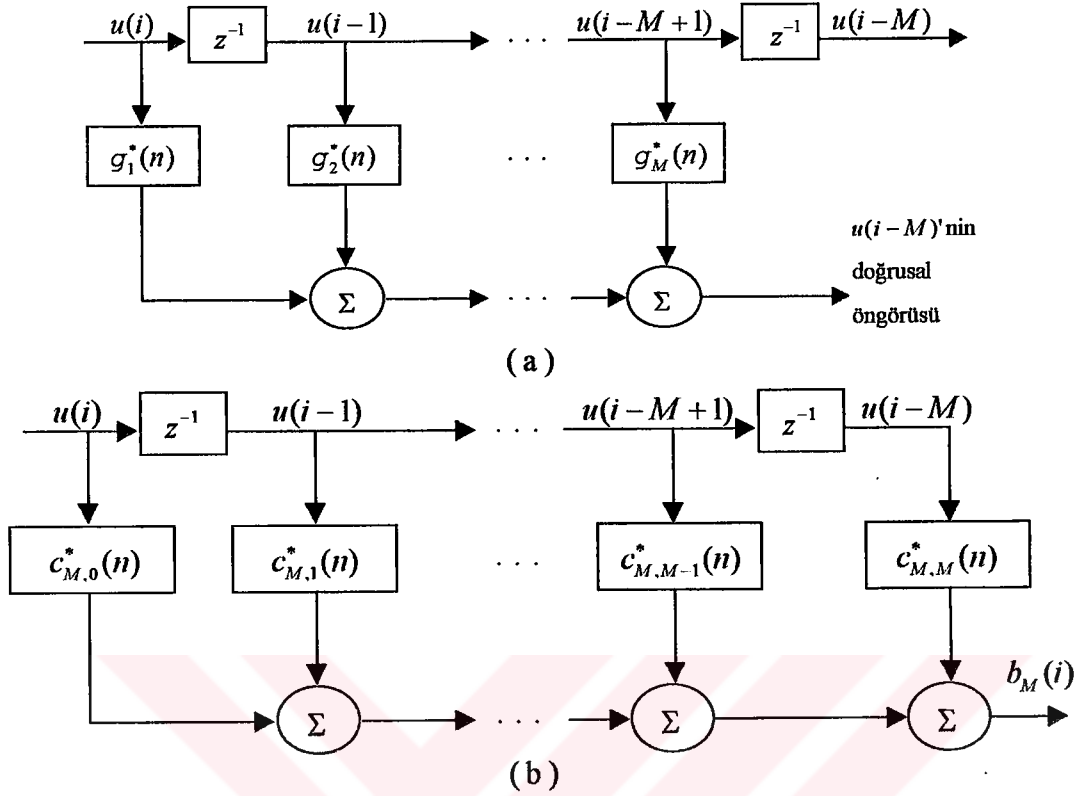
$$F_M(n) = \lambda F_M(n-1) + \eta_M(n) f_M^*(n) \quad (5.79)$$

yinelemesine ulaşılır. (5.38) yinelemesindeki düzeltme terimi için açıklanan nedenlerden dolayı, bu yinelemedeki düzeltme terimi $\eta_M(n) f_M^*(n)$ de her zaman için gerçel değerli bir skalerdir. Bu nedenle, (ileri yönde) önceki ve sonraki öngörü hataları arasında

$$\eta_M(n) f_M^*(n) = \eta_M^*(n) f_M(n) \quad (5.80)$$

eşitliği yazılabilir. Diğer bir ifadeyle, bu hataların iç ve dış çarpımları birbirine eşittir.

5.3.2 Geri yönde uyarlamalı doğrusal öngörü



Şekil 5.4 (a) M . dereceden geri yönde öngörücü; (b) buna karşı düşen öngörü hata filtresi (Haykin,1991)

Şekil 5.4(a)'da gösterilen M . dereceden geri yönde doğrusal öngörücü ele alınsın ve tap ağırlık vektörünün $\mathbf{g}(n)$, $1 \leq i \leq n$ gözlem aralığı boyunca en küçük kareler anlamında optimize edildiği kabul edilsin. $b_M(i)$, geri yönde öngörü hatası olup, istenen yanıt $u(i-M)$ ile öngörücünün $\mathbf{u}_M(i)$ tap giriş vektörüne karşılık i anında ürettiği cevap arasındaki fark olarak tanımlanır. Bu durumda, $\mathbf{u}_M(i)$ tap giriş vektörünün açık hali

$$\mathbf{u}_M^T(i) = [u(i), u(i-1), \dots, u(i-M+1)] \quad (5.81)$$

olmak üzere,

$$b_M(i) = u(i-M) - \mathbf{g}^H(n) \mathbf{u}_M(i), \quad 1 \leq i \leq n \quad (5.82)$$

olacaktır. $b_M(i)$ hesaplanırken, geri yönde öngörücüye ait tap ağırlık vektörünün *şimdiki* değeri $\mathbf{g}(n)$ kullanılmaktadır. Bu nedenle $b_M(i)$, geri yönde sonraki öngörü hatasıdır.

Geri yönde uyarlamalı doğrusal öngörü problemi, Şekil 5.4(a)'daki öngörücünün yanı sıra, Şekil 5.4(b)'de gösterilen öngörü hata filtresi ile de tanımlanabilir. $\mathbf{c}_M(n)$, M . dereceden öngörü hata filtresine ait $(M+1) \times 1$ boyutlu tap ağırlık vektörünü gösterebilir. Bu vektör ile Şekil 5.4(a)'daki öngörücüye ait $\mathbf{g}(n)$ tap ağırlık vektörü arasında

$$\mathbf{c}_M(n) = \begin{bmatrix} -\mathbf{g}(n) \\ 1 \end{bmatrix} \quad (5.83)$$

şeklinde bir ilişki vardır. Son elemanını, istenen yanıt $u(i-M)$ 'in ve geriye kalan M adet elemanını ise, tap giriş vektörü $\mathbf{u}_M(i)$ 'nin oluşturduğu

$$\mathbf{u}_{M+1}(i) = \begin{bmatrix} \mathbf{u}_M(i) \\ u(i-M) \end{bmatrix} \quad (5.84)$$

$(M+1) \times 1$ boyutlu $\mathbf{u}_{M+1}(i)$ vektörü tanımlansın. Bu durumda, geri yönde sonraki öngörü hatası (the backward a posteriori prediction error) $b_M(i)$ için yazılan (5.82) eşitliği,

$$b_M(i) = \mathbf{c}_M^H(n) \mathbf{u}_{M+1}(i), \quad 1 \leq i \leq n \quad (5.85)$$

haline gelir.

Tap ağırlık vektörü $\mathbf{g}(n)$, $1 \leq i \leq n$ aralığında geri yönde öngörü hata karelerinin ağırlıklı toplamı (the sum of weighted backward prediction error squares) minimum yapılarak elde edilen çözümdür. Öyleyse, geri yönde doğrusal öngörüye ait genişletilmiş normal denklemleri cinsinden en küçük kareler çözümü,

$$\Phi_{M+1}(n) \mathbf{c}_M(n) = \begin{bmatrix} \mathbf{0} \\ B_M(n) \end{bmatrix} \quad (5.86)$$

şeklinde ifade edilebilir. $\mathbf{0}$, $M \times 1$ boyutlu sıfır vektördür. $B_M(n)$ skaleri ise, geri yönde sonraki öngörü hata karelerinin ağırlıklı toplamının (the sum of weighted backward a posteriori prediction error squares) aldığı minimum değer olup,

$$B_M(n) = \sum_{i=1}^n \lambda^{n-i} |b_M(i)|^2 \quad (5.87)$$

eşitliği ile tanımlanır. $(M+1) \times (M+1)$ boyutlu $\Phi_{M+1}(n)$ matris, öngörü hata filtresinin tap giriş vektörü $\mathbf{u}_{M+1}(i)$ 'ye ait deterministik ilişki matrisidir ve daha önce, (5.69) eşitliğiyle tanımlanmıştır. $\Phi_{M+1}(n)$ matrisi, $\mathbf{u}_{M+1}(i)$ vektörünün (5.84) eşitliğiyle verilen parçalanmış haline dayanılarak,

$$\Phi_{M+1}(n) = \begin{bmatrix} \Phi_M(n) & \theta_2(n) \\ \theta_2^H(n) & U_2(n) \end{bmatrix} \quad (5.88)$$

şeklinde parçalanabilir. $U_2(n)$, geri yönde öngörüye ait istenen yanıt karelerinin

$$\begin{aligned} U_2(n) &= \sum_{i=1}^n \lambda^{n-i} |u(i-M)|^2 \\ &= \sum_{i=1}^{n-M} \lambda^{(n-M)-i} |u(i)|^2 \end{aligned} \quad (5.89)$$

şeklinde ağırlıklı toplamından oluşmaktadır. $\theta_2(n)$, istenen yanıt $u(i-M)$ ile geri yönde öngörücüye ait $\mathbf{u}_M(i)$ tap giriş vektörü arasındaki $M \times 1$ boyutlu çapraz ilişki vektörüdür ve (5.14) eşitliğinin yardımıyla,

$$\theta_2(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_M(i) u^*(i-M) \quad (5.90)$$

şeklinde yazılır. $\Phi_M(n)$ ise, $\mathbf{u}_M(i)$ tap giriş vektörünün $M \times M$ boyutlu deterministik ilişki matrisidir ve

$$\Phi_M(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_M(i) \mathbf{u}_M^H(i) \quad (5.91)$$

olarak ifade edilir.

Çizelge 5.1'de verilen karşılaştırmalar kullanılarak (5.26), (5.29) ve (5.30) yinelemelerinde yapılan değişiklikler sonucu, geri yönde öngörücünün tap ağırlık vektörünü güncelleyen yineleme

$$\mathbf{g}(n) = \mathbf{g}(n-1) + \mathbf{k}_M(n) \psi_M^*(n) \quad (5.92)$$

olarak elde edilir. $\psi_M(n)$, *geri yönde önceki öngörü hatası (the backward a priori prediction error)* veya diğer bir ifadeyle, $b_M(n)$ 'in tap ağırlık vektörü güncelleştirilmeden önceki kesin olmayan (tentative) değeridir. $\psi_M(n)$ için

$$\psi_M(n) = u(n-M) - \mathbf{g}^H(n-1) \mathbf{u}_M(n) \quad (5.93)$$

şeklinde $\alpha(n)$ ifadesine benzer bir eşitlik yazılabileceği gibi, bu eşitliğin (5.83) ve (5.84) eşitliklerinden yararlanılarak elde edilen eşdeğeri de

$$\begin{aligned} \psi_M(n) &= \begin{bmatrix} -\mathbf{g}^H(n-1), & 1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_M(n) \\ u(n-M) \end{bmatrix} \\ &= \mathbf{c}_M^H(n-1) \mathbf{u}_{M+1}(n) \end{aligned} \quad (5.94)$$

yazılabilir. M . dereceden geri yönde doğrusal öngörücüye ait kazanç vektörü

$$\mathbf{k}_M(n) = \Phi_M^{-1}(n) \mathbf{u}_M(n) \quad (5.95)$$

olmaktadır. (5.92) eşitliği (5.83) eşitliğinde kullanılarak, Şekil 5.4(b)'de gösterilen M . dereceden geri yönde öngörü hata filtresinin tap ağırlık vektörünü güncelleyen yineleme

$$\mathbf{c}_M(n) = \mathbf{c}_M(n-1) - \begin{bmatrix} \mathbf{k}_M(n) \\ 0 \end{bmatrix} \psi_M^*(n) \quad (5.96)$$

elde edilir. Son olarak, (5.38) eşitliğinden faydalanıp, geri yönde sonraki öngörü hata karelerinin ağırlıklı toplamının aldığı minimum değeri güncellemek için kullanılan

$$B_M(n) = \lambda B_M(n-1) + \psi_M(n) b_M^*(n) \quad (5.97)$$

yinelemesine ulaşılır. (5.38) ve (5.79) yinelemelerindeki düzeltme terimlerinde olduğu gibi, bu yinelemedeki düzeltme terimi $\psi_M(n) b_M^*(n)$ de her zaman için gerçel değerli bir skalerdir. Bu nedenle, (geri yönde) önceki ve sonraki öngörü hataları arasında

$$\psi_M(n) b_M^*(n) = \psi_M^*(n) b_M(n) \quad (5.98)$$

eşitliği yazılabilir. Diğer bir ifadeyle, bu hataların iç ve dış çarpımları birbirine eşittir.

Geri yönde doğrusal öngörücü ile geri yönde öngörü hata filtresine ait tap ağırlık vektörlerini güncelleyen (5.92) ve (5.96) yinelemelerine bakıldığında, kazanç vektörünün *şimdiki* değeri $\mathbf{k}_M(n)$ 'in bilinmesi gerektiği görülür. Oysa, ileri yönde doğrusal öngörücü ile ileri yönde öngörü hata filtresine ait tap ağırlık vektörlerini güncelleyen (5.74) ve (5.78) yinelemeleri, kazanç vektörünün *eski* değeri $\mathbf{k}_M(n-1)$ 'in bilinmesini gerektirir.

5.3.3 Genişletilmiş kazanç vektörü

(5.26) ve (5.95) eşitliklerinin yardımıyla, $(M+1) \times 1$ boyutlu *genişletilmiş kazanç vektörü*

$$\mathbf{k}_{M+1}(n) = \Phi_{M+1}^{-1}(n) \mathbf{u}_{M+1}(n) \quad (5.99)$$

olarak tanımlanır. Genişletilmiş kazanç vektörünün en yararlı özelliği, hem ileri yönde uyarlamalı doğrusal öngörüye ait $\mathbf{k}_M(n-1)$ kazanç vektörü cinsinden ve hem de geri yönde uyarlamalı doğrusal öngörüye ait $\mathbf{k}_M(n)$ kazanç vektörü cinsinden yazılabilesidir.

$(M+1) \times (M+1)$ boyutlu $\Phi_{M+1}(n)$ matrisinin (5.70) eşitliğindeki parçalanmış halinin tersi

$$\Phi_{M+1}^{-1}(n) = \begin{bmatrix} 0 & \mathbf{0}_M^T \\ \mathbf{0}_M & \Phi_M^{-1}(n-1) \end{bmatrix} + \frac{1}{F_M(n)} \mathbf{a}_M(n) \mathbf{a}_M^H(n) \quad (5.100)$$

olarak verilmektedir. $\mathbf{0}_M$, $M \times 1$ boyutlu sıfır vektördür. $i = n$ anındaki $(M+1) \times 1$ boyutlu tap giriş vektörünün ileri yönde öngörü için parçalanmış hali, (5.65) eşitliğinde i yerine n yazılarak elde edilir.

$$\mathbf{u}_{M+1}(n) = \begin{bmatrix} u(n) \\ \mathbf{u}_M(n-1) \end{bmatrix} \quad (5.101)$$

(5.100) eşitliğinin her iki yanı (5.101) eşitliği ile sağ taraftan çarpıldığında, genişletilmiş kazanç vektörü için

$$\mathbf{k}_{M+1}(n) = \begin{bmatrix} 0 \\ \mathbf{k}_M(n-1) \end{bmatrix} + \frac{f_M(n)}{F_M(n)} \mathbf{a}_M(n) \quad (5.102)$$

yinelemesine ulaşılır. Bu yinelemenin sağ tarafındaki ikinci terim, (5.66) eşitliğinde i yerine n alınarak yazılmıştır.

$(M+1) \times (M+1)$ boyutlu $\Phi_{M+1}(n)$ matrisinin (5.88) eşitliğindeki parçalanmış halinin tersi

$$\Phi_{M+1}^{-1}(n) = \begin{bmatrix} \Phi_M^{-1}(n) & \mathbf{0}_M \\ \mathbf{0}_M^T & 0 \end{bmatrix} + \frac{1}{B_M(n)} \mathbf{c}_M(n) \mathbf{c}_M^H(n) \quad (5.103)$$

olarak verilmektedir. $i = n$ anındaki $(M+1) \times 1$ boyutlu tap giriş vektörünün geri yönde öngörü için parçalanmış hali, (5.84) eşitliğinde i yerine n yazılarak elde edilir.

$$\mathbf{u}_{M+1}(n) = \begin{bmatrix} \mathbf{u}_M(n) \\ u(n-M) \end{bmatrix} \quad (5.104)$$

(5.103) eşitliğinin her iki yanı (5.104) eşitliği ile sağ taraftan çarpıldığında, genişletilmiş kazanç vektörü için bu defa

$$\mathbf{k}_{M+1}(n) = \begin{bmatrix} \mathbf{k}_M(n) \\ 0 \end{bmatrix} + \frac{b_M(n)}{B_M(n)} \mathbf{c}_M(n) \quad (5.105)$$

yinelemesi elde edilir. Bu yinelemenin sağ tarafındaki ikinci terim, (5.85) eşitliğinde i yerine n alınarak yazılmıştır. (5.102) ve (5.105) yinelemelerine bakıldığında, $\mathbf{k}_{M+1}(n)$ genişletilmiş kazanç vektörünün $\mathbf{k}_M(n-1)$ ve $\mathbf{k}_M(n)$ kazanç vektörleri arasında bağlantı sağladığı görülür. Özellikle, $\mathbf{k}_M(n-1)$ ile $\mathbf{k}_{M+1}(n)$ arasındaki ilişkiyi veren (5.102) yinelemesi, hem derece hem de zaman güncellemesi içerir. Çünkü bu yinelemede, derece olarak M değerinden $(M+1)$ değerine ve zaman olarak da $(n-1)$ değerinden n değerine çıkılmaktadır. (5.105) yinelemesi ise, sadece derece güncellemesi içermektedir.

5.3.4 Dönüşüm çarpanı

Bir enine filtrenin istenen yanıtın kestirimini hangi metoda göre üretmesi isteniyorsa, optimum tap ağırlık vektörünün de o metoda ait normal denklemler çözülerek elde edilmesi gerekir. Buna göre,

$$d(i) = \begin{cases} 1, & i = n \\ 0, & i = 1, 2, \dots, n-1 \end{cases} \quad (5.106)$$

şeklinde tanımlı bu istenen yanıtın en küçük kareler kestirimini üretecek enine filtrenin (Şekil 5.2) optimum tap ağırlık vektörünü $\hat{\mathbf{w}}(n)$ bulmak için, en küçük kareler metoduna ait normal denklemlerin (5.12) çözülmesi gerekir. Bu amaçla, önce (5.106) eşitliği (5.14) eşitliğinde yerine yazılır ve çapraz ilişki vektörü, $\mathbf{u}_M(n) = \mathbf{u}_M(n)$ şeklinde elde edilir. Elde edilen bu ifadeye göre (5.12) normal denklemlerinin $\hat{\mathbf{w}}_M(n)$ için çözümü

$$\hat{\mathbf{w}}_M(n) = \Phi_M^{-1}(n) \mathbf{u}_M(n) \quad (5.107)$$

olarak bulunur. Bu eşitlik (5.95) eşitliği ile karşılaştırıldığında, $\mathbf{k}_M(n) = \hat{\mathbf{w}}_M(n)$ olduğu görülür. Öyleyse, istenen yanıt (5.106) eşitliğindeki gibi olduğu zaman, $\mathbf{k}_M(n)$ kazanç vektörü de (5.12) normal denklemlerinin özel bir çözümü olmaktadır. Tap ağırlık vektörü $\mathbf{k}_M(n)$, tap giriş vektörü $\mathbf{u}_M(n)$ ve istenen yanıtı da (5.106) eşitliğindeki gibi olan M . dereceden bir enine filtrenin *kestirim hatası*

$$\begin{aligned} \gamma_M(n) &= 1 - \mathbf{k}_M^H(n) \mathbf{u}_M(n) \\ &= 1 - \mathbf{u}_M^H(n) \Phi_M^{-1}(n) \mathbf{u}_M(n) \end{aligned} \quad (5.108)$$

olacaktır. (2.16) eşitliği ile Hermitian formunun tanımı yapılmış ve daima negatif olmayan belirli olduğu söylenmişti. $\mathbf{u}_M^H(n) \Phi_M^{-1}(n) \mathbf{u}_M(n)$ terimi de Hermitian formuna sahip olduğu için, daima $\mathbf{u}_M^H(n) \Phi_M^{-1}(n) \mathbf{u}_M(n) \geq 0$ olacaktır. (5.108) eşitliğinde $\Phi_M^{-1}(n)$ ters matrisinin yerine (5.20) eşitliği kullanıldığında,

$$\gamma_M(n) = \frac{1}{1 + \lambda^{-1} \mathbf{u}_M^H(n) \Phi_M^{-1}(n-1) \mathbf{u}_M(n)} \quad (5.109)$$

sonucu bulunur. Bu sonuçta $\mathbf{u}_M^H(n) \Phi_M^{-1}(n-1) \mathbf{u}_M(n) \geq 0$ olduğuna göre, kestirim hatası

$$0 \leq \gamma_M(n) \leq 1 \quad (5.110)$$

sınırları arasında kalan gerçel değerli bir skalerdir.

Doğrusal en küçük kareler kestiriminde üç tür kestirim hatası vardır ve her bir kestirim hatası kendi içinde, önceki ve sonraki olmak üzere ikiye ayrılmaktadır. $\gamma_M(n)$ kestirim hatasına *dönüşüm çarpanı* (*conversion factor*) denilmesinin nedeni, önceki ve sonraki kestirim hataları arasında genel çarpan görevi görmesinden dolayıdır. Şöyle ki:

1. Yinelemeli en küçük kareler kestirimi için

$$\gamma_M(n) = \frac{e_M(n)}{\alpha_M(n)} \quad (5.111)$$

2. İleri yönde uyarlamalı doğrusal öngörü için

$$\gamma_M(n-1) = \frac{f_M(n)}{\eta_M(n)} \quad (5.112)$$

3. Geri yönde uyarlamalı doğrusal öngörü için

$$\gamma_M(n) = \frac{b_M(n)}{\psi_M(n)} \quad (5.113)$$

olmaktadır. $\gamma_M(n)$ dönüşüm çarpanı sayesinde; n anındaki sonraki kestirim hataları $\{e(n), f_M(n), b_M(n)\}$, kendilerinin hesaplanmasında kullanılan $\{\hat{\mathbf{w}}_M(n), \mathbf{a}_M(n), \mathbf{c}_M(n)\}$ tap ağırlık vektörlerinin hesaplanmasına gerek kalmadan, kendilerine karşı düşen $\{\alpha(n), \eta_M(n), \psi_M(n)\}$ önceki kestirim hatalarından kolayca hesaplanabilirler (Carayannis vd., 1983). (5.111), (5.29)'un Hermitian transpozunun sağ taraftan $\mathbf{u}_M(n)$ ile çarpılmasından elde edilen ifadede (5.30), (5.31) ve (5.108) eşitliklerinden; (5.112), (5.78)'in Hermitian transpozunun sağ taraftan $\mathbf{u}_{M+1}(n)$ vektörünün (5.101)'deki parçalanmış hali ile çarpılmasından elde edilen ifadede (5.76), (5.66) ve (5.108) eşitliklerinden; (5.113) ise, (5.96)'nın Hermitian transpozunun sağ taraftan $\mathbf{u}_{M+1}(n)$ vektörünün (5.104)'deki parçalanmış hali ile çarpılmasından elde edilen ifadede (5.94), (5.85) ve (5.108) eşitliklerinden yararlanılarak elde edilmiştir.

Son olarak, dönüşüm çarpanını güncellemek üzere iki ayrı yineleme elde edilecektir. Bunlardan birincisi, (5.102)'nin Hermitian transpozunun sağ taraftan $\mathbf{u}_{M+1}(n)$ vektörünün (5.101)'deki parçalanmış hali ile çarpılmasından elde edilen ifade (5.66) ve (5.108) eşitliklerinden yararlanılarak elde edilen

$$\gamma_{M+1}(n) = \gamma_M(n-1) - \frac{|f_M(n)|^2}{F_M(n)} \quad (5.114)$$

yinelemesi olup, dönüşüm çarpanını hem derece hem de zaman olarak güncellemektedir. İkincisi ise, (5.105)'in Hermitian transpozunun sağ taraftan $\mathbf{u}_{M+1}(n)$ vektörünün (5.104)'deki parçalanmış hali ile çarpılmasından elde edilen ifade (5.85) ve (5.108) eşitliklerinden yararlanılarak elde edilen

$$\gamma_{M+1}(n) = \gamma_M(n) - \frac{|b_M(n)|^2}{B_M(n)} \quad (5.115)$$

yinelemesi olup, dönüşüm çarpanını sadece derece olarak güncellemektedir.

5.4 Yinelemeli En Küçük Kareler Kafes Filtreler

Bu bölümde, en küçük kareler metoduna ait *hızlı algoritmalar* arasından, çok modüllü *kafes öngörücüye* dayalı olan tür ele alınacaktır. Bu türün, öngörü ve kestirim hatalarının önceki veya sonraki değerlerinin kullanılmasına ve yansıma katsayılarının doğrudan veya dolaylı şekilde hesaplanmasına bağlı olarak, dört farklı versiyonu vardır:

1. Öngörü ve kestirim hatalarının sonraki (posteriori) değerlerinin kullanıldığı ve yansıma katsayılarının dolaylı şekilde hesaplandığı versiyon
2. Öngörü ve kestirim hatalarının önceki (priori) değerlerinin kullanıldığı ve yansıma katsayılarının dolaylı şekilde hesaplandığı versiyon
3. Öngörü ve kestirim hatalarının önceki değerlerinin kullanıldığı ve yansıma katsayılarının doğrudan hesaplandığı versiyon
4. Öngörü ve kestirim hatalarının sonraki değerlerinin kullanıldığı ve yansıma katsayılarının doğrudan hesaplandığı versiyon

Teoride, iterasyon sayısı sonsuz kabul edildiği için, bu dört farklı versiyon matematik olarak birbirine eşdeğerdir. Ancak pratikte, iterasyon sayısı sonlu olduğundan, bu dört versiyon

farklı şekilde davranmaktadır. Özellikle, 1. ve 2. versiyonlar *sayısal kararsızlık* (*numerical instability*) gösterirken, 3. ve 4. versiyonlar (ileri ve geri yöndeki yansıma katsayılarının hesabında kullanılan geri besleme sayesinde) *sayısal kararlılık* göstermektedir.

Bu dört versiyonun hepsine birden, *yinelemeli en küçük kareler kafes algoritmaları* (*recursive least squares lattice algorithms*) veya kısaca *RLSL algoritmaları* denir. Bu algoritmalar, hem *derece güncelleme* (*order update*) ve hem de *zaman güncelleme* (*time update*) yinelemeleri içermektedir. Hızlı algoritmalar sınıfının bir üyesi olduğuna göre, RLSL algoritmaları da yüksek yakınsama hızına ve ayarlanabilir katsayılarının sayısı ile doğrusal olarak artan hesaplama karmaşıklığına sahiptir. RLSL algoritmaları, giriş işaretinin ilişki matrisine ait özdeğer yayılımında meydana gelen değişimlere karşı RLS algoritması kadar duyarsızdır. Bu bölümde ele alınan kafes öngörücüler, yansıma katsayılarının *zamanla değişiyor* olması bakımından, Bölüm 4.4'te durağan ortamlar için elde edilen *sabit* yansıma katsayılı kafes öngörücülerden farklıdır. Bu nedenle, bu bölümdeki kafes öngörücüler durağan olmayan ortamlarda da çalışabilirler.

RLSL algoritmalarına yönelik ilk çalışma, Morf (1974) olarak bilinmektedir. Bu çalışmada, öngörücü girişine ait Φ ilişki matrisinin Toeplitz yapıda olmayıp, bunun yerine Toeplitz yapılı iki dikdörtgen matrisin çarpımından oluşan özel bir yapıya sahip olması durumunda en küçük kareler kestirimi için etkili çözümler geliştirilmiştir. RLSL algoritmaları resmi olarak, Morf vd. (1977) ile Morf ve Lee (1978) tarafından sunulmuştur. Bu önemli çalışma, daha detaylı çalışmalara, genelleştirmelere ve uygulamalara öncülük etmiştir (Lee vd., 1981; Satorius ve Pack, 1981; Porat ve Kaitlath, 1983; Lev-Ari vd., 1984; Ling vd., 1985).

5.4.1 Derece güncelleme yinelemeleri

$1 \leq i \leq n$ olmak üzere $\Phi_{m+1}(n)$, m . dereceden ileri yönde öngörü hata filtresine uygulanan $\mathbf{u}_{m+1}(i)$ tap giriş vektörüne ait $(m+1) \times (m+1)$ boyutlu deterministik ilişki matrisi olsun. Bu filtreyi tanımlayacak olan genişletilmiş normal denklemleri, (5.67) eşitliğinden

$$\Phi_{m+1}(n) \mathbf{a}_m(n) = \begin{bmatrix} F_m(n) \\ \mathbf{0}_m \end{bmatrix} \quad (5.116)$$

olarak yazılır. $\mathbf{0}_m$, $m \times 1$ boyutlu sıfır vektördür. m . ve $(m-1)$. dereceden ileri yönde öngörüye ait $\mathbf{a}_m(n)$ ve $\mathbf{a}_{m-1}(n)$ tap ağırlık vektörleri arasında ilişki kurabilmek için, $\Phi_{m+1}(n)$ matrisinin iki değişik parçalanmış halinden (5.88) eşitliğiyle verilmiş olanı

$$\Phi_{m+1}(n) = \begin{bmatrix} \Phi_m(n) & | & \theta_2(n) \\ \hline \theta_2^H(n) & | & U_2(n) \end{bmatrix} \quad (5.117)$$

kullanılır. (5.88) ile (5.117) eşitlikleri arasındaki tek fark, burada M yerine m alınmasıdır. (5.89) eşitliğinde olduğu gibi, $U_2(n)$ 'in burada da $n-m \leq 0$ için sıfır olacağı hatırlanmalıdır. (5.117) eşitliği, ilk elemanı $\mathbf{a}_{m-1}(n)$ vektörü ve son elemanı ise sıfır olan $(m+1) \times 1$ boyutlu bir vektör ile sağ taraftan çarpıldığında,

$$\Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ \hline 0 \end{bmatrix} = \begin{bmatrix} \Phi_m(n) & | & \theta_2(n) \\ \hline \theta_2^H(n) & | & U_2(n) \end{bmatrix} \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ \hline 0 \end{bmatrix} = \begin{bmatrix} \Phi_m(n) \mathbf{a}_{m-1}(n) \\ \hline \theta_2^H(n) \mathbf{a}_{m-1}(n) \end{bmatrix} \quad (5.118)$$

elde edilir. (5.116) eşitliğinden, $(m-1)$. dereceden ileri yönde öngörü hata filtresinin n anındaki genişletilmiş normal denklemleri

$$\Phi_m(n) \mathbf{a}_{m-1}(n) = \begin{bmatrix} F_{m-1}(n) \\ \hline \mathbf{0}_{m-1} \end{bmatrix} \quad (5.119)$$

olacağına göre,

$$\Delta_{m-1}(n) = \theta_2^H(n) \mathbf{a}_{m-1}(n) \quad (5.120)$$

şeklinde bir skalerin tanımlanmasıyla (5.118) eşitliği

$$\Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ \hline 0 \end{bmatrix} = \begin{bmatrix} F_{m-1}(n) \\ \hline \mathbf{0}_{m-1} \\ \hline \Delta_{m-1}(n) \end{bmatrix} \quad (5.121)$$

haline gelir.

$$\Phi_{m+1}(n) \mathbf{c}_m(n) = \begin{bmatrix} \mathbf{0}_m \\ B_m(n) \end{bmatrix} \quad (5.122)$$

genişletilmiş normal denklemleri ile tanımlanan m . dereceden geri yönde öngörü hata filtresi ele alınsın. Bu filtre için ise, $\Phi_{m+1}(n)$ matrisinin iki değişik parçalanmış halinden (5.70) eşitliğiyle verilmiş olanı

$$\Phi_{m+1}(n) = \begin{bmatrix} U_1(n) & | & \boldsymbol{\theta}_1^H(n) \\ \hline & & \\ \boldsymbol{\theta}_1(n) & | & \Phi_m(n-1) \end{bmatrix} \quad (5.123)$$

kullanılır. (5.123) eşitliği, ilk elemanı sıfır olan ve geriye kalan m adet elemanı ise $(m-1)$. dereceden geri yönde öngörü hata filtresine ait $\mathbf{c}_{m-1}(n-1)$ tap ağırlık vektörü ile tanımlanan $(m+1) \times 1$ boyutlu bir vektörü ile sağ taraftan çarpıldığında,

$$\begin{aligned} \Phi_{m+1}(n) \begin{bmatrix} 0 \\ \hline \mathbf{c}_{m-1}(n-1) \end{bmatrix} &= \begin{bmatrix} U_1(n) & | & \boldsymbol{\theta}_1^H(n) \\ \hline & & \\ \boldsymbol{\theta}_1(n) & | & \Phi_m(n-1) \end{bmatrix} \begin{bmatrix} 0 \\ \hline \mathbf{c}_{m-1}(n-1) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{\theta}_1^H(n) \mathbf{c}_{m-1}(n-1) \\ \hline \Phi_m(n-1) \mathbf{c}_{m-1}(n-1) \end{bmatrix} \end{aligned} \quad (5.124)$$

elde edilir. (5.122) eşitliğinden, $(m-1)$. dereceden geri yönde öngörü hata filtresinin $(n-1)$ anındaki genişletilmiş normal denklemleri

$$\Phi_m(n-1) \mathbf{c}_{m-1}(n-1) = \begin{bmatrix} \mathbf{0}_{m-1} \\ B_{m-1}(n-1) \end{bmatrix} \quad (5.125)$$

olacağına göre,

$$\Delta'_{m-1}(n) = \boldsymbol{\theta}_1^H(n) \mathbf{c}_{m-1}(n-1) \quad (5.126)$$

şeklinde ikinci bir skalerin tanımlanmasıyla (5.124) eşitliği

$$\Phi_{m+1}(n) \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} = \begin{bmatrix} \Delta'_{m-1}(n) \\ \mathbf{0}_{m-1} \\ B_{m-1}(n-1) \end{bmatrix} \quad (5.127)$$

haline gelir. (5.120) ve (5.126) eşitliklerini birbirinden ayırt edebilmek için, (5.126) eşitliğinde üs işareti (') kullanılmıştır.

(5.121) ve (5.127) eşitliklerini (5.116)'daki genişletilmiş normal denklemlerine benzeyecek şekilde birleştirebilmek için, (5.127) eşitliği $\Delta_{m-1}(n)/B_{m-1}(n-1)$ oranı ile çarpılıp (5.121) eşitliğinden çıkarılır. Bunun sonucunda elde edilen

$$\Phi_{m+1}(n) \left[\begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} - \frac{\Delta_{m-1}(n)}{B_{m-1}(n-1)} \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} \right] = \begin{bmatrix} F_{m-1}(n) - \frac{\Delta_{m-1}(n) \Delta'_{m-1}(n)}{B_{m-1}(n-1)} \\ \mathbf{0}_m \end{bmatrix} \quad (5.128)$$

eşitliği (5.116) eşitliği ile karşılaştırıldığında,

$$\mathbf{a}_m(n) = \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} - \frac{\Delta_{m-1}(n)}{B_{m-1}(n-1)} \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} \quad (5.129)$$

$$F_m(n) = F_{m-1}(n) - \frac{\Delta_{m-1}(n) \Delta'_{m-1}(n)}{B_{m-1}(n-1)} \quad (5.130)$$

derece güncelleme yinelemelerine ulaşılır.

(5.121) ve (5.127) eşitliklerini (5.122)'deki genişletilmiş normal denklemlerine benzeyecek şekilde birleştirebilmek içinse, (5.121) eşitliği $\Delta'_{m-1}(n)/F_{m-1}(n)$ oranı ile çarpılıp (5.127) eşitliğinden çıkarılır. Bunun sonucunda elde edilen

$$\Phi_{m+1}(n) \left[\begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} - \frac{\Delta'_{m-1}(n)}{F_{m-1}(n)} \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \right] = \begin{bmatrix} \mathbf{0}_m \\ B_{m-1}(n-1) - \frac{\Delta_{m-1}(n) \Delta'_{m-1}(n)}{F_{m-1}(n)} \end{bmatrix} \quad (5.131)$$

eşitliği (5.122) eşitliği ile karşılaştırıldığında,

$$\mathbf{c}_m(n) = \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} - \frac{\Delta'_{m-1}(n)}{F_{m-1}(n)} \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \quad (5.132)$$

$$B_m(n) = B_{m-1}(n-1) - \frac{\Delta_{m-1}(n) \Delta'_{m-1}(n)}{F_{m-1}(n)} \quad (5.133)$$

derece güncelleme yinelemelerine ulaşılır.

5.4.1.1 $\Delta_{m-1}(n)$ ve $\Delta'_{m-1}(n)$ parametreleri arasındaki ilişki

(5.120) ve (5.126) eşitlikleriyle tanımlanan $\Delta_{m-1}(n)$ ve $\Delta'_{m-1}(n)$ parametreleri, birbirlerinin kompleks eşleniğidir. Diğer bir deyişle, aralarında

$$\Delta'_{m-1}(n) = \Delta_{m-1}^*(n) \quad (5.134)$$

şeklinde bir ilişki vardır. Bu ilişki iki aşamada ispatlanabilir:

1. (5.121) eşitliğinin her iki yanı $[0, \mathbf{c}_{m-1}^H(n-1)]$ vektörü ile sol taraftan çarpıldığında,

$$\begin{aligned} [0, \mathbf{c}_{m-1}^H(n-1)] \Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} &= [0, \mathbf{c}_{m-1}^H(n-1)] \begin{bmatrix} F_{m-1}(n) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n) \end{bmatrix} \\ &= \Delta_{m-1}(n) \end{aligned} \quad (5.135)$$

$\Delta_{m-1}(n)$ skaleri elde edilir. Çünkü, $\mathbf{c}_{m-1}(n-1)$ vektörünün son elemanı daima 1 olmaktadır.

2. (5.127) eşitliğinin her iki yanının Hermitian transpozunu alındığında, ilişki matrisi $\Phi_{m+1}(n)$ 'in Hermitian özelliğinden ve $B_{m-1}(n-1)$ 'in gerçel değerli olmasından dolayı,

$$[0, \mathbf{c}_{m-1}^H(n-1)] \Phi_{m+1}(n) = [\Delta_{m-1}'^*(n), \mathbf{0}_{m-1}^T, B_{m-1}(n-1)]$$

sonucu elde edilir. $\mathbf{a}_{m-1}(n)$ vektörünün ilk elemanı her zaman için 1 olduğuna göre,

$$\begin{aligned} [0, \mathbf{c}_{m-1}^H(n-1)] \Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} &= [\Delta_{m-1}'^*(n), \mathbf{0}_{m-1}^T, B_{m-1}(n-1)] \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \\ &= \Delta_{m-1}'^*(n) \end{aligned} \quad (5.136)$$

olacaktır. (5.135) ve (5.136) eşitliklerinin son satırları karşılaştırıldığında, (5.134) eşitliğinin doğruluğu ispatlanmış olur.

5.4.1.2 Sonraki öngörü hataları için derece güncelleme yinelemeleri

Kafes öngörücünün son derecesi M ile gösterilmek üzere, *ileri yönde yansımaya katsayısı*

$$\Gamma_{f,m}(n) = -\frac{\Delta_{m-1}(n)}{B_{m-1}(n-1)}, \quad m = 1, 2, \dots, M \quad (5.137)$$

eşitliğiyle ve *geri yönde yansımaya katsayısı* ise

$$\Gamma_{b,m}(n) = -\frac{\Delta'_{m-1}(n)}{F_{m-1}(n)} = -\frac{\Delta^*_{m-1}(n)}{F_{m-1}(n)}, \quad m = 1, 2, \dots, M \quad (5.138)$$

eşitliğiyle tanımlansın. Genelde $F_{m-1}(n) \neq B_{m-1}(n-1)$ olduğu için, RLSL algoritmalarında

$\Gamma_{f,m}(n) \neq \Gamma^*_{b,m}(n)$ olmaktadır. (5.137) ve (5.138) tanımları kullanılarak, (5.129) ve (5.132)

eşitlikleri sırasıyla

$$\mathbf{a}_m(n) = \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} + \Gamma_{f,m}(n) \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} \quad (5.139)$$

$$\mathbf{c}_m(n) = \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} + \Gamma_{b,m}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \quad (5.140)$$

şeklinde yeniden yazılabilir. (5.139) ve (5.140) eşitlikleri, Levinson–Durbin yinelemesinin deterministik karşılığı olarak düşünülebilir. Ayrıca, yansımaya katsayılarının ve öngörü hata filtrelerine ait tap ağırlık vektörlerinin zamanla değiştiği de unutulmamalıdır.

İleri yönde sonraki öngörü hatası $f_m(n)$, m . dereceden ileri yönde öngörü hata filtresinin n anındaki $\mathbf{u}_{m+1}(n)$ tap giriş vektörüne karşı çıkışta ürettiği yanıtı ve

$$f_m(n) = \mathbf{a}_m^H(n) \mathbf{u}_{m+1}(n) \quad (5.141)$$

eşitliği ile gösterilir. $\mathbf{u}_{m+1}(n)$ vektörü iki değişik biçimde parçalara ayrılabilmesine göre,

$$\mathbf{u}_{m+1}(n) = \begin{bmatrix} \mathbf{u}_m(n) \\ u(n-m) \end{bmatrix} \quad (5.142)$$

$$\mathbf{u}_{m+1}(n) = \begin{bmatrix} u(n) \\ \mathbf{u}_m(n-1) \end{bmatrix} \quad (5.143)$$

(5.139) eşitliği (5.141) eşitliğinde yerine yazılır ve (5.139) eşitliğinin sağ tarafındaki ilk terim için (5.142), ikinci terim için de (5.143) parçalama biçimlerinden yararlanılırsa, ileri yönde sonraki öngörü hatası için derece güncelleme yinelemesi

$$f_m(n) = f_{m-1}(n) + \Gamma_{f,m}^*(n) b_{m-1}(n-1) \quad , \quad m = 1, 2, \dots, M \quad (5.144)$$

şeklinde elde edilir.

Geri yönde sonraki öngörü hatası $b_m(n)$, m . dereceden geri yönde öngörü hata filtresinin n anındaki $\mathbf{u}_{m+1}(n)$ tap giriş vektörüne karşı çıkışta ürettiği yanıttır ve

$$b_m(n) = \mathbf{c}_m^H(n) \mathbf{u}_{m+1}(n) \quad (5.145)$$

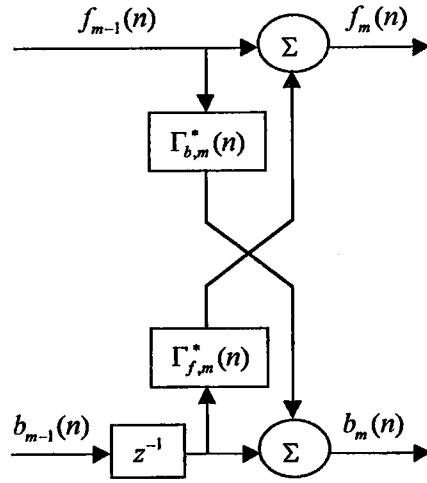
eşitliği ile gösterilir. (5.140) eşitliği (5.145) eşitliğinde yerine yazılır ve (5.140) eşitliğinin sağ tarafındaki ilk terim için (5.143), ikinci terim için de (5.142) parçalama biçimlerinden yararlanılırsa, geri yönde sonraki öngörü hatasına ait derece güncelleme yinelemesi

$$b_m(n) = b_{m-1}(n-1) + \Gamma_{b,m}^*(n) f_{m-1}(n) \quad , \quad m = 1, 2, \dots, M \quad (5.146)$$

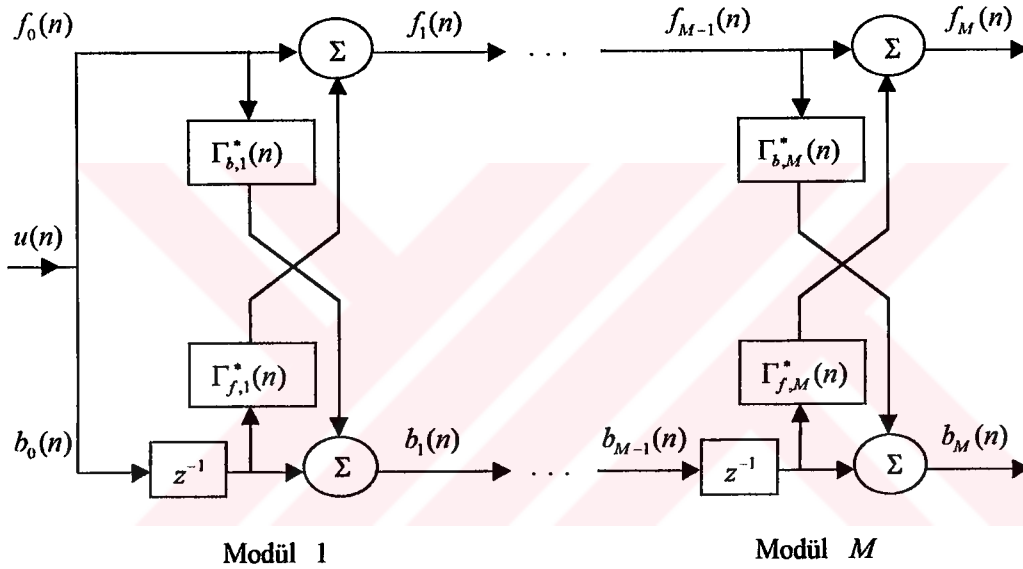
şeklinde elde edilir. (5.144) ve (5.146) eşitlikleri, kafes görünümlü işaret akış diyagramı ile gösterilir (Şekil 5.5(a)). Öngörü derecesi m , değişken bir parametre olup $1, 2, \dots, M$ değerlerini alır. Giriş verisi üzerinde hiçbir öngörü işleminin gerçekleştirilmediği ve

$$f_0(n) = b_0(n) = u(n) \quad (5.147)$$

değerlerinin elde edildiği $m = 0$ durumu, (5.144) ve (5.146) yinelemelerinin *başlangıç koşulu* olarak adlandırılır. Dolayısıyla, (5.147)'deki değerlere, bu iki yinelemenin *başlangıç değerleri* (*initial values*) denir. Öngörü derecesinin $m = 1$ 'den $m = M$ değerine doğru birer birer artırılması sonucu, Şekil 5.5(b) ile gösterilen *çok modüllü en küçük kareler kafes öngörücü* elde edilir.



(a)



(b)

Şekil 5.5 (a) En küçük kareler kafes modülü; (b) çok modüllü en küçük kareler kafes öngörücü (Haykin, 1991)

$f_m(n)$ ve $b_m(n)$ öngörü hatalarının en küçük kareler metoduyla elde edilen buradaki derece güncelleme yinelemelerinin, Bölüm 4.4'te elde edilen derece güncelleme yinelemelerine yapı olarak özdeş olması oldukça dikkat çekicidir. Çünkü Bölüm 4.4'teki yinelemeler, durağan girişlere ait ensemble ortalamalı $\mathbf{R}_m$ ilişki matrisinin Toeplitz özelliğinden kaynaklanmaktaydı. Oysa buradaki yinelemeler, zaman ortalamalı (deterministik) $\Phi_m(n)$ ilişki matrisinin genelde Toeplitz olmamasına rağmen (Bkz. Sf. 113) elde edilmiştir.

5.4.1.3 Sonraki öngörü hata karelerinin ağırlıklı toplamına ait derece güncelleme yinelemeleri

(5.130) ve (5.133) eşitlikleri sırasıyla, ileri yönde sonraki öngörü hata karelerinin ağırlıklı toplamı $F_m(n)$ ve geri yönde sonraki öngörü hata karelerinin ağırlıklı toplamı $B_m(n)$ için olan derece güncelleme yinelemelerini tanımlar. (5.134) ile verilen ilişkinin bu eşitliklerde kullanılması sonucunda,

$$F_m(n) = F_{m-1}(n) - \frac{|\Delta_{m-1}(n)|^2}{B_{m-1}(n-1)} \quad (5.148)$$

$$B_m(n) = B_{m-1}(n-1) - \frac{|\Delta_{m-1}(n)|^2}{F_{m-1}(n)} \quad (5.149)$$

derece güncelleme yinelemeleri elde edilir. (5.144), (5.146), (5.148) ve (5.149) eşitlikleri ve yansıma katsayıları için yapılan (5.137) ve (5.138) tanımları, en küçük kareler kafes öngörücüye ait temel derece güncelleme yinelemeleridir. Bu yinelemeler sayesinde, ileri yönde sonraki öngörü hata dizisi $f_0(n), f_1(n), \dots, f_M(n)$ ve geri yönde sonraki öngörü hata dizisi $b_0(n), b_1(n), \dots, b_M(n)$ üretilmektedir.

5.4.1.4 $\gamma_m(n-1)$ için derece güncelleme yinelemesi

RLSL algoritmasına ait derece güncelleme yinelemeleri, Bölüm 5.3.4'te anlatılan $\gamma_m(n)$ dönüşüm çarpanının birim geciktirilmiş $\gamma_m(n-1)$ için yazılacak yineleme ile son bulur. (5.105) eşitliğinden, $\mathbf{k}_m(n-1)$ kazanç vektörünün derece güncelleme yinelemesi

$$\mathbf{k}_m(n-1) = \begin{bmatrix} \mathbf{k}_{m-1}(n-1) \\ 0 \end{bmatrix} + \frac{b_{m-1}(n-1)}{B_{m-1}(n-1)} \mathbf{c}_{m-1}(n-1), \quad m = 1, 2, \dots, M \quad (5.150)$$

olarak yazılır. Bu yinelemenin Hermitian transpozunu alındıktan sonra, her iki yanı $\mathbf{u}_m(n-1)$ vektörü ile sağ taraftan çarpılır ise,

$$\mathbf{k}_m^H(n-1) \mathbf{u}_m(n-1) = [\mathbf{k}_{m-1}^H(n-1), 0] \mathbf{u}_m(n-1) + \frac{b_{m-1}^*(n-1)}{B_{m-1}(n-1)} \mathbf{c}_{m-1}^H(n-1) \mathbf{u}_m(n-1) \quad (5.151)$$

elde edilir. (5.108) eşitliğiyle verilen tanım gereği,

$$\mathbf{k}_m^H(n-1) \mathbf{u}_m(n-1) = 1 - \gamma_m(n-1) \quad (5.152)$$

$$\begin{aligned} [\mathbf{k}_{m-1}^H(n-1), 0] \mathbf{u}_m(n-1) &= [\mathbf{k}_{m-1}^H(n-1), 0] \begin{bmatrix} \mathbf{u}_{m-1}(n-1) \\ u(n-m) \end{bmatrix} \\ &= \mathbf{k}_{m-1}^H(n-1) \mathbf{u}_{m-1}(n-1) \\ &= 1 - \gamma_{m-1}(n-1) \end{aligned} \quad (5.153)$$

olmaktadır. Ayrıca, (5.145) eşitliğinden

$$\mathbf{c}_{m-1}^H(n-1) \mathbf{u}_m(n-1) = b_{m-1}(n-1) \quad (5.154)$$

yazılabileceğine göre, (5.151) eşitliği

$$\gamma_m(n-1) = \gamma_{m-1}(n-1) - \frac{|b_{m-1}(n-1)|^2}{B_{m-1}(n-1)}, \quad m = 1, 2, \dots, M \quad (5.155)$$

halini alır. Bu ise, RLSL algoritmasına ait derece güncelleme yinelemelerinin sonucusudur.

5.4.2 Zaman güncelleme yinelemesi

Bölüm 5.4.1’de elde edilen yinelemeler, $m = 0$ ilk koşulu ile başlayarak, ileri ve geri yönde sonraki öngörü hatalarını ve bu hataların karelerinin ağırlıklı toplamını derece olarak günceller. Bu yinelemeleri zaman olarak da güncelleyebilmek, diğer bir deyişle, uyarlamalı bir hale getirebilmek için, bu yinelemelerin hesaplanmasında kullanılan $\Delta_{m-1}(n)$ parametresine ait zaman güncelleme yinelemesinin elde edilmesi gerekir.

$(m-1)$. dereceden ileri yönde öngörü hata filtresinin $(n-1)$ anında hesaplanan $m \times 1$ boyutlu tap ağırlık vektörü $\mathbf{a}_{m-1}(n-1)$ ele alınsın. Bu vektörün ilk elemanı daima 1 olduğu için, (5.136) eşitliğinde $\mathbf{a}_{m-1}(n)$ yerine $\mathbf{a}_{m-1}(n-1)$ alınması sonucu değiştirmez. Öyleyse, (5.134) ve (5.136) eşitliklerine dayanılarak, $\Delta_{m-1}(n)$ için

$$\Delta_{m-1}(n) = [\Delta_{m-1}(n), \mathbf{0}_{m-1}^T, B_{m-1}(n-1)] \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \quad (5.156)$$

eşitliğini yazmak mümkündür. (5.127) eşitliğinin her iki yanının Hermitian transpozu alındığında, $\Phi_{m+1}(n)$ matrisinin Hermitian özelliğinden ve (5.134) ilişkisinden dolayı,

$$[0, \mathbf{c}_{m-1}^H(n-1)] \Phi_{m+1}(n) = [\Delta_{m-1}(n), \mathbf{0}_{m-1}^T, B_{m-1}(n-1)] \quad (5.157)$$

olacaktır. (5.157) eşitliği (5.156) eşitliğinde yerine yazılarak,

$$\Delta_{m-1}(n) = [0, \mathbf{c}_{m-1}^H(n-1)] \Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \quad (5.158)$$

elde edilir. $\Phi_{m+1}(n)$ ilişki matrisinin zaman güncelleme yinelemesi (5.16) eşitliğinden

$$\Phi_{m+1}(n) = \lambda \Phi_{m+1}(n-1) + \mathbf{u}_{m+1}(n) \mathbf{u}_{m+1}^H(n) \quad (5.159)$$

şeklinde yazılabileceğine göre, (5.158) eşitliği

$$\begin{aligned} \Delta_{m-1}(n) &= \lambda [0, \mathbf{c}_{m-1}^H(n-1)] \Phi_{m+1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \\ &\quad + [0, \mathbf{c}_{m-1}^H(n-1)] \mathbf{u}_{m+1}(n) \mathbf{u}_{m+1}^H(n) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \end{aligned} \quad (5.160)$$

haline gelir. Sırasıyla (5.76) ve (5.145) eşitliklerinden yararlanılarak,

$$\begin{aligned} \mathbf{u}_{m+1}^H(n) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} &= [\mathbf{u}_m^H(n), u^*(n-m)] \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \\ &= \mathbf{u}_m^H(n) \mathbf{a}_{m-1}(n-1) \\ &= \eta_{m-1}^*(n) \end{aligned} \quad (5.161)$$

$$\begin{aligned} [0, \mathbf{c}_{m-1}^H(n-1)] \mathbf{u}_{m+1}(n) &= [0, \mathbf{c}_{m-1}^H(n-1)] \begin{bmatrix} u(n) \\ \mathbf{u}_m(n-1) \end{bmatrix} \\ &= \mathbf{c}_{m-1}^H(n-1) \mathbf{u}_m(n-1) \\ &= b_{m-1}(n-1) \end{aligned} \quad (5.162)$$

eşitliklerine ulaşılır. Diğer yandan, (5.121) eşitliğinde n yerine $(n-1)$ yazıldığında,

$$\Phi_{m+1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} = \begin{bmatrix} F_{m-1}(n-1) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n-1) \end{bmatrix} \quad (5.163)$$

olmaktadır. $\mathbf{c}_{m-1}(n-1)$ vektörünün son elemanı her zaman 1 olduğuna göre,

$$\begin{aligned} [0, \mathbf{c}_{m-1}^H(n-1)] \Phi_{m+1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} &= [0, \mathbf{c}_{m-1}^H(n-1)] \begin{bmatrix} F_{m-1}(n-1) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n-1) \end{bmatrix} \\ &= \Delta_{m-1}(n-1) \end{aligned} \quad (5.164)$$

olacaktır. (5.161), (5.162) ve (5.164) eşitliklerinin (5.160) eşitliğinde kullanılmasıyla,

$$\Delta_{m-1}(n) = \lambda \Delta_{m-1}(n-1) + b_{m-1}(n-1) \eta_{m-1}^*(n) \quad (5.165)$$

şeklinde bir zaman güncelleme yinelemesi elde edilir. İleri yönde önceki öngörü hatası yerine (5.112) eşitliğinden faydalanılarak yazılan

$$\eta_{m-1}(n) = \frac{f_{m-1}(n)}{\gamma_{m-1}(n-1)} \quad (5.166)$$

eşitliği kullanılırsa, $\Delta_{m-1}(n)$ parametresine ait zaman güncelleme yinelemesi

$$\Delta_{m-1}(n) = \lambda \Delta_{m-1}(n-1) + \frac{b_{m-1}(n-1) f_{m-1}^*(n)}{\gamma_{m-1}(n-1)} \quad (5.167)$$

haline gelir.

(5.167) yinelemesindeki düzeltme terimi, $1/\gamma_{m-1}(n-1)$ çarpanı ile kuvvetlendirilmektedir. Bu parametre sayesinde RLSL algoritması, giriş verisindeki ani değişimlere çok hızlı bir şekilde adapte olabilmektedir (Morf ve Lee, 1978). $\Delta_{m-1}(n)$ 'e ait zaman güncelleme yinelemesinden $1/\gamma_{m-1}(n-1)$ kuvvetlendirme çarpanının kaldırılması sonucunda geriye kalan ifade, $b_{m-1}(n-1)$ ile $f_{m-1}(n)$ arasındaki zaman ortalamalı çapraz ilişkiye eşit olur.

(5.110) ile de belirtildiği gibi, dönüşüm çarpanı $\gamma_{m-1}(n-1)$ sıfır ile 1 arasında gerçel değerler alan bir skalerdir. Dönüşüm çarpanı sıfıra eşit olduğunda, (5.167) yinelemesi durur. Çünkü bu durumda, $\Phi_{m+1}(n)$ ilişki matrisinin tersi alınamaz. Öyleyse, $\gamma_{m-1}(n-1)$ dönüşüm çarpanının zamanla değişimi takip edilerek, filtrenin durumu hakkında bilgi edinilebilir.

Sonraki kestirim hatalarına dayalı RLSL algoritmasını oluşturan derece ve zaman güncelleme yinelemeleri şu sırada listelenir: (5.167), (5.137), (5.138), (5.144), (5.146), (5.148), (5.149) ve (5.155). Bu yinelemelerin bölme işlemi de içeriyor olması, bölme işlemindeki bölenin çok küçük değerler alıp almadığına dikkat edilmesini gerektirir. Çünkü, yüksek doğrulukta işlem yapan bir bilgisayar kullanılmaması durumunda, $F_0(0)$ ve $B_0(0)$ başlangıç değerlerini belirleyen δ sabitinin seçimi, RLSL algoritmasının başlangıç anındaki geçici hal başarısını kötü yönde etkileyebilir. Bu durum karşısında, Friedlander (1982) tarafından bir eşik seviyesinin kullanılması önerilmiştir. Bu öneriye göre, RLSL algoritmasındaki herhangi bir bölme işleminin bölen değeri önceden belirlenen eşik seviyesinden küçük oluyor ise, o bölme işlemi sıfır kabul edilmektedir.

5.4.3 Yinelemeli en küçük kareler kafes algoritmasının başlangıç koşulları

Sonraki kestirim hatalarına dayalı RLSL algoritmasına başlangıç koşullarının verilmesine, (5.144) ve (5.146) yinelemelerinde olduğu gibi, $m = 0$ ile başlanır. Bunun dışında, sonraki öngörü hata karelerinin ağırlıklı toplamına ait başlangıç değerlerine de ihtiyaç duyulur. (5.79) eşitliğiyle verildiği üzere, ileri yönde sonraki öngörü hata karelerinin ağırlıklı toplamına ait zaman güncelleme yinelemesi

$$F_m(n) = \lambda F_m(n-1) + f_m(n) \eta_m^*(n) \quad (5.168)$$

olmaktaydı. $m = 0$ için $f_0(n) = \eta_0(n) = u(n)$ olduğuna göre, (5.168) eşitliği $m = 0$ için

$$F_0(n) = \lambda F_0(n-1) + |u(n)|^2 \quad (5.169)$$

olacaktır. $u(n)$, kafes öngörücünün n anındaki girişidir. Benzer şekilde, (5.97) eşitliğiyle verilen geri yönde sonraki öngörü hata karelerinin ağırlıklı toplamına ait zaman güncelleme yinelemesi $m = 0$ için

$$B_0(n) = \lambda B_0(n-1) + |u(n)|^2 \quad (5.170)$$

olacaktır. $\gamma_m(n-1)$ dönüşüm çarpanının sıfır ile 1 arasında sınırlı olması nedeniyle, başlangıç değerinin

$$\gamma_0(n-1) = 1, \quad n \geq 1 \quad (5.171)$$

şeklinde seçilmesi mantıklı olur. RLSL algoritmasına başlangıç koşullarının verilmesi, $n = 0$ anı için

$$\Delta_{m-1}(0) = 0 \quad (5.172)$$

$$I'_0(0) = B_0(0) = \delta \quad (5.173)$$

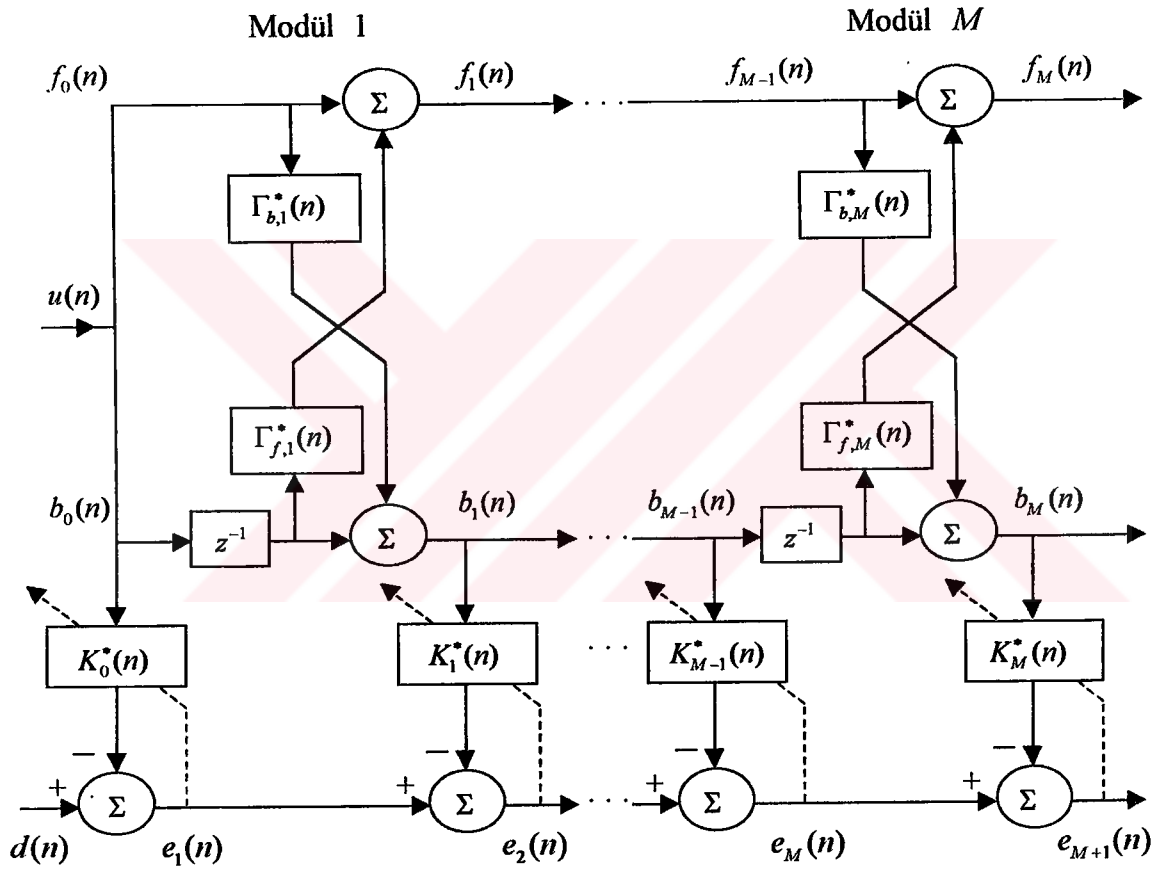
alınarak tamamlanır. Bölüm 5.2.3.2'de anlatıldığı gibi, $\Phi_{m+1}(n)$ ilişki matrisinin tekil olmayan bir matris olmasını sağlamak için kullanılan δ sabiti, küçük bir pozitif sayıdır.

5.4.4 Birleşik süreç kestirimi

Temeli Bölüm 4.5'teki Gram-Schmidt dikleştirme yöntemine dayanan birleşik süreç kestirimi, Bölüm 4.6'da da ele alınmıştı. Ancak, Bölüm 4.5 ve 4.6'da elde edilen eşitlikler, ensemble ortalamalı oldukları ve uyarlamalı filtreleme özelliğine sahip olmadıkları için, bu bölümde geçerli olamazlar. Çünkü buradaki birleşik süreç kestirimi, RLSL algoritması kullanılarak gerçekleştirileceği için, zaman ortalamalı (deterministik) bir yapıya ve uyarlamalı filtreleme özelliğine sahip olmaktadır. Dolayısıyla, bu bölümde elde edilecek yinelemeli eşitlikler, (birleşik süreç kestirimini gerçekleştirebilmesi için) RLSL algoritmasının sonuna yapılması gereken ilavelerdir. RLSL algoritmasına birleşik süreç kestirimi adı altında ilave edilecek bu yinelemeli eşitlikler, RLSL algoritmasına herhangi bir istenen yanıtın en küçük kareler kestirimini üretebilme özelliği kazandırır. Bu ilave yapılmazsa, RLSL algoritması sadece ileri ve geri yönde öngörü hata dizisi üretir.

RLSL algoritması, m modüllü bir kafes öngörücü için zamanın her anında birbirleriyle *zaman ortalamalı anlamda ilişkisiz (uncorrelated in a time-averaged sense)* olan $b_0(n), b_1(n), \dots, b_m(n)$ geri yönde öngörü hata dizisini üretir. Birbirleriyle zaman ortalamalı

anlamda ilişkisiz oldukları için, bu hatalara ait zaman ortalamalı (deterministik) ilişki matrisi diyagonal bir matris olur. Bu özelliğe, RLSL algoritmasının *birbirinden ayrılma özelliği (decoupling property)* denir. Buna göre, $b_0(n), b_1(n), \dots, b_m(n)$ geri yönde öngörü hataları kendilerine karşı düşen $K_0(n), K_1(n), \dots, K_m(n)$ gerileme katsayılarına tap girişleri olarak uygulandığında (Şekil 5.6), $d(n)$ istenen yanıtının en küçük kareler kestirimi oldukça doğru ve etkili bir şekilde elde edilir. Bu yapı; $\{u(n)\}$ sürecinin birbirleriyle ilişkili gözlem değerlerinden $\{d(n)\}$ sürecinin kestirimini, bu iki süreci $\{d(n), u(n)\}$ şeklinde birleşik süreç haline getirerek üretir ve bundan dolayı da, *birleşik süreç kestirici* olarak adlandırılır.



Şekil 5.6 Sonraki kestirim hatalarına dayalı RLSL algoritmasını kullanan birleşik süreç kestirici (Haykin, 1991)

Şekil 5.6'daki gerileme katsayılarını hesaplayan algoritmayı elde etmeden önce, RLSL algoritmasının birbirinden ayrılma özelliği, yani $b_0(n), b_1(n), \dots, b_m(n)$ öngörü hatalarının birbirleriyle zaman ortalamalı anlamda ilişkisiz oldukları ispatlanacaktır. Bunun için, m . dereceden geri yönde öngörü hata filtresi ele alınsın ve bu filtreye ait

$$\mathbf{c}_m^T(n) = [c_{m,m}(n), c_{m,m-1}(n), \dots, 1] \quad (5.174)$$

şeklindeki $(m+1) \times 1$ boyutlu $\mathbf{c}_m(n)$ tap ağırlık vektörünün, $1 \leq i \leq n$ gözlem aralığı boyunca en küçük kareler anlamında optimize edildiği kabul edilsin. Bu durumda, $(m+1) \times 1$ boyutlu $\mathbf{u}_{m+1}(i)$ tap giriş vektörü için bu filtrenin çıkışta üreteceği geri yönde sonraki öngörü hatası $b_m(i)$

$$\mathbf{u}_{m+1}^T(i) = [u(i), u(i-1), \dots, u(i-m)] , \quad i > m \quad (5.175)$$

$$\begin{aligned} b_m(i) &= \mathbf{c}_m^H(n) \mathbf{u}_{m+1}(i) \\ &= \sum_{k=0}^m c_{m,k}^*(n) u(i-m+k) , \quad \begin{matrix} m < i \leq n \\ m = 0, 1, 2, \dots \end{matrix} \end{aligned} \quad (5.176)$$

olacaktır. $\mathbf{b}_{m+1}(i)$,

$$\mathbf{b}_{m+1}^T(i) = [b_0(i), b_1(i), \dots, b_m(i)] , \quad \begin{matrix} i > m \\ m = 0, 1, 2, \dots \end{matrix} \quad (5.177)$$

şeklinde tanımlanan $(m+1) \times 1$ boyutlu geri yönde sonraki öngörü hata vektörünü gösterebilir. (5.176) eşitliği (5.177) eşitliğinde yerine yazıldığında, $\mathbf{u}_{m+1}(i)$ giriş vektörünü kendisine karşı düşen geri yönde sonraki öngörü hatalarına dönüştüren

$$\mathbf{b}_{m+1}(i) = \mathbf{L}_m(n) \mathbf{u}_{m+1}(i) \quad (5.178)$$

dönüşüm ifadesi elde edilir. $\mathbf{L}_m(n)$ ise

$$\mathbf{L}_m(n) = \begin{bmatrix} 1 & 0 & \dots & 0 \\ c_{1,1}^*(n) & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,m}^*(n) & c_{m,m-1}^*(n) & \dots & 1 \end{bmatrix} \quad (5.179)$$

şeklinde tanımlı $(m+1) \times (m+1)$ boyutlu dönüşüm matrisidir. Bu matrisin özellikleri:

1. l . satırında bulunan sıfırdan farklı elemanları, $(l-1)$. dereceden geri yönde öngörü hata filtresine ait tap ağırlıklarından oluşur. Bu durumda $\mathbf{L}_m(n)$ simgesindeki m alt indisi, bu

matrisi oluşturan geri yönde öngörü hata filtreleri içinden en büyük dereceli olanın derecesini göstermektedir.

2. Ana diyagonalindeki tüm elemanları 1 olmaktadır. Bu ise, geri yönde öngörü hata filtrelerinin sonuncu tap ağırlıklarının daima 1 olmasından (5.174) kaynaklanır.

3. Alt üçgensel matristir. Ayrıca, ana diyagonalindeki tüm elemanları 1'e eşit olduğundan, her m değeri için determinantı 1 olur. Bu ise, her m değeri için tersinin mevcut olduğu anlamına gelir.

(5.13) eşitliğinden, $\mathbf{b}_{m+1}(i)$ geri yönde sonraki öngörü hata vektörüne ait $(m+1) \times (m+1)$ boyutlu zaman ortalamalı ilişki matrisi

$$\mathbf{D}_{m+1}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{b}_{m+1}(i) \mathbf{b}_{m+1}^H(i), \quad \begin{matrix} m < i \leq n \\ m = 0, 1, 2, \dots \end{matrix} \quad (5.180)$$

olarak yazılır. (5.178) eşitliği (5.180) eşitliğinde yerine yazıldığında,

$$\begin{aligned} \mathbf{D}_{m+1}(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{L}_m(n) \mathbf{u}_{m+1}(i) \mathbf{u}_{m+1}^H(i) \mathbf{L}_m^H(n) \\ &= \mathbf{L}_m(n) \left[\sum_{i=1}^n \lambda^{n-i} \mathbf{u}_{m+1}(i) \mathbf{u}_{m+1}^H(i) \right] \mathbf{L}_m^H(n) \end{aligned} \quad (5.181)$$

elde edilir. Köşeli parantez içindeki ifade $\mathbf{u}_{m+1}(i)$ giriş vektörünün zaman ortalamalı ilişki matrisi $\Phi_{m+1}(n)$ 'e eşit olduğuna göre, (5.181) eşitliği

$$\mathbf{D}_{m+1}(n) = \mathbf{L}_m(n) \Phi_{m+1}(n) \mathbf{L}_m^H(n) \quad (5.182)$$

haline gelir. Geri yönde doğrusal öngörüye ait genişletilmiş normal denklemleri (5.86) kullanıldığında, $\Phi_{m+1}(n) \mathbf{L}_m^H(n)$ çarpımının (ana diyagonalı $B_0(n), B_1(n), \dots, B_m(n)$ değerlerinden oluşan) bir alt üçgensel matris olduğu görülür. $\mathbf{L}_m(n)$ matrisi de bir alt üçgensel matris olduğuna göre, iki alt üçgensel matrisin çarpımı olan $\mathbf{L}_m(n) \Phi_{m+1}(n) \mathbf{L}_m^H(n)$ çarpımı da bir alt üçgensel matris olacaktır. Diğer yandan, $\mathbf{L}_m^H(n)$ bir üst üçgensel matristir ve benzer şekilde, $\Phi_{m+1}(n) \mathbf{L}_m^H(n)$ çarpımının Hermitian transpozu olan $\mathbf{L}_m(n) \Phi_{m+1}(n)$ çarpımı da bir üst üçgensel matristir. Dolayısıyla, iki üst üçgensel matrisin çarpımı olan

$\mathbf{L}_m(n) \Phi_{m+1}(n) \mathbf{L}_m^H(n)$ çarpımı da bir üst üçgensel matris olacaktır. Bu durumda $\mathbf{D}_{m+1}(n)$ matrisinin, hem alt üçgensel ve hem de üst üçgensel bir matris olduğu ortaya çıkar. Bu da demektir ki, $\mathbf{D}_{m+1}(n)$ matrisi

$$\begin{aligned} \mathbf{D}_{m+1}(n) &= \mathbf{L}_m(n) \Phi_{m+1}(n) \mathbf{L}_m^H(n) \\ &= \text{diag} [B_0(n), B_1(n), \dots, B_m(n)] \end{aligned} \quad (5.183)$$

şeklinde tanımlı bir diyagonal matristir. (5.180) ve (5.183) eşitlikleri birlikte incelendiğinde, $b_0(n), b_1(n), \dots, b_m(n)$ geri yönde sonraki öngörü hatalarının zamanın her anında birbirleriyle zaman ortalamalı anlamda ilişkisiz oldukları ispatlanmış olur.

Öyleyse, RLS algoritmasının birbirleriyle ilişkili $u(i), u(i-1), \dots, u(i-m)$ giriş dizisini birbirleriyle ilişkisiz $b_0(i), b_1(i), \dots, b_m(i)$ geri yönde sonraki öngörü hata dizisine dönüştürmesini, Gram-Schmidt dikleştirme yönteminin zaman ortalamalı (deterministik) hali olarak düşünmek yanlış olmaz.

5.4.4.1 Yinelemeli en küçük kareler metoduna ait $\hat{\mathbf{w}}_m(n)$ tap ağırlık vektörü ile $\mathbf{K}_m(n)$ gerileme katsayı vektörü arasındaki ilişki

$u(n), u(n-1), \dots, u(n-m)$ tap girişlerine ve bu girişlere karşı düşen $\hat{w}_0(n), \hat{w}_1(n), \dots, \hat{w}_m(n)$ tap ağırlıklarına sahip bir enine filtre ele alınsın. Filtrenin yukarıda elemanları verili $(m+1) \times 1$ boyutlu $\hat{\mathbf{w}}_m(n)$ tap ağırlık vektörünün, yinelemeli en küçük kareler (RLS) metoduyla elde edildiği kabul edilsin. Bu durumda $\hat{\mathbf{w}}_m(n)$, RLS metoduna ait normal denklemleri (5.12) ile

$$\Phi_{m+1}(n) \hat{\mathbf{w}}_m(n) = \boldsymbol{\theta}_{m+1}(n) \quad (5.184)$$

şeklinde tanımlanır. $\Phi_{m+1}(n)$, filtrenin tap girişlerine ait $(m+1) \times (m+1)$ boyutlu ilişki matrisi ve $\boldsymbol{\theta}_{m+1}(n)$ ise, istenen yanıt $d(i)$ ile tap girişleri arasındaki $(m+1) \times 1$ boyutlu çapraz ilişki vektörüdür. (5.184) eşitliğine aşağıdaki değişikliklerin her ikisi birlikte uygulanırsa:

1. Eşitliğin her iki yanını $(m+1) \times (m+1)$ boyutlu $\mathbf{L}_m(n)$ matrisiyle sol taraftan çarpmak,
2. $\Phi_{m+1}(n)$ matrisi ile $\hat{\mathbf{w}}_m(n)$ vektörü arasına $(m+1) \times (m+1)$ boyutlu $\mathbf{I} = \mathbf{L}_m^H(n) \mathbf{L}_m(n)$ birim matrisini yazmak,

$$\mathbf{L}_m(n) \Phi_{m+1}(n) \mathbf{L}_m^H(n) \mathbf{L}_m^{-H}(n) \hat{\mathbf{w}}_m(n) = \mathbf{L}_m(n) \boldsymbol{\theta}_{m+1}(n) \quad (5.185)$$

şeklinde bir ifade elde edilir.

$b_0(n), b_1(n), \dots, b_m(n)$ geri yönde sonraki öngörü hataları, Şekil 5.6'daki enine filtrenin tap girişleridir. Enine filtrenin bu tap girişleri ile istenen yanıt $d(i)$ arasındaki $(m+1) \times 1$ boyutlu çapraz ilişki vektörü, $\mathbf{t}_{m+1}(n)$ ile gösterilsin. (5.14) ile verilen tanım gereği,

$$\mathbf{t}_{m+1}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{b}_{m+1}(i) d^*(i) \quad (5.186)$$

olacaktır. (5.178) eşitliği (5.186) eşitliğinde yerine yazıldığında,

$$\begin{aligned} \mathbf{t}_{m+1}(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{L}_m(n) \mathbf{u}_{m+1}(i) d^*(i) \\ &= \mathbf{L}_m(n) \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_{m+1}(i) d^*(i) \\ &= \mathbf{L}_m(n) \boldsymbol{\theta}_{m+1}(n) \end{aligned} \quad (5.187)$$

sonucuna ulaşılır. (5.182) ve (5.187) eşitlikleri yardımıyla, (5.185) eşitliği

$$\mathbf{D}_{m+1}(n) \mathbf{L}_m^{-H}(n) \hat{\mathbf{w}}_m(n) = \mathbf{t}_{m+1}(n) \quad (5.188)$$

haline gelir. Böylelikle, tap ağırlık vektörü (5.184) eşitliğiyle tanımlanan tek başına bir enine filtre ile yine RLS metoduna dayalı ama bu defa bir kafes filtreye bağlı $K_0, K_1, \dots, K_m$ tap ağırlıklı Şekil 5.6'daki enine filtre arasında, (5.184)'e uygulanan iki değişikliğin sayesinde bir ilişki (5.188) kurulmuş oldu. Bu ilişkiyi tam anlamıyla açıklığa kavuşturmak üzere, Şekil 5.6'daki enine filtre ele alınsın. Bu filtrenin tap ağırlık vektörü, açık hali

$$\mathbf{K}_m^T(n) = [K_0(n), K_1(n), \dots, K_m(n)] \quad (5.189)$$

şeklinde olan $(m+1) \times 1$ boyutlu $\mathbf{K}_m(n)$ gerileme katsayı vektörüdür. Bu filtre de RLS metoduyla tasarlandığına göre; $\mathbf{K}_m(n)$ tap ağırlık vektörü, $\xi(n)$ başarı göstergesi (5.11) minimum yapılarak elde edilir ve $1 \leq i \leq n$ gözlem aralığı boyunca sabit kalır.

$$\xi_{\min}(n) = \sum_{i=1}^n \lambda^{n-i} |d(i) - \mathbf{K}_m^H(n) \mathbf{b}_{m+1}(i)|^2$$

Tap girişlerine ait ilişki matrisi $\mathbf{D}_{m+1}(n)$ ve tap girişleri ile istenen yanıt arasındaki çapraz ilişki vektörü $\mathbf{t}_{m+1}(n)$ olan böyle bir enine filtre için geriye kalan, tap ağırlık vektörünü RLS metoduna ait normal denklemleri (5.12) ile

$$\mathbf{D}_{m+1}(n) \mathbf{K}_m(n) = \mathbf{t}_{m+1}(n) \quad (5.190)$$

şeklinde tanımlamaktır. (5.188) ile (5.190) karşılaştırıldığında, birbirinin eşdeğeri olan

$$\mathbf{K}_m(n) = \mathbf{L}_m^{-H}(n) \hat{\mathbf{w}}_m(n) \quad (5.191)$$

$$\hat{\mathbf{w}}_m(n) = \mathbf{L}_m^H(n) \mathbf{K}_m(n) \quad (5.192)$$

bu iki ilişki elde edilir. Görüldüğü üzere $\mathbf{L}_m(n)$ matrisi, $\mathbf{b}_{m+1}(i)$ ve $\mathbf{u}_{m+1}(i)$ vektörlerini birbirine dönüştürdüğü gibi (5.178), $\mathbf{K}_m(n)$ ve $\hat{\mathbf{w}}_m(n)$ vektörlerini de birbirine dönüştürebilmektedir.

5.4.4.2 Gerileme katsayılarını hesaplayan algoritma

(5.187) ve (5.190) eşitliklerinden, $\mathbf{K}_m(n)$ gerileme katsayı vektörü için

$$\mathbf{K}_m(n) = \mathbf{D}_{m+1}^{-1}(n) \mathbf{L}_m(n) \boldsymbol{\theta}_{m+1}(n) \quad (5.193)$$

ifadesi elde edilir. $\mathbf{D}_{m+1}(n)$ diyagonal bir matris olduğuna göre,

$$\mathbf{D}_{m+1}^{-1}(n) = \text{diag} [B_0^{-1}(n), B_1^{-1}(n), \dots, B_m^{-1}(n)] \quad (5.194)$$

olacaktır. (5.174) ile verilen tanımdan ve $m \times 1$ boyutlu $\mathbf{0}_m$ sıfır vektöründen yararlanılarak, (5.179) ile verilen tanım

$$\mathbf{L}_m(n) = \begin{bmatrix} 1, \mathbf{0}_m^T \\ \mathbf{c}_1^H(n), \mathbf{0}_{m-1}^T \\ \vdots \\ \mathbf{c}_{m-1}^H(n), 0 \\ \mathbf{c}_m^H(n) \end{bmatrix} \quad (5.195)$$

şeklinde de yazılabilir. (5.194) ve (5.195) eşitliklerinden, (5.193) eşitliğindeki $(m+1)$ adet skaler için

$$K_m(n) = B_m^{-1}(n) \mathbf{c}_m^H(n) \boldsymbol{\theta}_{m+1}(n), \quad m = 0, 1, \dots, M \quad (5.196)$$

genel ifadesi elde edilir. M , Şekil 5.6'daki kafes öngörücünün son modülünü, diğer bir deyişle, öngörü derecesini gösterir. $K_m(n)$ gerileme katsayısını zaman olarak güncelleyecek yinelemeyi elde etmek için,

$$\rho_m(n) = \mathbf{c}_m^H(n) \boldsymbol{\theta}_{m+1}(n), \quad m = 0, 1, \dots, M \quad (5.197)$$

şeklinde bir skaler tanımlansın. Bu durumda (5.196) genel ifadesi,

$$K_m(n) = \frac{\rho_m(n)}{B_m(n)}, \quad m = 0, 1, \dots, M \quad (5.198)$$

haline gelir. (5.113) dönüşümü kullanılarak, m . dereceden geri yönde öngörü hata filtresine ait tap ağırlık vektörünün zaman güncelleme yinelemesi (5.96)

$$\mathbf{c}_m(n) = \mathbf{c}_m(n-1) - \frac{b_m^*(n)}{\gamma_m(n)} \begin{bmatrix} \mathbf{k}_m(n) \\ 0 \end{bmatrix} \quad (5.199)$$

şeklinde yazılabilir. Hatırlanacağı gibi, $(m+1) \times 1$ boyutlu $\boldsymbol{\theta}_{m+1}(n)$ çapraz ilişki vektörünün zaman güncelleme yinelemesi (5.17)

$$\boldsymbol{\theta}_{m+1}(n) = \lambda \boldsymbol{\theta}_{m+1}(n-1) + \mathbf{u}_{m+1}(n) d^*(n) \quad (5.200)$$

olmaktaydı. Buna göre, (5.199) ve (5.200) zaman güncelleme yinelemeleri (5.197)'de yerlerine yazıldığında, $\rho_m(n)$ skaleri için aranılan

$$\rho_m(n) = \lambda \mathbf{c}_m^H(n-1) \boldsymbol{\theta}_{m+1}(n-1) + \mathbf{c}_m^H(n-1) \mathbf{u}_{m+1}(n) d^*(n) - \frac{b_m(n)}{\gamma_m(n)} \mathbf{k}_m^H(n) \boldsymbol{\theta}_m(n) \quad (5.201)$$

zaman güncelleme yinelemesi elde edilir. Bu yinelemeyi basitleştirmeye başlamadan önce, sağ taraftaki sonuncu terimin nasıl yazıldığını açıklamakta fayda vardır: $(m+1) \times 1$ boyutlu $\boldsymbol{\theta}_{m+1}(n)$ vektörünün son elemanı haricindeki ilk m adet elemanı, $m \times 1$ boyutlu $\boldsymbol{\theta}_m(n)$ vektörünün tüm elemanlarını oluşturur. Bu nedenle, (5.199) yinelemesinin sağındaki ikinci vektörün Hermitian transpozu $\boldsymbol{\theta}_{m+1}(n)$ vektörü ile sağ taraftan çarpıldığında, $\boldsymbol{\theta}_{m+1}(n)$

vektörünün son elemanı ortadan kalkar ve geriye, ilk m adet elemanının çarpımından doğan $\mathbf{k}_m^H(n) \boldsymbol{\theta}_m(n)$ çarpımı kalır. (5.201) yinelemesi basitleştirmek amacıyla:

1. (5.197) tanımından yararlanılarak,

$$\rho_m(n-1) = \mathbf{c}_m^H(n-1) \boldsymbol{\theta}_{m+1}(n-1)$$

2. Geri yönde önceki öngörü hatasına ait (5.94) eşitliğinde (5.113) dönüşümünden yararlanılarak,

$$\frac{b_m(n)}{\gamma_m(n)} = \mathbf{c}_m^H(n-1) \mathbf{u}_{m+1}(n)$$

3. Kazanç vektörüne ait (5.26) eşitliğinin Hermitian transpozunun $\boldsymbol{\theta}_m(n)$ vektörü ile sağ taraftan çarpılması sonucu elde edilen ifadede RLS metoduna ait normal denklemlerinden (5.184) yararlanılarak,

$$\begin{aligned} \mathbf{k}_m^H(n) \boldsymbol{\theta}_m(n) &= \mathbf{u}_m^H(n) \boldsymbol{\Phi}_m^{-1}(n) \boldsymbol{\theta}_m(n) \\ &= \mathbf{u}_m^H(n) \hat{\mathbf{w}}_{m-1}(n) \end{aligned}$$

eşitliği yazılır.

Yukarıdaki eşitliğin son satırı (5.31) eşitliği ile karşılaştırıldığında, $d(n)$ istenen yanıtının RLS metodu ile elde edilmiş olan kestiriminin kompleks eşleniğine eşit olduğu görülür. Bu kestirim $\hat{d}(n|U_n)$ ile gösterilirse, (5.201) yinelemesi

$$\rho_m(n) = \lambda \rho_{m-1}(n-1) + \frac{b_m(n)}{\gamma_m(n)} [d^*(n) - \hat{d}^*(n|U_n)] \quad (5.202)$$

haline gelir. Bölüm 5.4.4.1'in başında anlatılan, ama bu defa m adet tap girişine $u(n), u(n-1), \dots, u(n-m+1)$ ve m adet tap ağırlığına $\hat{w}_0(n), \hat{w}_1(n), \dots, \hat{w}_{m-1}(n)$ sahip bir enine filtreye ait sonraki kestirim hatası

$$\begin{aligned} e_m(n) &= d(n) - \hat{d}(n|U_n) \\ &= d(n) - \hat{\mathbf{w}}_{m-1}^H(n) \mathbf{u}_m(n) \end{aligned} \quad (5.203)$$

olarak tanımlansın. Böylece, $\rho_m(n)$ skalerine ait zaman güncelleme yinelemesi

$$\rho_m(n) = \lambda \rho_{m-1}(n-1) + \frac{b_m(n)}{\gamma_m(n)} e_m^*(n) , \quad m = 0, 1, \dots, M \quad (5.204)$$

şeklinde en son basitleştirilmiş halini alır.

(5.167) yinelemesindeki benzer biçimde, (5.204) yinelemesindeki düzeltme terimi, $1/\gamma_m(n)$ çarpanı ile kuvvetlendirilmektedir. (5.204) yinelemesinden $1/\gamma_m(n)$ kuvvetlendirme çarpanının kaldırılması sonucunda geriye kalan ifade, $b_m(n)$ ile $e_m(n)$ arasındaki zaman ortalamalı çapraz ilişkiye eşit olur.

Birleşik süreç kestirimini gerçekleştirecek olan algoritmayı tamamlayabilmek için, (5.198) ve (5.204) zaman güncelleme yinelemelerine ek olarak, bir başka yinelemeye daha ihtiyaç duyulur. Bu da, sonraki kestirim hatasına ait derece güncelleme yinelemesidir (bilinen bir $e_m(n)$ değerinden $e_{m+1}(n)$ değerinin hesaplanmasını sağlar). Bunun için önce, (5.203) eşitliğinde m yerine $(m+1)$ alınarak,

$$e_{m+1}(n) = d(n) - \hat{\mathbf{w}}_m^H(n) \mathbf{u}_{m+1}(n) \quad (5.205)$$

eşitliği yazılır. Ardından bu eşitlik, (5.178) ve (5.192) eşitlikleri yardımıyla,

$$\begin{aligned} e_{m+1}(n) &= d(n) - \mathbf{K}_m^H(n) \mathbf{b}_{m+1}(n) \\ &= d(n) - \sum_{l=0}^m K_l^*(n) b_l(n) \end{aligned} \quad (5.206)$$

haline getirilir. Son olarak, (5.206) eşitliğindeki son satır

$$e_{m+1}(n) = d(n) - \sum_{l=0}^{m-1} K_l^*(n) b_l(n) - K_m^*(n) b_m(n)$$

şeklinde yazıldığında, sonraki kestirim hatası için aranan derece güncelleme yinelemesi

$$e_{m+1}(n) = e_m(n) - K_m^*(n) b_m(n) , \quad m = 0, 1, \dots, M \quad (5.207)$$

olarak elde edilir.

Çizelge 5.2 Sonraki kestirim hatalarına dayalı RLSL algoritmasının özeti

Öngörü:

Çok modüllü en küçük kareler kafes öngörücünün son modülü M ile gösterilmek üzere, aşağıda sıralı olarak verilen güncellemeler, $n = 1$ ile başlanıp her bir $n = 1, 2, 3, \dots$ değeri için $m = 1, 2, \dots, M$ alınarak hesaplanır.

$$\Delta_{m-1}(n) = \lambda \Delta_{m-1}(n-1) + \frac{b_{m-1}(n-1) f_{m-1}^*(n)}{\gamma_{m-1}(n-1)}$$

$$\Gamma_{f,m}(n) = -\frac{\Delta_{m-1}(n)}{B_{m-1}(n-1)}$$

$$\Gamma_{b,m}(n) = -\frac{\Delta_{m-1}^*(n)}{F_{m-1}(n)}$$

$$f_m(n) = f_{m-1}(n) + \Gamma_{f,m}^*(n) b_{m-1}(n-1)$$

$$b_m(n) = b_{m-1}(n-1) + \Gamma_{b,m}^*(n) f_{m-1}(n)$$

$$F_m(n) = F_{m-1}(n) - \frac{|\Delta_{m-1}(n)|^2}{B_{m-1}(n-1)}$$

$$B_m(n) = B_{m-1}(n-1) - \frac{|\Delta_{m-1}(n)|^2}{F_{m-1}(n)}$$

$$\gamma_m(n-1) = \gamma_{m-1}(n-1) - \frac{|b_{m-1}(n-1)|^2}{B_{m-1}(n-1)}$$

Birleşik süreç kestirimi:

Yukarıdaki güncellemelerin hesaplanması bittikten sonra, aşağıda sıralı olarak verilen güncellemeler, yine $n = 1$ ile başlanıp her bir $n = 1, 2, 3, \dots$ değeri için bu defa $m = 0, 1, \dots, M$ alınarak hesaplanır.

$$\rho_m(n) = \lambda \rho_{m-1}(n-1) + \frac{b_m(n)}{\gamma_m(n)} e_m^*(n)$$

$$K_m(n) = \frac{\rho_m(n)}{B_m(n)}$$

$$e_{m+1}(n) = e_m(n) - K_m^*(n) b_m(n)$$

Başlangıç koşulları:

1. Algoritma, $n = 0$ anında aşağıdaki ilk değerler ile koşullandırılır:

$$\Delta_{m-1}(0) = 0$$

$$F_{m-1}(0) = \delta, \quad \delta = \text{Pozitif değerli küçük bir sabit sayı}$$

$$B_{m-1}(0) = \delta$$

Her bir $n \geq 1$ anında, aşağıdaki sıfır dereceli değişkenler üretilir:

$$f_0(n) = b_0(n) = u(n)$$

$$F_0(n) = B_0(n) = \lambda F_0(n-1) + |u(n)|^2$$

$$\gamma_0(n-1) = 1$$

2. Birleşik süreç kestirimi, $n = 0$ anında aşağıdaki ilk değer ile koşullandırılır:

$$\rho_m(0) = 0$$

Her bir $n \geq 1$ anında, aşağıdaki sıfır dereceli değişken üretilir:

$$e_0(n) = d(n)$$

Uyarı: $u(n)$ girişi ve $d(n)$ istenen yanıtı, $n \leq 0$ anları için sıfır kabul edilir.

6 . SAYISAL UYGULAMA

Bilinmeyen bir ortamda oldukça verimli çalışabilmek ve de giriş istatistiklerinin zamanla değişimini izleyebilmek, uyarlamalı filtreleri işaret-işleme ve kontrol uygulamaları için vazgeçilmez hale getiren özelliklerdir. Uyarlamalı filtreler ile haberleşme, kontrol, radar, sonar, deprem bilimi, görüntü işleme ve biomedikal mühendislik gibi farklı alanlarda başarılı uygulamalar gerçekleştirilmiştir. Nitelik olarak birbirlerinden farklı olmalarına karşın, bu uygulamaların tek ortak özelliği: bir giriş vektörüne ve bir istenen yanıtı sahip olmalarıdır. Bu iki değer sayesinde, filtrenin ayarlanabilir katsayılarının alacağı değerleri kontrol etmeye yarayan kestirim hatası hesaplanır. Ayarlanabilir katsayılar, kullanılan filtrenin yapısına göre, tap ağırlıkları (enine filtre) veya yansıma katsayıları (kafes filtre) olarak adlandırılır. Uyarlamalı filtre uygulamaları arasındaki en önemli fark ise, istenen yanıtın elde ediliş tarzıdır. Bu farklılığa göre, uyarlamalı filtre uygulamaları dört temel sınıfa ayrılabilir (Çizelge 6.1).

Çizelge 6.1 Uyarlamalı filtre uygulamaları

Uygulamanın Sınıfı	Uygulamanın Adı
1. ÖZDEŞLEME	Sistem Özdeşleme
2. TERSİNİ MODELLEME	Öngörüsül Dekonvolüsyon Uyarlamalı Dengeleme
3. ÖNGÖRÜ	Doğrusal Öngörüsül Kodlama Uyarlamalı Farksal Darbe Kod Modülasyonu Özbağımlı (AR) Spektrum Analizi İşaret Sezme
4. İNTERFERANS GİDERME	Uyarlamalı Gürültü Giderme Yankı Giderme Radar Polarimetri Uyarlamalı Dalga Biçimlendirme

Bu uygulamalarda, uyarlamalı filtrenin ne amaçla kullanıldığı ve istenen yanıtın nasıl elde edildiği aşağıda kısaca açıklanmaktadır:

1. Özdeşleme (Identification)

Uyarlamalı filtrenin bu uygulama sınıfındaki görevi: bilinmeyen bir sisteme en çok benzeyecek bir doğrusal model sağlamaktır. Bilinmeyen sistem ve uyarlamalı filtre aynı giriş ile sürülür. Bilinmeyen sistemin çıkışı, uyarlamalı filtre için gereken istenen yanıt olur.

2. Tersini Modelleme (Inverse Modeling)

Uyarlamalı filtrenin bu uygulama sınıfındaki görevi: bilinmeyen gürültülü bir sistemin tersine en çok benzeyecek olan bir ters model sağlamaktır. Elde edilen ters modelin transfer fonksiyonu bilinmeyen sistemin transfer fonksiyonunun tersine eşit oluyor ise, ideal durum sağlanmış demektir. Uyarlamalı filtre için gereken istenen yanıt, bilinmeyen sistemin girişi geciktirilerek sağlanır. Gecikme miktarı, bilinmeyen sisteme ve kullanılan uyarlamalı filtreye bağlı olarak değişir. Bazı uygulamalarda, bilinmeyen sistemin girişi hiç geciktirilmeden de istenen yanıt olarak kullanılmaktadır.

3. Öngörü (Prediction)

Uyarlamalı filtrenin buradaki görevi: bir rasgele işaretin şimdiki ana ait değerinin en iyi öngörüsünü elde etmektir. Öyleyse rasgele işaretin şimdiki değeri, uyarlamalı filtre için gereken istenen yanıt olur. İşaretin geçmiş değerleri ise, filtrenin girişi olarak kullanılır.

4. İnterferans Giderme (Interference Cancelling)

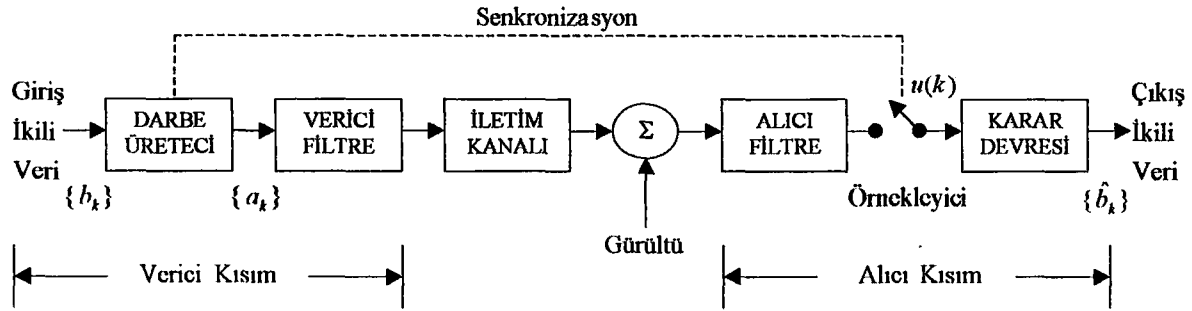
Uyarlamalı filtrenin buradaki görevi: bir ana işaretin bilgi taşıyan bileşeninde bulunan bilinmeyen interferansı gidermektir. Bu durumda ana işaret, uyarlamalı filtre için gereken istenen yanıt olur. Filtrenin girişi ise, bir referans işaret ile sürülür.

6.1 Uyarlamalı Dengeleme Kavramı

Mevcut olan kanal bant genişliğini en verimli şekilde kullanabilecek veri iletim sistemleri geliştirmek, üzerinde yoğun çalışmaların yapıldığı bir konudur. Amaç: kabul edilebilir bir hata oranı (error rate) veya ortalama sembol hata olasılığı (average probability of symbol error) ile, mümkün olan en yüksek hızda veri iletimi yapabilecek bir sistem tasarlamaktır. Doğrusal bir haberleşme kanalından sayısal veri iletimi yapmayı sınırlayan iki faktör vardır:

1. İntersemler interferansı (Intersymbol interference, ISI): Verici filtre, iletim kanalı ve alıcı filtre üçlüsünde meydana gelen dağılma (dispersion) nedeniyle oluşur.
2. Toplamsal ısı gürültü (Additive thermal noise).

Telefon kanalları gibi sınırlı bant genişliğine sahip kanallar için intersembol interferansı, yüksek hızlı veri iletim sistemlerinin tasarımında belirleyici baş faktördür. İkili darbe genlik modülasyonu (binary pulse-amplitude modulation) sisteminin eşdeğer temel bant modeli, Şekil 6.1’de gösterilmiştir. Temel bant terimi, orijinal işarete ait frekansların bulunduğu frekans bandına verilen isimdir.



Şekil 6.1 Bir temel bant veri iletim sistemine ait blok diyagram (dengelemesiz)
(Haykin,1991)

Sistemin verici kısmının girişine uygulanan işaret, 1 ve 0 değerlerinden oluşan $\{b_k\}$ ikili veri dizisidir. Bu dizi sırasıyla, darbe üretici, verici filtre ve iletim kanalı yolunu izleyerek alıcı filtreye varır. $u(k)$, alıcı filtrenin örneklenmiş çıkışını gösterebilir. Örnekleyici, vericideki darbe üretici ile senkron olarak çalıştırılır. $u(k)$ çıkışına, karar devresi vasıtasıyla bir eşik seviyesi uygulanır. Alıcı, bu eşik seviyesi aşıyor ise 1 sembolüne, aşılmıyor ise 0 sembolüne karar verir.

$$a_k = \begin{cases} +1, & b_k \text{ giriş biti 1 ise} \\ -1, & b_k \text{ giriş biti 0 ise} \end{cases} \quad (6.1)$$

şeklinde bir çarpan tanımlansın. Buna göre, gürültünün olmadığı durumlarda $u(k)$ için

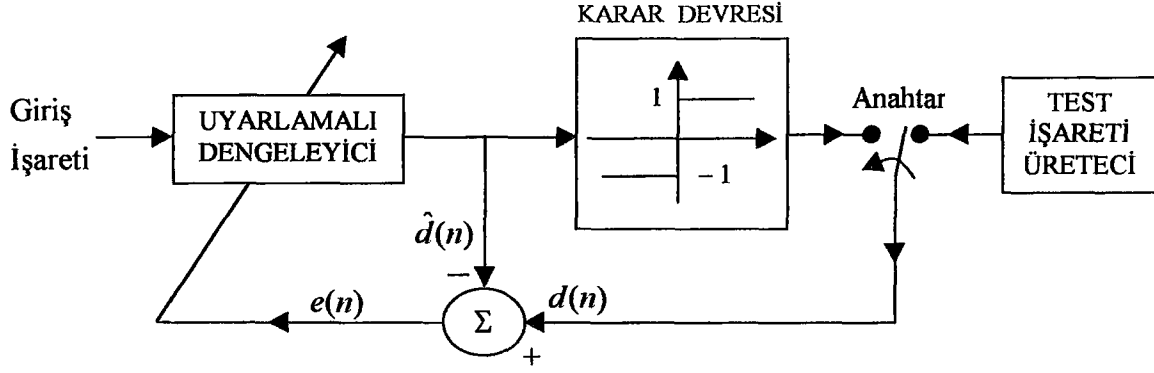
$$\begin{aligned} u(k) &= \sum_n a_n p(k-n) \\ &= a_k p(0) + \sum_{\substack{n \\ n \neq k}} a_n p(k-n) \end{aligned} \quad (6.2)$$

eşitliği yazılabilir. $p(n)$, birbirine kaskad bağlı verici filtre, iletim kanalı ve alıcı filtre üçlüsünün oluşturduğu sisteme ait $p(t)$ impuls yanıtının zaman domeninde örneklenmiş halidir. (6.2) eşitliğinin sağ tarafındaki ilk terim $a_k p(0)$, istenen yanıtı ve ikinci terim ise,

kaskad bağı verici filtre, iletim kanalı ve alıcı filtre üçlüsünün oluşturduğu sistemden kaynaklanan intersembol interferansını gösterir. Eğer bu intersembol interferansı kontrol altına alınmaz ise, karar devresi (girişine gelen örneklenmiş işaret hakkında) hatalı kararlar vermeye başlar. Öyleyse, intersembol interferansının üstesinden gelebilmek için, $p(n)$ fonksiyonunun kontrol altında tutulması gerekir. Teoride, iletim kanalının karakteristikleri tam olarak bilinir ise, intersembol interferansının örnekleme anlarındaki etkisini azaltacak bir verici ve alıcı filtre çiftinin tasarlanabileceği kabul edilir. Oysa pratikte, durum farklıdır; iletim kanalı, halen bilinmekte olan iletim kanalı modelleri içinden rasgele herhangi bir tanesine eşit veya benzer olmaktadır. Bu nedenle, mevcut kanal modelleri göz önüne alınarak elde edilen ortalama kanal karakteristiklerine göre tasarlanacak sabit bir verici ve alıcı filtre çifti, intersembol interferansını istenilen ölçüde azaltmayabilir. Bu durum, kanalın zaman yanıtı üzerinde tam bir kontrol sağlayan *uyarlamalı dengeleyici (adaptive equalizer)* kullanımını gerektirir (Lucky, 1965, 1966; Lucky vd., 1968; Proakis, 1975; Qureshi, 1985). Dengeleyici terimi telefon kanallarında, kanalın genlik ve gecikme karakteristiklerini düzenlemekle görevlendirilen bir filtreyi tarif etmek için kullanılır.

Veri iletim sistemlerinin dengelenmesi için öne sürülen temel görüşler arasında, iletim sisteminin ilk önce verici kısımda ve sonra da alıcı kısımda olmak üzere iki defa dengelenmesi gerektiği de yer almaktadır. Bu eski teknik, bir geri besleme yolu kullanmayı gerektirdiği için, burada sadece alıcı kısımda yapılan dengeleme işlemi ele alınacaktır. Uyarlamalı dengeleyici alıcı kısımda, örnekleyici ile karar devresi arasına yerleştirilir. Teoride, uyarlamalı dengeleyicideki ayarlanabilir katsayıların sayısını sınırsız derecede arttırarak, intersembol interferansı etkisinin azaltılabileceği kabul edilir.

Uyarlamalı filtreleme algoritmaları, $e(n)$ hata işareti olmadan çalışamazlar. Hata işaretini hesaplayabilmek için ise, $d(n)$ istenen yanıtının bilinmesi gerekir. Teoride, verici kısım çıkışındaki iletilen işaret, uyarlamalı dengeleyici için gereken istenen yanıttır. Oysa pratikte, uyarlamalı dengeleyici alıcı kısma yerleştirildiği için, dengeleyici ile istenen yanıt fiziksel olarak birbirlerinden ayrılmışlardır. Bu yüzden, alıcı kısımda istenen yanıtın tam bir kopyasının üretilmesi gerekmektedir. Şekil 6.2'deki blok diyagramda uyarlamalı dengeleyici, alıcı kısma yerleştirilmiştir ve istenen yanıtın kopyasını üretmek için şu iki metodu kullanmaktadır:



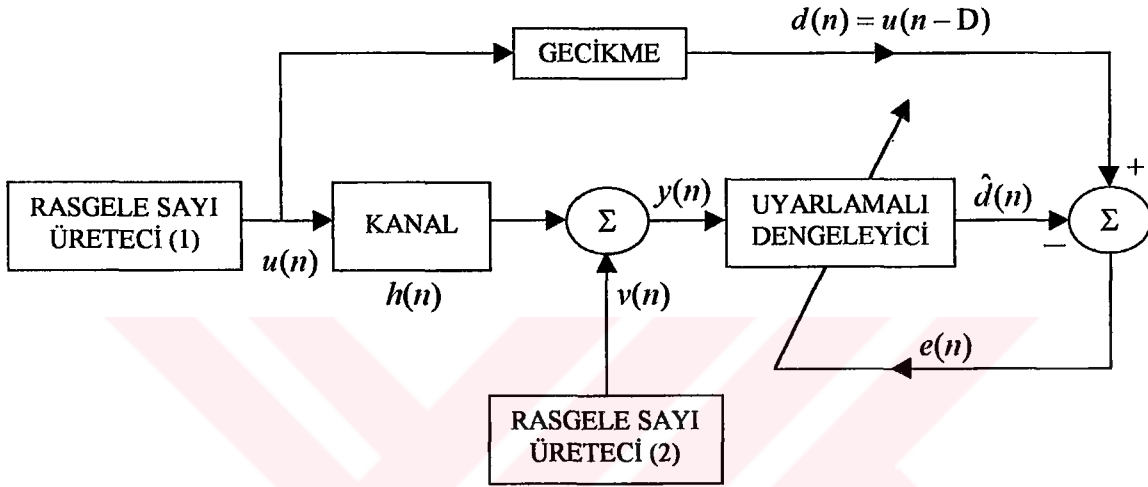
Şekil 6.2 Karar-yönetimli metot ile çalışan bir uyarlamalı dengeleyiciye ait blok diyagram (Hayes,1996)

1. *Eğitim metodu (Training method)*: Bu metot, alıcı ve vericinin birbirine bağlandıkları ilk anda gerçekleşmeye başlar. Bu metot sırasında, iletim kanalının nasıl olduğunu test etmek amacıyla, verici kısımdan alıcıya sözde rasgele olan fakat alıcı tarafından bilinen bir test işareti gönderilir. Test işaretinin alıcı tarafından biliniyor olmasının nedeni, bu işaretin alıcı kısımda depolanmış olmasından veya alıcı kısımda bulunan bir test işareti üretici tarafından üretiliyor olmasından dolayıdır. Gönderilen test işaretine dengeleyicinin verdiği yanıt, istenen yanıt görevini gören test işaretinden çıkarılarak kestirim hatası elde edilir. Elde edilen bu ilk kestirim hatasıyla, dengeleyicinin ayarlanabilir katsayılarına *başlangıç değerleri* verilmiş olur.

2. *Karar-yönetimli metot (Decision-directed method)*: Eğitim metoduyla dengeleyicinin ayarlanabilir katsayılarına başlangıç değerleri verildikten hemen sonra, alıcının verici kısımdan ne gönderildiğine dair hiçbir ön bilgisi olmaksızın, uyarlamalı dengeleyicinin katsayıları bu metot vasıtasıyla sürekli olarak ayarlanır. Bu metotta, dengeleyici çıkışı bir karar devresinin girişine uygulanmakta ve karar devresinin çıkışı da dengeleyici için gereken istenen yanıt oluşturmaktadır. İstenen yanıtın kopyası olarak alıcının kendi verdiği kararlar kullanıldığı için, bu metoda karar-yönetimli denilmiştir. Dolayısıyla, bu metot ile üretilen istenen yanıt aslına ne denli benzer olacağı, karar devresinin vereceği doğru kararlara bağlıdır. Bu nedenle, ortalama sembol hata olasılığı belli bir seviyeyi aşmadığı (yüzde 10'dan az olduğu) sürece, verilen kararların dengeleyici katsayılarının uygun biçimde ayarlanabilmesine yetecek kadar doğru olduğu kabul edilir. Eğer belirlenen seviye aşıyor ise, hata işareti dengeleyicinin yanlış ayarlar yapmasına neden olur. Alıcı, böyle bir durum gerçekleşmeden önce, test işaretinin tekrar iletilerek dengeleyici katsayılarının yeni baştan koşullandırılmasını isteyebilir.

6.2 Yinelemeli En Küçük Kareler Kafes Algoritmasını Kullanarak Kanal Dengeleme

Bu bölümde, bilinmeyen bozulma (distortion) yaratan doğrusal dağılımlı (linear dispersive) bir iletim kanalının uyarlamalı dengelenmesi için, yinelemeli en küçük kareler kafes (RLSL) algoritmasının kullanımı incelenecektir. Üzerinde çalışacağımız sistemin blok diyagramı Şekil 6.3'te gösterilmiştir.



Şekil 6.3 Uyarlamalı kanal dengeleme uygulamasına ait blok diyagram (Haykin,1991)

Rasgele sayı üretici (1), kanalın nasıl olduğunu test etmek amacıyla kullanılan $u(n)$ test işaretini üretir. Rasgele sayı üretici (2) ise, kanalın çıkışındaki işarete karışan $v(n)$ toplamsal beyaz gürültü (additive white noise) kaynağı olarak kullanılır. Bu iki rasgele sayı üretici, birbirinden bağımsızdır. Uyarlamalı dengeleyicinin görevi: iletilen $u(n)$ işaretinde kanalın yarattığı bozulmayı, toplamsal beyaz gürültünün varlığına rağmen düzeltmektir. Bu göreve genel bir ifadeyle, kanalın tersini modellemek (inverse modeling) denir.

Kanalın girişine uygulanan rasgele dizi $\{u(n)\}$, eşit olasılıkla üretilen ± 1 değerlerinden oluşur ve bu nedenle de, sıfır ortalamalıdır. Kanalın impuls yanıtı olarak,

$$h(n) = \begin{cases} \frac{1}{2} \left[1 + \cos \left(\frac{2\pi}{W} (n-2) \right) \right] , & n = 1, 2, 3 \\ 0 , & \text{diğer} \end{cases} \quad (6.3)$$

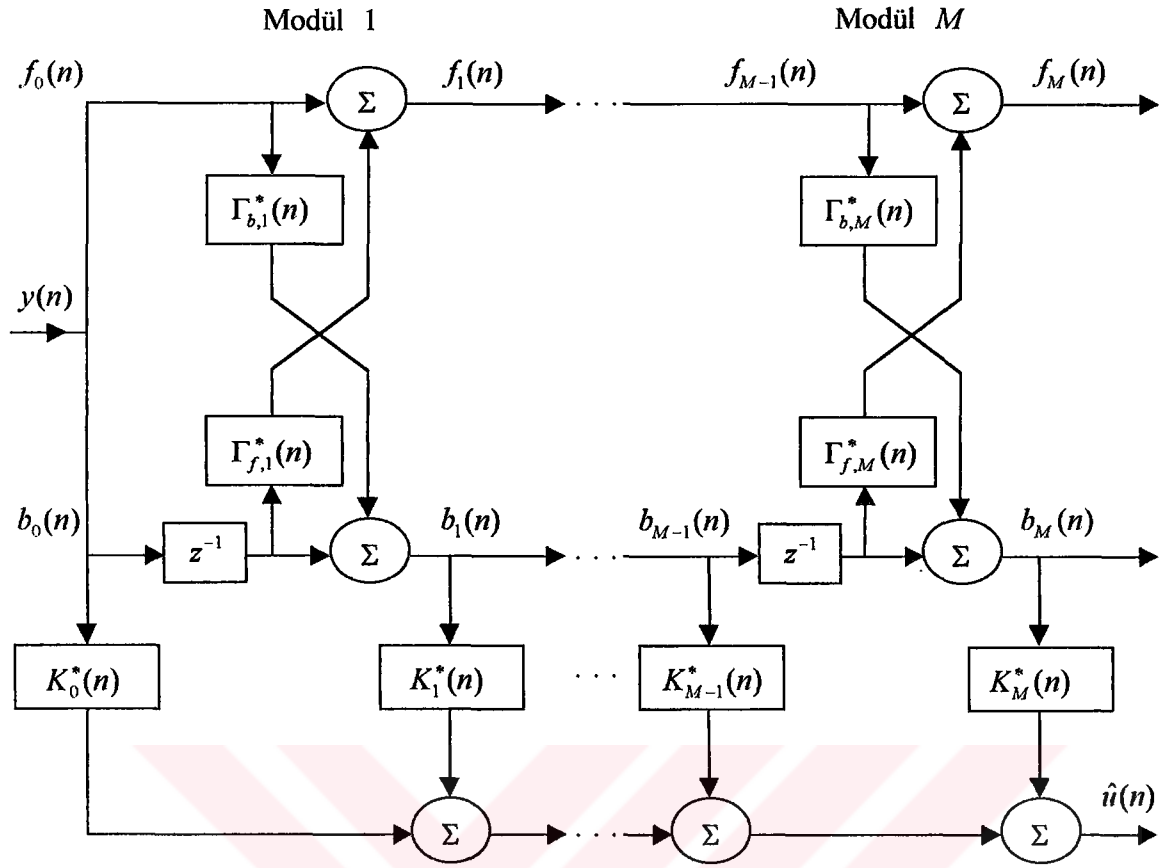
şeklinde tanımlanan yükseltilmiş kosinüs (raised cosine) kullanılacaktır. W parametresi asıl olarak, kanalın yarattığı bozulmanın genliğini kontrol eder. Ancak, kanalın yarattığı bozulma dengeleyici girişindeki işareti etkilediği için, W parametresi dolaylı olarak, dengeleyicinin tap girişlerine ait ilişki matrisinin özdeğer yayılımını da kontrol etmiş olur. W değeri arttıkça, kanalın yarattığı bozulmanın miktarı artar. Bozulma miktarının artması ise, özdeğer yayılımını arttırır.

Rasgele sayı üretici (2) tarafından üretilen rasgele dizi $\{v(n)\}$, sıfır ortalamalı ve σ_v^2 değişintili olup, gerçel sayılardan oluşur. σ_v^2 değeri, kanal çıkışında ölçülen işaret-gürültü oranını da gösterir. Örneğin; $\sigma_v^2 = 0.001$ için, kanal çıkışındaki işaret-gürültü oranı 30 dB olmaktadır. Buraya kadar anlatılanlara göre, uyarlamalı dengeleyicinin n anındaki girişi

$$y(n) = \sum_{k=1}^3 u(n-k) h(k) + v(n) \quad (6.4)$$

olarak yazılabilir.

Uyarlamalı dengeleyici olarak, sonraki kestirim hatalarına dayalı RLSL algoritmasını kullanan birleşik süreç kestirici (Şekil 5.6) kullanılmaktadır; ancak, Şekil 6.3'teki blok diyagrama uygun olması için, Şekil 6.4'teki hale getirilmiştir. Görünümleri farklı olmasına karşın, Şekil 5.6 ve Şekil 6.4 birbirinin aynısıdır. Bu iki şekil arasındaki görünüm farkı, Şekil 5.6'da enine filtre tap ağırlıklarının sonraki kestirim hataları $e_m(n)$ ile ayarlandığının gösterilmesinden kaynaklanır. Birleşik süreç kestiricide öngörü işlemini M modüllü kafes filtre gerçekleştirdiği için, uyarlamalı dengeleyicinin öngörü derecesi M olur. Uyarlamalı dengeleyicinin girişi $y(n)$, kafes filtrenin girişinden; çıkışı $\hat{d}(n) = \hat{u}(n)$ ise, kafes filtrenin altına bağlanan ve kafes filtrenin ürettiği geri yönde öngörü hata dizisini tap girişleri olarak kullanan $(M+1)$ adet tap ağırlıklı enine filtrenin çıkışından yapılmaktadır. $u(n)$ işaretinin, uyarlamalı dengeleyici çıkışına ulaşınca kadar, kanal ve uyarlamalı dengeleyicide toplam D miktar gecikmeye uğradığı düşünülür. Bu nedenle, $u(n)$ işaretini $d(n) = u(n)$ şeklinde doğrudan istenen yanıt olarak kullanmak yerine, kanal ve uyarlamalı dengeleyicide uğradığı toplam gecikme miktarı D kadar geciktirip $d(n) = u(n-D)$ şeklinde istenen yanıt olarak kullanmak daha doğru olur.



Şekil.6.4 Sonraki kestirim hatalarına dayalı RLSL algoritmasını kullanan birleşik süreç kestiricinin Şekil 6.3'teki blok diyagrama uygun hali

Uyarlamalı bir enine filtrenin tap ağırlıkları, *öğrenme modu* boyunca kestirim hataları kullanılarak yapılan sürekli ayarlamalar sonucunda optimum değerlerine ulaşırlar ve böylece, enine filtre *kalıcı halde* çalışmaya başlamış olur. Giriş verisinin istatistiksel karakteristiklerinde herhangi bir değişiklik olduğu zaman, öğrenme modu yeniden başlar (*öğrenme modu = eğitim metodu + karar-yönetimli metot*). (Uyarlamalı veya uyarlamasız) bir enine filtrenin impuls yanıtı, o filtrenin optimum tap ağırlıkları ile tanımlanır. Öyleyse, bir enine filtrenin impuls yanıtının uzunluğu, tap ağırlıklarının sayısına eşittir. Uyarlamalı dengeleyici olarak kullanılan Şekil 6.4'teki birleşik süreç kestiricinin impuls yanıtı $w(n)$ olsun. Çıkış $(M+1)$ adet tap ağırlıklı enine filtre tarafından belirlendiğine göre, $w(n)$ 'in uzunluğu $(M+1)$ olacak demektir. Ancak, çıkışın enine filtre tarafından belirleniyor olması, $w(n)$ 'in alacağı $(M+1)$ adet değerin doğrudan $\{K_0, K_1, \dots, K_M\}$ gerileme katsayılarına eşit olacağı anlamına gelmez; çünkü, enine filtrenin tap girişlerini kafes filtre belirlemektedir. Kafes filtre olmasaydı ve enine filtre tek başına uyarlamalı dengeleyici olarak kullanılsaydı, $w(n)$ 'in alacağı değerler doğrudan gerileme katsayılarına eşit olurdu. Bu durumda,

uyarlamalı dengeleyicinin impuls yanıtı $w(n)$ için şu sonuç ortaya çıkar: uzunluğu, enine filtrenin tap ağırlıklarının sayısına $(M+1)$ eşittir ama aldığı değerler, bu tap ağırlıklarına $\{K_0, K_1, \dots, K_M\}$ eşit olmamaktadır. $w(n)$ 'in aldığı değerleri bulmak için, dengeleyicinin n anındaki çıkışı $\hat{u}(n)$ ile n anındaki tap girişleri $\{y(n), y(n-1), \dots, y(n-M)\}$ arasında yazılacak eşitliğe ihtiyaç duyulur.

Kanalın impuls yanıtı $h(n)$

$$n = \frac{(\text{Kanalın impuls yanıtının uzunluğu}) + 1}{2} \quad (6.5)$$

anına göre *simetrik* oluyor ise, uyarlamalı dengeleyicinin impuls yanıtı $w(n)$: M için seçilen değer tek sayı olması durumunda,

$$n = \frac{(M+1)}{2} + 1 \quad (6.6)$$

anına göre *yaklaşık olarak simetrik*; M için seçilen değerin çift sayı olması durumunda ise,

$$n = \frac{M}{2} + 1 \quad (6.7)$$

anına göre *tam simetrik* olmaktadır. Buna göre; uyarlamalı dengeleyici için gereken istenen yanıtı sağlarken, $u(n)$ işaretinin toplam

$$D = \begin{cases} \frac{(\text{Kanalın impuls yanıtının uzunluğu}) + 1}{2} + \frac{(M+1)}{2} & , \quad M \text{ tek sayı ise} \\ \frac{(\text{Kanalın impuls yanıtının uzunluğu}) + 1}{2} + \frac{M}{2} & , \quad M \text{ çift sayı ise} \end{cases} \quad (6.8)$$

örnek geciktirilmesi gerekir. Örneğin; kanalın impuls yanıtının (6.3) eşitliğindeki gibi tanımlandığı ve uyarlamalı dengeleyicideki kafes filtrenin $M = 10$ modüllü olduğu kabul edilsin. Kanalın impuls yanıtı $n = 2$ anına göre simetrik olduğundan, uyarlamalı dengeleyici için gereken istenen yanıtı sağlamak üzere $u(n)$ işareti toplam

$$D = \frac{(3+1)}{2} + \frac{10}{2}$$

$$D = 7$$

örnek geciktirilmelidir, yani $d(n) = u(n-7)$ olmalıdır.

Hem RLS hem de RLST algoritmasında, *durağan* giriş verisi için üstel ağırlık çarpanı $\lambda = 1$ alınır. Başlangıç koşullarında kullanılan δ sabiti için ise, genellikle 0.004 değeri tercih edilir.

6.2.1 Uyarlamalı dengeleyicinin öğrenme eğrisi hakkında kısa bilgi

Uyarlamalı dengeleyicinin çalışma prensibi için RLST algoritmalarının Çizelge 5.2’de özetlenen sonraki öngörü ve kestirim hatalarına $\{f_m(n), b_m(n), e_m(n)\}$ dayalı versiyonunun kullanılmasına karşın, öğrenme eğrisinin çiziminde önceki kestirim hatası $\alpha_{M+1}(n)$ ’den yararlanılır. Sonraki kestirim hatası $e_{M+1}(n)$ yerine önceki kestirim hatası $\alpha_{M+1}(n)$ ’in kullanılmasının nedeni, daha önce Bölüm 5.2.5.3’te açıklanan neden ile aynıdır: RLST algoritmasını LMS ve RLS algoritmaları ile kıyaslayabilmek için, öğrenme eğrisinin bu iki algoritmanın öğrenme eğrilerine benzemesini sağlamak. Bunun için de, (5.111) eşitliği kullanılarak yazılan

$$\alpha_{M+1}(n) = \frac{e_{M+1}(n)}{\gamma_{M+1}(n)} \quad (6.9)$$

dönüşüm ifadesinden yararlanılır.

Bir algoritmanın öğrenme eğrisi, o algoritmanın ürettiği kestirim hatasına ait karesel ortalama değerin n iterasyon sayısına göre çizilmesiyle elde edilir. RLST algoritmasının öğrenme eğrisi için kullanılacak kestirim hatası $\alpha_{M+1}(n)$ olarak seçildiğine göre, karesel ortalama değerinin $E[|\alpha_{M+1}(n)|^2]$ hesaplanması gerekir. Uyarlamalı kanal dengeleme deneyinde (Şekil 6.3), $\alpha_{M+1}(n)$ ’in karesel ortalama değeri (yani karesinin ensemble ortalaması) şu şekilde hesaplanır:

1. Çizelge 5.2’de $u(n)$ yerine $y(n)$ ve $d(n)$ yerine de $u(n-D)$ alınır. D değeri, kanalın impuls yanıtı $h(n)$ ’in simetrik olması durumunda, (6.8) eşitliğinden kolayca hesaplanır.
2. $n = 1, 2, \dots, \infty$ olması gerektiği için, S değeri yeterince büyük seçilerek $n = 1, 2, \dots, S$ şeklinde iterasyon sayısı belirlenir.
3. 1 ve 2’de belirlenen değerler kullanılarak, RLSSL algoritması ile 100 bağımsız deneme gerçekleştirilir. Diğer bir deyişle, RLSSL algoritması kurulan bir döngü ile $\text{Den} = 100$ defa çalıştırılır. Burada dikkat edilecek nokta: her bir bağımsız denemenin farklı $u(n)$ işareti ile gerçekleştirildiği; buna karşın, iterasyon sayısı S ve kanalın impuls yanıtı $h(n)$ ’in 100 bağımsız deneme boyunca hep aynı kaldığıdır.
4. 100 bağımsız deneme gerçekleştirilirken: (6.9) eşitliği ile hesaplanan $\alpha_{M+1}(n)$ ’in her bir bağımsız denemede $n = 1, 2, \dots, S$ anları için aldığı değerlerin karesi, bir sonraki bağımsız denemede aynı $n = 1, 2, \dots, S$ anları için alacağı değerlerin karesi ile toplanır.
5. 4’te yapılan genel toplam alma işlemi sayesinde, 100 bağımsız deneme sonunda herhangi bir n anı için $|\alpha_{M+1}(n)|^2$ değeri

$$|\alpha_{M+1}(n)|^2 = |\alpha_{M+1}(n)|_1^2 + |\alpha_{M+1}(n)|_2^2 + \dots + |\alpha_{M+1}(n)|_{100}^2 \quad (6.10)$$

haline gelir. Den, tek bir deney için gerçekleştirilen bağımsız deneme sayısı ve $|\alpha_{M+1}(n)|_i^2$ ise, $\alpha_{M+1}(n)$ ’in i . bağımsız denemede n anı için aldığı değerin karesi olmak üzere, farklı Den değerleri için (6.10) eşitliği

$$|\alpha_{M+1}(n)|^2 = \sum_{i=1}^{\text{Den}} |\alpha_{M+1}(n)|_i^2 \quad (6.11)$$

şeklinde geliştirilebilir.

6. Böylece, uyarlamalı kanal dengeleme deneyi için gerçekleştirilen 100 (veya Den) adet bağımsız deneme sonunda, $\alpha_{M+1}(n)$ ’in n anındaki karesel ortalama değeri

$$E[|\alpha_{M+1}(n)|^2] = \frac{\sum_{i=1}^{\text{Den}} |\alpha_{M+1}(n)|_i^2}{\text{Den}} \quad (6.12)$$

olarak hesaplanır.

$E[|\alpha_{M+1}(n)|^2]$ değerine, (önceki) kestirim hatasının karesel ortalaması demek yerine, tıpkı Wiener filtreler ve RLS algoritmasında olduğu gibi, kısaca *karesel ortalama hata* denir ve uyarlamalı dengeleyicinin *istatistiksel başarı göstergesi* olarak kullanılır.

6.2.2 Uyarlamalı dengeleyicinin impuls yanıtının hesaplanması

Uyarlamalı dengeleyicinin görevi: iletilen $u(n)$ işaretinde kanalın yarattığı bozulmayı, toplamsal beyaz gürültüye rağmen düzeltmektir. Dengeleyici bu düzeltmeyi, kendi transfer fonksiyonu $W(w)$ 'i kanalın transfer fonksiyonu $H(w)$ 'in tersine mümkün olabildiğince benzeterek yapar; ideal durum olarak, $W(w)=1/H(w)$ olması istenir. Öyleyse, $|W(w)|$ eğrisi $|1/H(w)|$ eğrisine ne kadar çok benziyor ise, uyarlamalı dengeleyici de o kadar başarılı çalışıyor demektir. Böyle bir karşılaştırmanın yapılabilmesi için, $W(w)$ ve $H(w)$ fonksiyonlarının bulunması gerekir. $W(w)$ fonksiyonu, uyarlamalı dengeleyicinin impuls yanıtı $w(n)$ 'e ve $H(w)$ fonksiyonu ise, kanalın impuls yanıtı $h(n)$ 'e Ayrık Zamanlı Fourier Dönüşümü (Discrete Time Fourier Transform, DTFT) uygulanarak bulunur. Uyarlamalı kanal dengeleme deneyinde (Şekil 6.3), kanalın impuls yanıtını biz seçtiğimiz için, $h(n)$ bilinmektedir; fakat, $w(n)$ 'in ayrıca hesaplanması gerekir.

Uyarlamalı dengeleyicinin yansıma $\{\Gamma_{f,m}(n), \Gamma_{b,m}(n)\}$ ve gerileme $\{K_m(n)\}$ katsayıları öğrenme modu boyunca sürekli değiştiğinden, öğrenme modu sırasında $w(n)$ hesaplanamaz; kestirim hataları kullanılarak yapılan ayarlamaların bitip, dengeleyicinin kalıcı halde çalışmaya başlaması beklenir. Çünkü kalıcı hal, dengeleyiciye ait yansıma ve gerileme katsayılarının optimum değerlerine ulaştıkları ve giriş verisinin karakteristiklerinde meydana gelebilecek herhangi bir değişikliğe kadar da sabit kaldıkları çalışma durumudur. Öyleyse, hesaplamalarda kolaylık sağlamak amacıyla: $n = 1, 2, \dots, S$ değerlerinden kalıcı hal için $n = S$ seçerek, $\{\Gamma_{f,m}(n)\}$ ve $\{\Gamma_{b,m}(n)\}$ yerine $\{\Gamma_{f,m}(S) = \Gamma_{f,m}\}$ ve $\{\Gamma_{b,m}(S) = \Gamma_{b,m}\}$ sabit değerlerini; enine filtrenin tap ağırlıkları olan $\{K_m(n)\}$ gerileme katsayıları yerine de $\{K_m(S) = K_m\}$ sabit değerlerini kullanabiliriz. Ayrıca, $u(n)$, $h(n)$ ve $v(n)$ 'in gerçel değerli olmalarından dolayı, yansıma ve gerileme katsayıları da gerçel değerli olmaktadır. M modüllü kafes filtreden oluşan, $w(n)$ impuls yanıtı ve $v(n)$ girişli uyarlamalı dengeleyicinin (Şekil 6.4) çıkışı

$$\begin{aligned}
\hat{u}(n) &= \sum_{k=0}^M y(n-k) w(k) \\
&= [y(n), y(n-1), \dots, y(n-M)] \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(M) \end{bmatrix}
\end{aligned} \tag{6.13}$$

şeklinde yazılabileceği gibi,

$$\begin{aligned}
\hat{u}(n) &= \sum_{i=0}^M b_i(n) K_i \\
&= [b_0(n), b_1(n), \dots, b_M(n)] \begin{bmatrix} K_0 \\ K_1 \\ \vdots \\ K_M \end{bmatrix}
\end{aligned} \tag{6.14}$$

şeklinde de yazılabilir. (6.13) ve (6.14) eşitliklerinin sol tarafı aynı olduğuna göre, $w(n)$ impuls yanıtını oluşturan $\{w(0), w(1), \dots, w(M)\}$ değerleri

$$[y(n), y(n-1), \dots, y(n-M)] \begin{bmatrix} w(0) \\ w(1) \\ \vdots \\ w(M) \end{bmatrix} = [b_0(n), b_1(n), \dots, b_M(n)] \begin{bmatrix} K_0 \\ K_1 \\ \vdots \\ K_M \end{bmatrix} \tag{6.15}$$

eşitliğinden kolayca hesaplanır. Aşağıda, kafes filtre modül sayısının $M = 1$ ve $M = 2$ olması durumunda uyarlamalı dengeleyicinin sahip olacağı $w(n)$ impuls yanıtları örnek olarak hesaplanmıştır.

(5.144) ve (5.146) derece güncelleme yinelemeleri, uyarlamalı dengeleyicinin kalıcı halde çalışmaya başlaması sonucu

$$f_m(n) = f_{m-1}(n) + \Gamma_{f,m} b_{m-1}(n-1) \quad , \quad m = 1, 2, \dots, M \tag{6.16}$$

$$b_m(n) = b_{m-1}(n-1) + \Gamma_{b,m} f_{m-1}(n) \quad , \quad m = 1, 2, \dots, M \tag{6.17}$$

haline gelir. Başlangıç koşulları ise, $f_0(n) = b_0(n) = y(n)$ olmaktadır.

$M = 1$ için:

$$f_1(n) = y(n) + \Gamma_{f,1} y(n-1)$$

$$b_1(n) = y(n-1) + \Gamma_{b,1} y(n)$$

olur. Dengeleyici çıkışı; (6.13) eşitliğinden,

$$\hat{u}(n) = [y(n), y(n-1)] \begin{bmatrix} w(0) \\ w(1) \end{bmatrix}$$

şeklinde ve (6.14) eşitliğinden de,

$$\hat{u}(n) = b_0(n) K_0 + b_1(n) K_1$$

şeklinde yazılır. $b_0(n)$ ve $b_1(n)$ öngörü hataları yerlerine yazıldığında,

$$\begin{aligned} \hat{u}(n) &= (K_0 + K_1 \Gamma_{b,1}) y(n) + K_1 y(n-1) \\ &= [y(n), y(n-1)] \begin{bmatrix} K_0 + K_1 \Gamma_{b,1} \\ K_1 \end{bmatrix} \end{aligned}$$

elde edilir. Buradan, $M = 1$ için $w(n)$ impuls yanıtını oluşturan $\{w(0), w(1)\}$ değerleri

$$\begin{bmatrix} w(0) \\ w(1) \end{bmatrix} = \begin{bmatrix} K_0 + K_1 \Gamma_{b,1} \\ K_1 \end{bmatrix}$$

olarak bulunur.

$M = 2$ için:

$$f_2(n) = f_1(n) + \Gamma_{f,2} b_1(n-1)$$

$$b_2(n) = b_1(n-1) + \Gamma_{b,2} f_1(n)$$

olur. $f_1(n)$ ve $b_1(n)$ öngörü hatalarının yerlerine yazılmasıyla,

$$f_2(n) = y(n) + (\Gamma_{f,1} + \Gamma_{f,2} \Gamma_{b,1}) y(n-1) + \Gamma_{f,2} y(n-2)$$

$$b_2(n) = \Gamma_{b,2} y(n) + (\Gamma_{b,1} + \Gamma_{b,2} \Gamma_{f,1}) y(n-1) + y(n-2)$$

haline gelir. Dengeleyici çıkışı; (6.13) eşitliğinden,

$$\hat{u}(n) = [y(n), y(n-1), y(n-2)] \begin{bmatrix} w(0) \\ w(1) \\ w(2) \end{bmatrix}$$

şeklinde ve (6.14) eşitliğinden de,

$$\hat{u}(n) = b_0(n) K_0 + b_1(n) K_1 + b_2(n) K_2$$

şeklinde yazılır. $b_0(n)$, $b_1(n)$ ve $b_2(n)$ öngörü hataları yerlerine yazıldığında,

$$\begin{aligned} \hat{u}(n) &= (K_0 + K_1 \Gamma_{b,1} + K_2 \Gamma_{b,2}) y(n) + [K_1 + K_2 (\Gamma_{b,1} + \Gamma_{b,2} \Gamma_{f,1})] y(n-1) + K_2 y(n-2) \\ &= [y(n), y(n-1), y(n-2)] \begin{bmatrix} K_0 + K_1 \Gamma_{b,1} + K_2 \Gamma_{b,2} \\ K_1 + K_2 (\Gamma_{b,1} + \Gamma_{b,2} \Gamma_{f,1}) \\ K_2 \end{bmatrix} \end{aligned}$$

elde edilir. Buradan, $M = 2$ için $w(n)$ impuls yanıtını oluşturan $\{w(0), w(1), w(2)\}$ değerleri

$$\begin{bmatrix} w(0) \\ w(1) \\ w(2) \end{bmatrix} = \begin{bmatrix} K_0 + K_1 \Gamma_{b,1} + K_2 \Gamma_{b,2} \\ K_1 + K_2 (\Gamma_{b,1} + \Gamma_{b,2} \Gamma_{f,1}) \\ K_2 \end{bmatrix}$$

olarak bulunur.

$M = 2$ için elde edilen $\{w(0), w(1), w(2)\}$ değerleri, $M = 1$ için elde edilen $\{w(0), w(1)\}$ değerleri ile karşılaştırıldığında, kafes filtrenin modüler yapısının etkisi ortaya çıkar. Çünkü öngörü derecesi $M = 1$ 'den $M = 2$ 'ye arttırılmasına rağmen, bu durum önceki hesaplamaları etkilememiştir. Diğer bir deyişle, önceki hesaplamalar bozulmamış, sadece o hesaplamalara $w(0)$ 'da $K_2 \Gamma_{b,2}$ değeri, $w(1)$ 'de $K_2 (\Gamma_{b,1} + \Gamma_{b,2} \Gamma_{f,1})$ değeri ve fazladan $w(2)$ katılmıştır.

Dolayısıyla, derece tekrar $M = 1$ yapılmak istendiğinde, $M = 2$ modülünden kaynaklanan K_2 gerileme katsayısını içeren tüm terimlerin kaldırılması yeterli olur. Öngörü derecesi $M > 2$ olan uyarlamalı dengeleyiciye ait $w(n)$ impuls yanıtı da, $M = 1$ ve $M = 2$ için kullanılan yöntemle hesaplanır.

Uyarlamalı dengeleyicinin impuls yanıtı: Ek 3'te $M = 6$ için, Ek 4'te $M = 8$ için ve Ek 5'te ise $M = 10$ için hesaplanmıştır. Elde edilen $\{W(1), W(2), \dots, W(M+1)\}$ değerleri, $w(n)$ 'i oluşturan $\{w(0), w(1), \dots, w(M)\}$ değerlerinin bilgisayar ortamındaki karşılıklarıdır. Hesaplamalarda kullanılan SK, SKF ve SKB değişkenleri, K, KF ve KB değişkenlerinin Den adet bağımsız deneme üzerinden alınan ensemble ortalamalarıdır. Bu ortalamaların alınışı, Ek 1'de verilen program incelendiğinde daha iyi anlaşılacaktır: .

1. Ek 1'deki programda: ileri yönde yansıma katsayıları $\Gamma_{f,m}(n)$ için KF(m, n), geri yönde yansıma katsayıları $\Gamma_{b,m}(n)$ için KB(m, n) ve gerileme katsayıları $K_m(n)$ için ise K(m, n) değişkenleri kullanılmaktadır.

2. Bu programda, her bir deney için Den adet bağımsız deneme gerçekleştirilmekte ve her bir bağımsız denemede: K(m, n) değişkeninin $m = 1, 2, \dots, M+1$ ve $n = 2, 3, \dots, SS$ için aldığı değerler, KF(m, n) ve KB(m, n) değişkenlerinin ise $m = 1, 2, \dots, M$ ve $n = 2, 3, \dots, SS+1$ için aldığı değerler, bir sonraki bağımsız denemede aynı m ve n için alacakları değerler ile

$$SK = SK + K(1:M+1, 2:SS);$$

$$SKF = SKF + KF(1:M, 2:SS+1);$$

$$SKB = SKB + KB(1:M, 2:SS+1);$$

şeklinde toplanmaktadır. Buna benzer durum, (6.10) ve (6.11) eşitliklerinde de vardır.

3. Bağımsız denemeler sona erdiğinde, SK, SKF ve SKB genel toplamaları bağımsız deneme sayısına bölünerek, K, KF ve KB değişkenlerinin ensemble ortalamaları

$$E[K(m, n)] = \frac{SK(m, n)}{\text{Den}}, \quad E[KF(m, n)] = \frac{SKF(m, n)}{\text{Den}}, \quad E[KB(m, n)] = \frac{SKB(m, n)}{\text{Den}}$$

elde edilir.

K(m, n), KF(m, n) ve KB(m, n) değişkenleri, ensemble ortalamaları alınmadan impuls yanıtı hesaplamalarında kullanılsalardı, her bir deney için Den adet bağımsız deneme gerçekleştirmenin bir anlamı kalmazdı. Çünkü bu durumda, hesaplamalarda doğrudan

$K(m, S)$, $KF(m, S)$ ve $KB(m, S)$ değerleri kullanılırdı. Bu değerler ise, baştan itibaren gerçekleştirilen (Den-1) adet bağımsız denemenin herhangi birine değil, sadece en son gerçekleştirilen Den. bağımsız denemeye aittir. Öyleyse, bize lazım olan her değişkenin, ensemble ortalamasını almak gerekir. Aksi taktirde, baştan itibaren gerçekleştirilen (Den-1) adet bağımsız denemenin o değişken için hiçbir anlamı kalmaz ve geriye sadece Den. bağımsız denemede aldığı değerler kalır.

6.3 Bilgisayar Deneyleri

Bu bölümde, doğrusal dağılımlı iletim kanallarının yinelemeli en küçük kareler kafes (RLSL) algoritması ile uyarlamalı dengelenmesi üzerine yapılan bilgisayar deneyleri ele alınacaktır. Bu deneyleri gerçekleştirmek için kullanılan programlar, Matlab ortamında yazılmış olup, Ekler bölümünde sunulmuşlardır.

Bilgisayar deneylerini yapmaktaki amaç, uyarlamalı dengeleyicinin farklı $h(n)$, σ_v^2 ve M değerleri için ne denli başarılı çalıştığını ölçmektir. Bölüm 6.2.1 ve 6.2.2’de de bahsedildiği gibi, uyarlamalı dengeleyicinin başarı göstergesi olarak iki farklı eğriden yararlanılabilir. Bunlardan birincisi, uyarlamalı dengeleyicinin ürettiği kestirim hatasına ait karesel ortalama değer ulaşığı minimum değeri gösteren öğrenme eğrisidir. Bu eğriden ayrıca, uyarlamalı dengeleyicinin içinde çalıştığı ortamın koşullarına kendini ne kadar hızlı adapte edebildiği, diğer bir deyişle, kaç iterasyon sonunda içinde çalıştığı ortamı öğrenebildiği, yani kısacası uyarlamalı dengeleyicinin *yakınsama hızı* da görülür. Eğrilerden ikincisi ise, uyarlamalı dengeleyicinin transfer fonksiyonu $W(w)$ ’e ait genlik eğrisidir. $|W(w)|$ ’in normalize frekans $(w/2)/\pi$ ’e bağlı olarak çizilmesiyle elde edilen bu eğriyi iletim kanalının tersine ait $|1/H(w)|$ eğrisi ile karşılaştırarak, uyarlamalı dengeleyicinin kendisini kanalın tersine ne kadar benzetebildiği görülür.

Yukarıda anlatılanlardan anlaşılacağı gibi, bu tez çalışmasında, uyarlamalı dengeleyicinin başarısı hakkında yorum yapmak için, dengeleyicinin öğrenme eğrisi ve transfer fonksiyonunun genlik eğrisi baz alınmaktadır. Bu nedenle, dengeleyicinin öğrenme eğrisini elde etmek için, Ek 1’de verilen program ve transfer fonksiyonunun genlik eğrisini elde etmek için de, Ek 2’den Ek 8’e kadar verilen programlar hazırlanmıştır. Burada dikkat

edilmesi gereken nokta: Ek 2'den Ek 8'e kadar olan programlara dışardan hiçbir değerin girilmediği; bunun yerine bu programların, Ek 1'deki programın çalıştırılmasıyla elde edilen değişkenlerden bazılarını kullandığıdır. Buna göre, Ek 1'deki program ana programdır ve çalışması biter bitmez, elde ettiği değişkenlerin kalıcı olarak kaydedilmesi gerekir. Ek 1'de verilen program, Şekil 6.3'te gösterilen blok diyagramı farklı $h(n)$, σ_v^2 ve M değerleri ile gerçeklememizi sağlar. Bu blok diyagramın her bir bloğu Bölüm 6.2'de ayrıntılı olarak anlatıldığı için, burada tekrar ele alınmayacaktır; ancak, Ek 1'deki RLSL algoritmasının, Çizelge 5.2'de $\lambda = 1$ ve $\delta = 0.004$ alınarak yazıldığıının tekrar belirtilmesinde fayda vardır.

Yapılan bilgisayar deneylerinde kanal modeli olarak, impuls yanıtı (6.3) eşitliğiyle tanımlanan yükseltilmiş kosinüsün yanı sıra, iki farklı kanal modeli daha kullanılmıştır. Öte yandan, yükseltilmiş kosinüse ait W parametresi, kanalın yarattığı bozulmanın genliğini kontrol ettiği için: uyarlamalı dengeleyicinin giderek artan kanal bozulması karşısında elde edeceği başarıyı ölçmek amacıyla, bu parametreye $W = \{2.9, 3.1, 3.3, 3.5\}$ değerleri verilmiştir. Sonuç olarak, dört tanesi yükseltilmiş kosinüse ait olmak üzere, toplam altı adet doğrusal kanal modelinin uyarlamalı dengelenmesi gerçekleştirilmiştir. Bu altı adet modelin her biri için, kafes filtre modül sayısı $M = \{10, 8, 6\}$ ve beyaz gürültünün değışintisi ise $\sigma_v^2 = \{0.001, 0.01, 0.1\}$ olarak alınmıştır. Diğer bir deyişle, Ek 1'de verilen program kullanılarak, her bir kanal modeli için $s(M) \times s(\sigma_v^2) = 3 \times 3 = 9$ adet deney gerçekleştirilmiştir; bu da, toplam 54 deney demektir. Bu deneylerin hepsinde, bağımsız deneme sayısı $Den = 100$ alınırken; iterasyon sayısı S , öğrenme eğrisinin yakınsamasını net olarak görmemizi sağlayacak şekilde, 200 seçilmiştir. Elde edilen öğrenme eğrileri, tek tek ve karşılaştırmalı olarak Bölüm 6.3.1'de verilmiştir. Ek 1 ile yapılan deneylerden elde edilen değişkenlerin Ek 2'den Ek 8'e kadar olan programlarda kullanılması sonucu, $|W(w)|$, $|H(w)|$ ve $|1/H(w)|$ eğrileri bulunur. Bu eğrilerden $|H(w)|$ eğrileri tek başlarına, $|W(w)|$ ve $|1/H(w)|$ eğrileri ise karşılaştırmalı olarak Bölüm 6.3.2'de verilmiştir. Seçilen kanal modellerinin doğrusal dağılımlı olmalarından başka sahip oldukları bir diğer özellik de, hepsinin bir n anına göre simetrik (6.5) olmalarıdır. Kanalın impuls yanıtı simetrik olduğuna göre, uyarlamalı dengeleyiciye ait $w(n)$ impuls yanıtı da, kafes filtre modül sayısı M 'e bağlı olarak ya yaklaşık simetrik (6.6) veya tam simetrik (6.7) olacaktır. Burada, $w(n)$ impuls yanıtının tam simetrik olmasını sağlamak amacıyla, M değerleri çift sayılardan seçilmiştir.

6.3.1 Öğrenme eğrileri

Uyarlamalı dengelenmesi gerçekleştirilen kanal modellerinin impuls yanıtları şu şekildedir:

1. (6.3) eşitliğinde $W = 2.9$ alınarak elde edilen

$$h(n) = \begin{cases} 0.2194, & n = 1, 3 \\ 1, & n = 2 \end{cases} \quad (6.18)$$

2. (6.3) eşitliğinde $W = 3.1$ alınarak elde edilen

$$h(n) = \begin{cases} 0.2798, & n = 1, 3 \\ 1, & n = 2 \end{cases} \quad (6.19)$$

3. (6.3) eşitliğinde $W = 3.3$ alınarak elde edilen

$$h(n) = \begin{cases} 0.3365, & n = 1, 3 \\ 1, & n = 2 \end{cases} \quad (6.20)$$

4. (6.3) eşitliğinde $W = 3.5$ alınarak elde edilen

$$h(n) = \begin{cases} 0.3887, & n = 1, 3 \\ 1, & n = 2 \end{cases} \quad (6.21)$$

5. Kendi belirlediğimiz

$$h(n) = \begin{cases} 0.2, & n = 1, 7 \\ 0.3, & n = 2, 6 \\ 0.4, & n = 3, 5 \\ 1, & n = 4 \end{cases} \quad (6.22)$$

6. Kendi belirlediğimiz

$$h(n) = \begin{cases} 0.3, & n = 1, 5 \\ 0.2, & n = 2, 4 \\ 1, & n = 3 \end{cases} \quad (6.23)$$

İmpuls yanıtı (6.18) eşitliğiyle verilen kanal modeli için $Den = 100$, $S = 200$, $M = \{10, 8, 6\}$ ve $\sigma_v^2 = \{0.001, 0.01, 0.1\}$ değerleri kullanılarak yapılan deneylere ait öğrenme eğrileri, Şekil 6.5, 6.6 ve 6.7’de verilmiştir. Hepsinde aynı kanal dengelenmeye çalışıldığı için, bu deneylerin genel sabiti $W = 2.9$ olmaktadır. $M = \{10, 8, 6\}$ olmasından dolayı, elimizde üç adet uyarlamalı dengeleyicinin olduğu düşünölsün. Bu durumda yapılması gereken: kanalın yarattığı bozulma ile birlikte güröltünün yarattığı bozulmayı da sabit alıp, dengeleyicilerin her birinin bu eşit bozulma koşulları karşısında elde edeceği başarıyı ölçmektir; yani, sabit W ve σ_v^2 değerleri karşısında M ’in değişimini incelemektir. Şekil 6.5 (a), (b) ve (c), $M = \{10, 8, 6\}$ için $W = 2.9$ ve $\sigma_v^2 = 0.001$; Şekil 6.6 (a), (b) ve (c), $M = \{10, 8, 6\}$ için $W = 2.9$ ve $\sigma_v^2 = 0.01$; Şekil 6.7 (a), (b) ve (c) ise, $M = \{10, 8, 6\}$ için $W = 2.9$ ve $\sigma_v^2 = 0.1$ sabit alınarak elde edilmiş öğrenme eğrileridir. Bu eğrilerin kendi aralarında karşılaştırılmaları ise, Şekil 6.8 ve 6.9’da gösterilmiştir.

İmpuls yanıtı (6.19) eşitliğiyle verilen kanal modeli için aynı Den , S , M ve σ_v^2 değerleri kullanılarak yapılan deneylere ait öğrenme eğrileri, Şekil 6.10, 6.11 ve 6.12’de verilmiştir. Bu eğrilerin kendi aralarında karşılaştırılmaları ise, Şekil 6.13 ve 6.14’te gösterilmiştir.

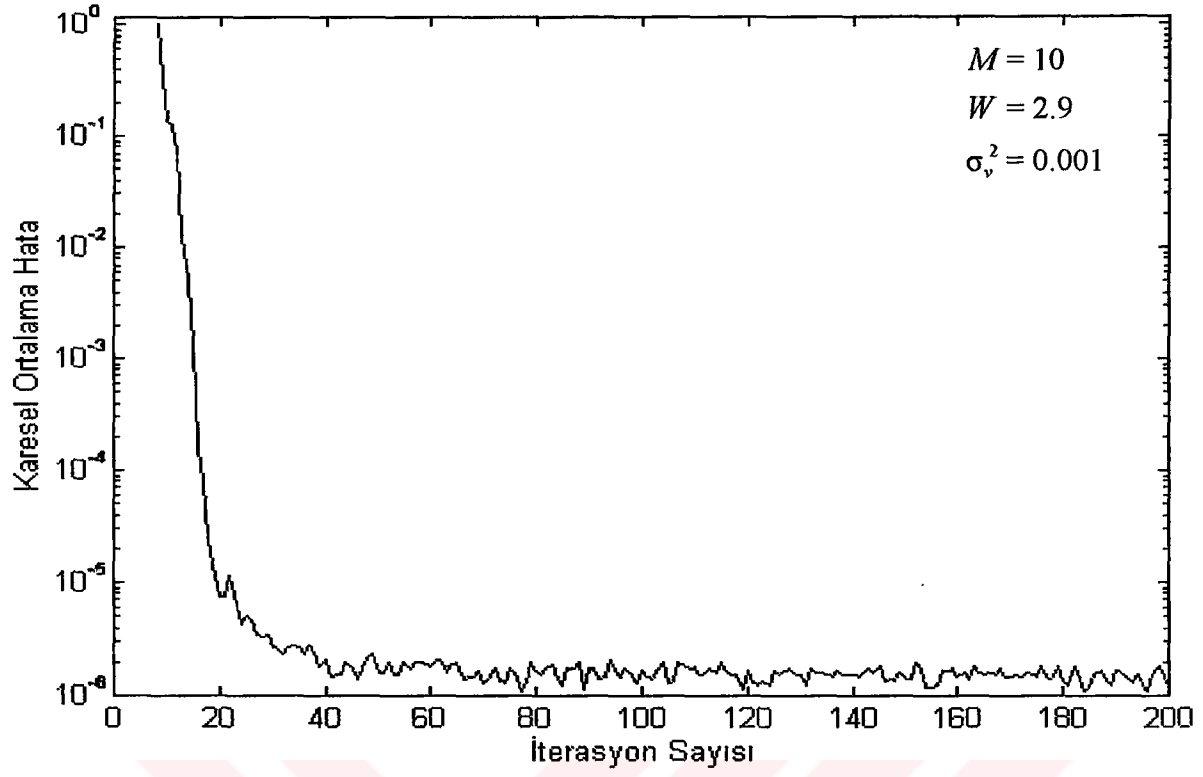
İmpuls yanıtı (6.20) eşitliğiyle verilen kanal modeli için aynı Den , S , M ve σ_v^2 değerleri kullanılarak yapılan deneylere ait öğrenme eğrileri, Şekil 6.15, 6.16 ve 6.17’de verilmiştir. Bu eğrilerin kendi aralarında karşılaştırılmaları ise, Şekil 6.18 ve 6.19’da gösterilmiştir.

İmpuls yanıtı (6.21) eşitliğiyle verilen kanal modeli için aynı Den , S , M ve σ_v^2 değerleri kullanılarak yapılan deneylere ait öğrenme eğrileri, Şekil 6.20, 6.21 ve 6.22’de verilmiştir. Bu eğrilerin kendi aralarında karşılaştırılmaları ise, Şekil 6.23 ve 6.24’te gösterilmiştir.

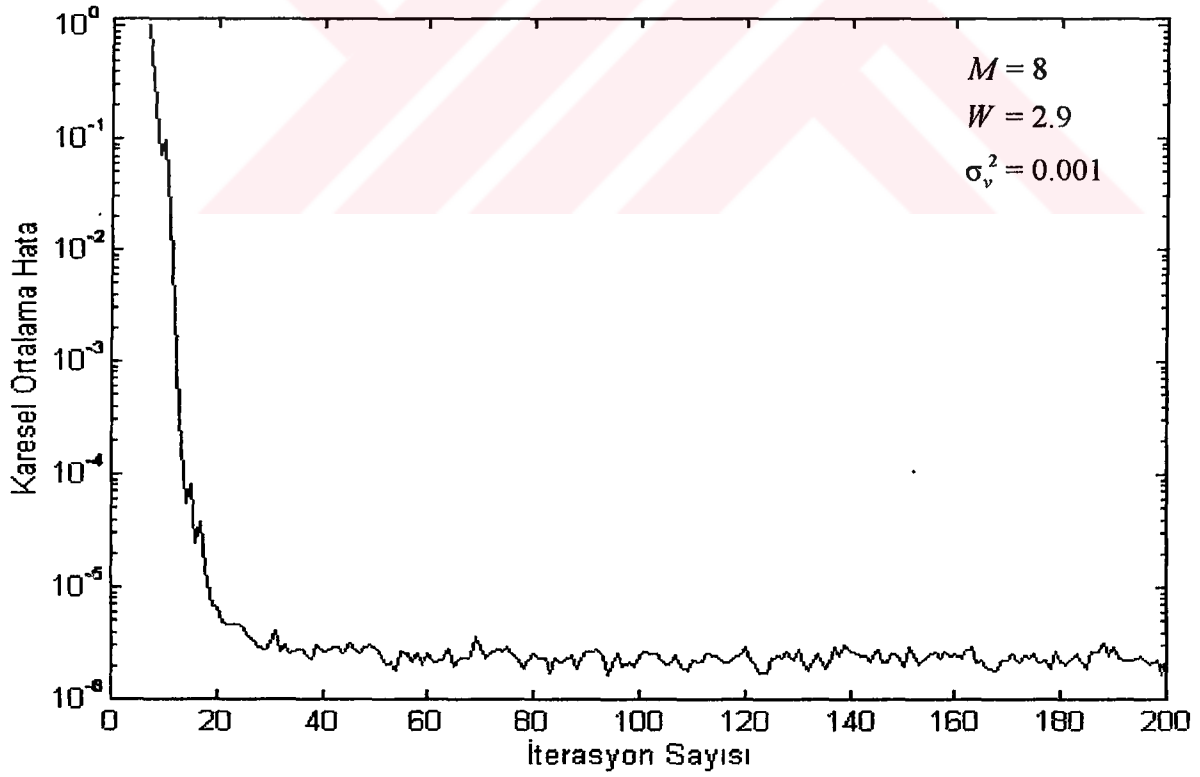
İmpuls yanıtları (6.18), (6.19), (6.20) ve (6.21) eşitlikleriyle verilen kanal modelleri için elde ettiğimiz öğrenme eğrilerini M ve σ_v^2 değerlerini sabit alarak incelediğimizde, W parametresindeki değişimin uyarlamalı dengeleyici üzerindeki etkisini görürüz. Şekil 6.5’ten 6.24’e kadar olan öğrenme eğrilerinin bu amaçla yapılan karşılaştırılmaları, Şekil 6.25, 6.26 ve 6.27’de gösterilmiştir.

Uyarlamalı dengeleyicinin farklı impuls yanıtına sahip kanal modellerine karşı elde edeceği başarıyı ölçmek için, (6.22) ve (6.23) eşitlikleriyle verilen kanal modelleri kullanılmıştır. Bu kanal modellerinin impuls yanıtları (uzunluk, simetri merkezi ve simetri merkezi etrafındaki katsayıların aldığı değerler bakımından) yükseltilmiş kosinüsten farklıdır. Kendi belirlediğimiz bu iki kanal modelinden: impuls yanıtı (6.22) eşitliğiyle verilen kanal modeli için yapılan deneylere ait öğrenme eğrileri ve bu eğrilerin kendi aralarında karşılaştırılmaları, Şekil 6.28, 6.29, 6.30, 6.31 ve 6.32’de; impuls yanıtı (6.23) eşitliğiyle verilen kanal modeli için yapılan deneylere ait öğrenme eğrileri ve bu eğrilerin kendi aralarında karşılaştırılmaları ise, Şekil 6.33, 6.34, 6.35, 6.36 ve 6.37’de gösterilmiştir.

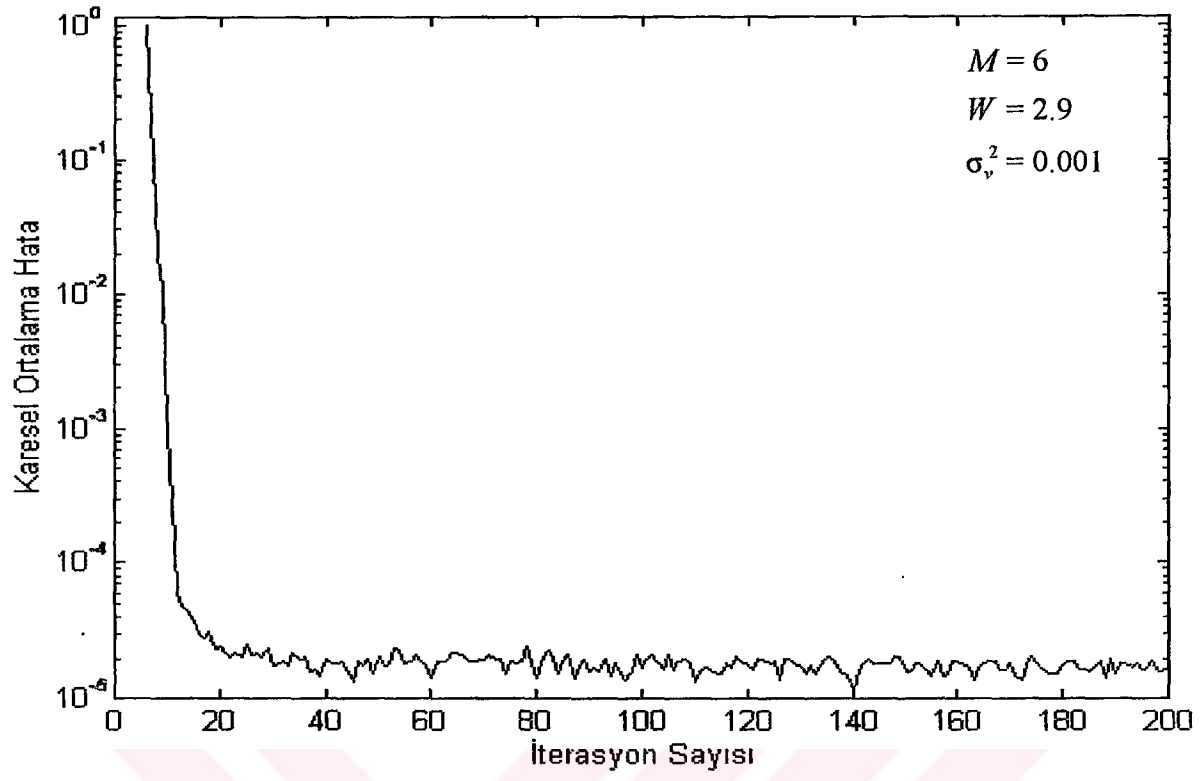




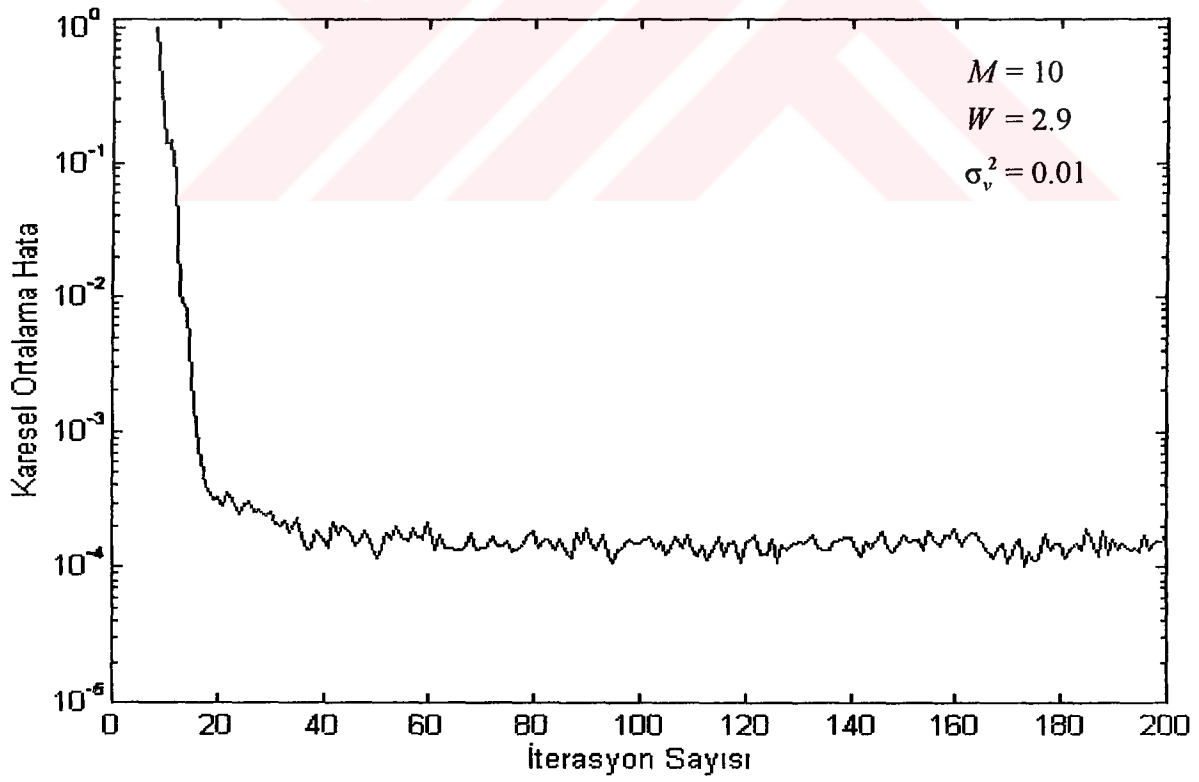
Şekil 6.5 (a)



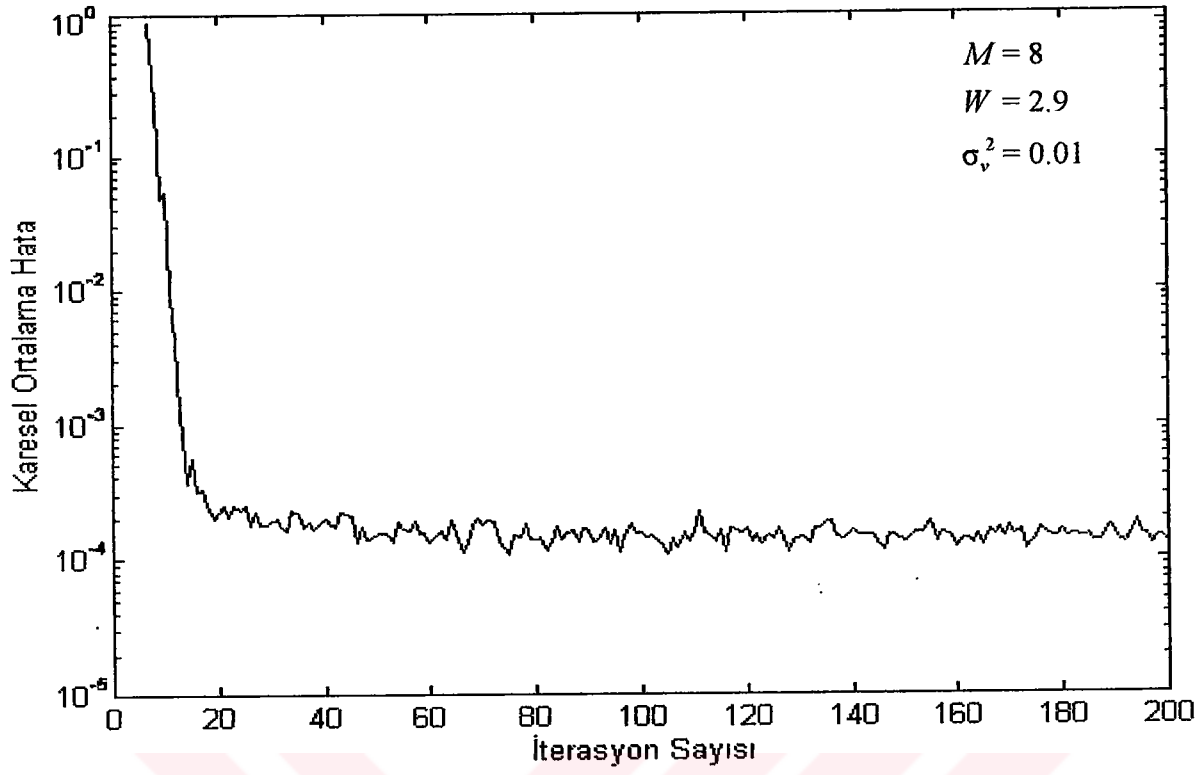
Şekil 6.5 (b)



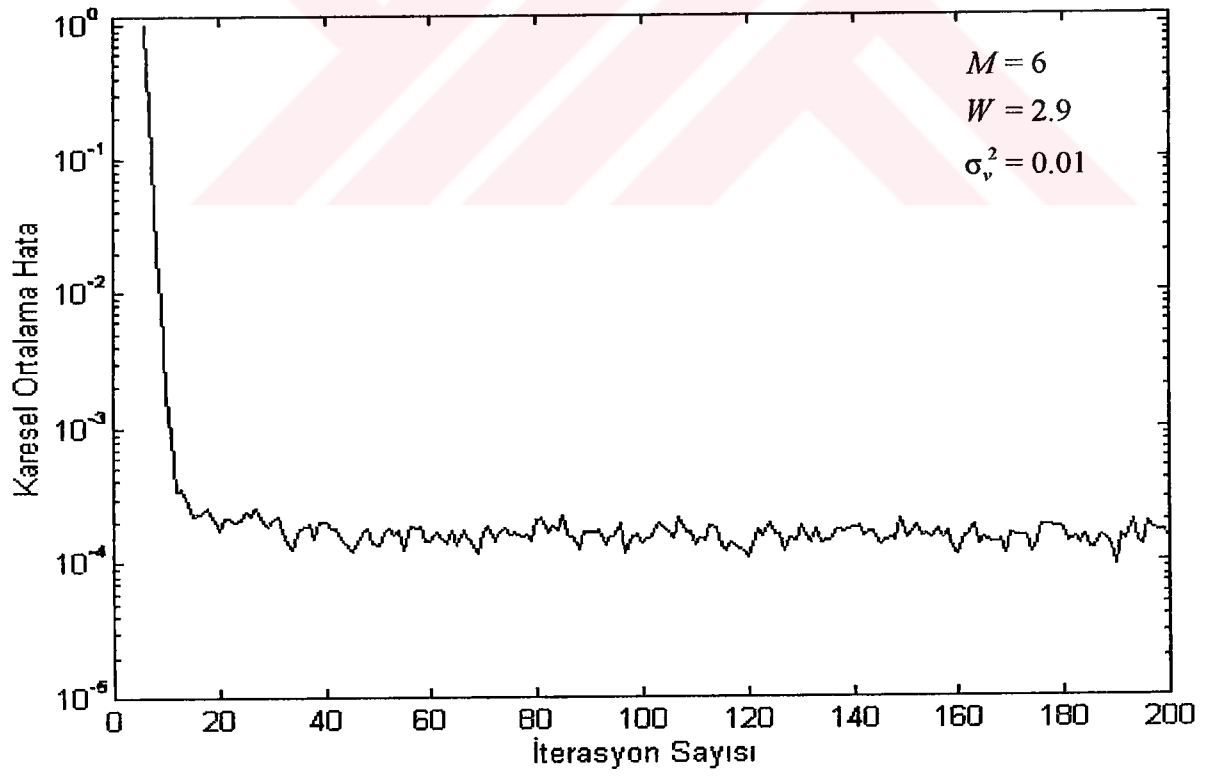
Şekil 6.5 (c)



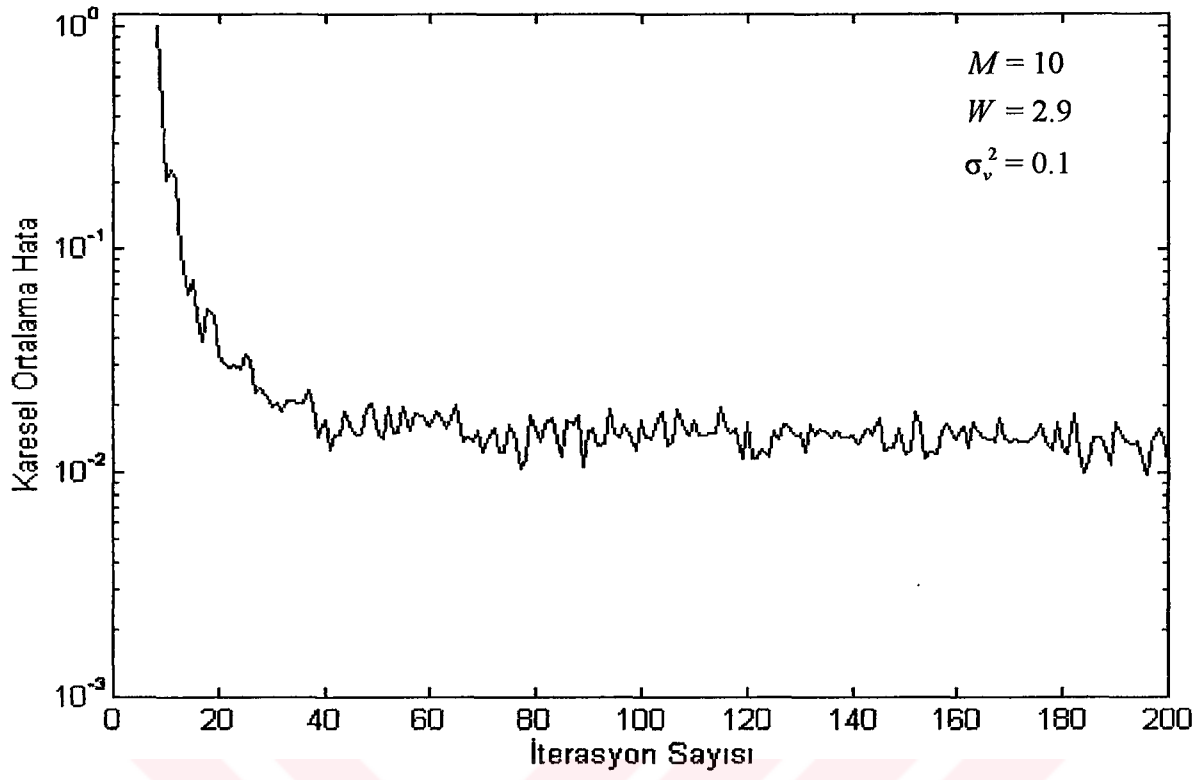
Şekil 6.6 (a)



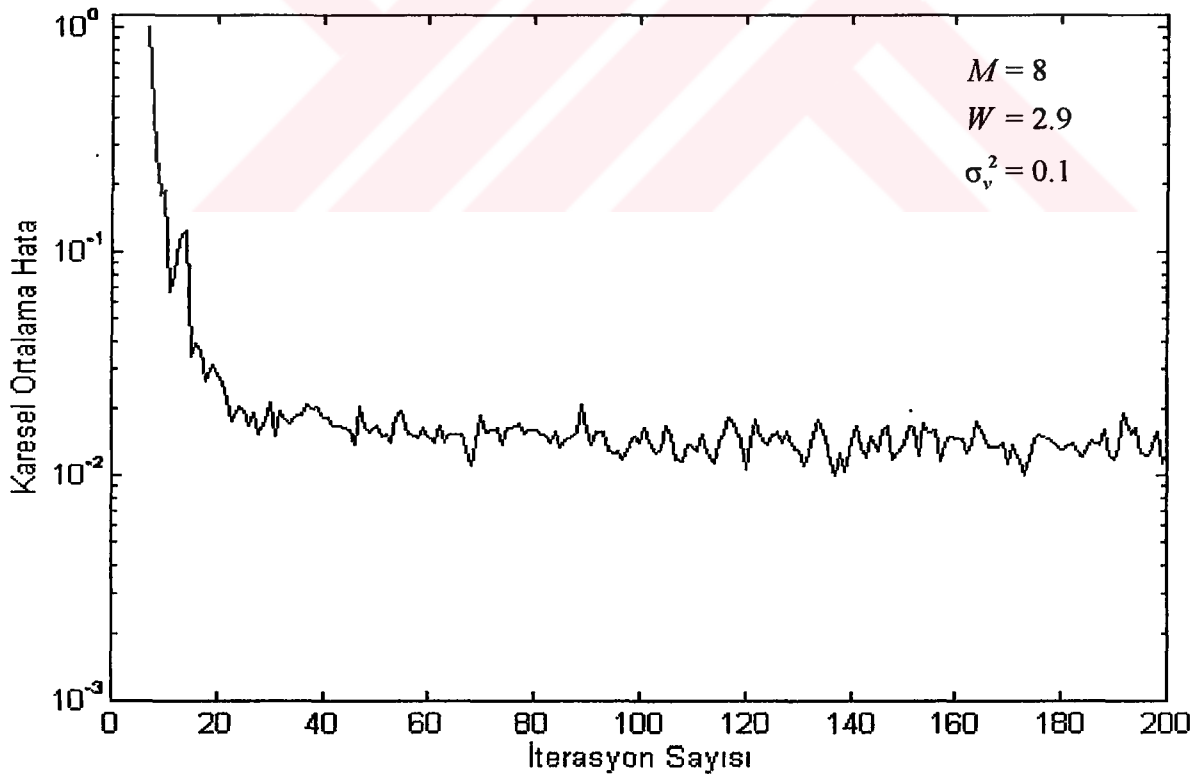
Şekil 6.6 (b)



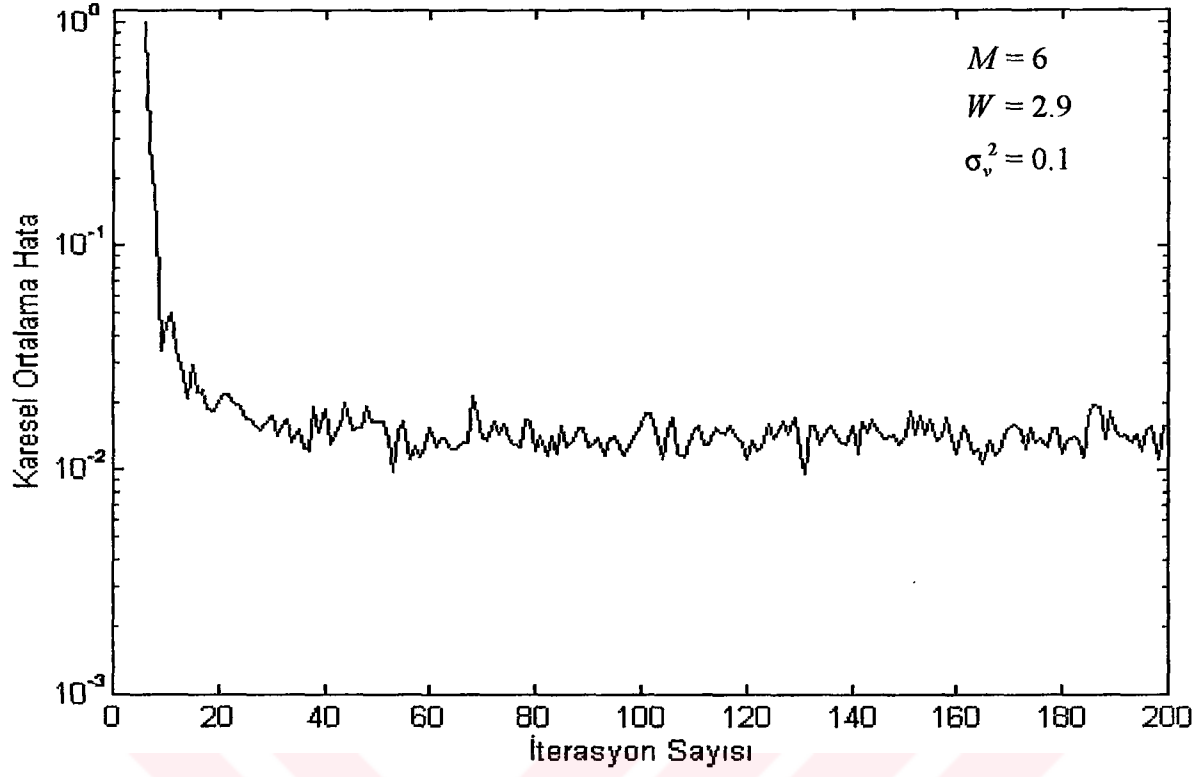
Şekil 6.6 (c)



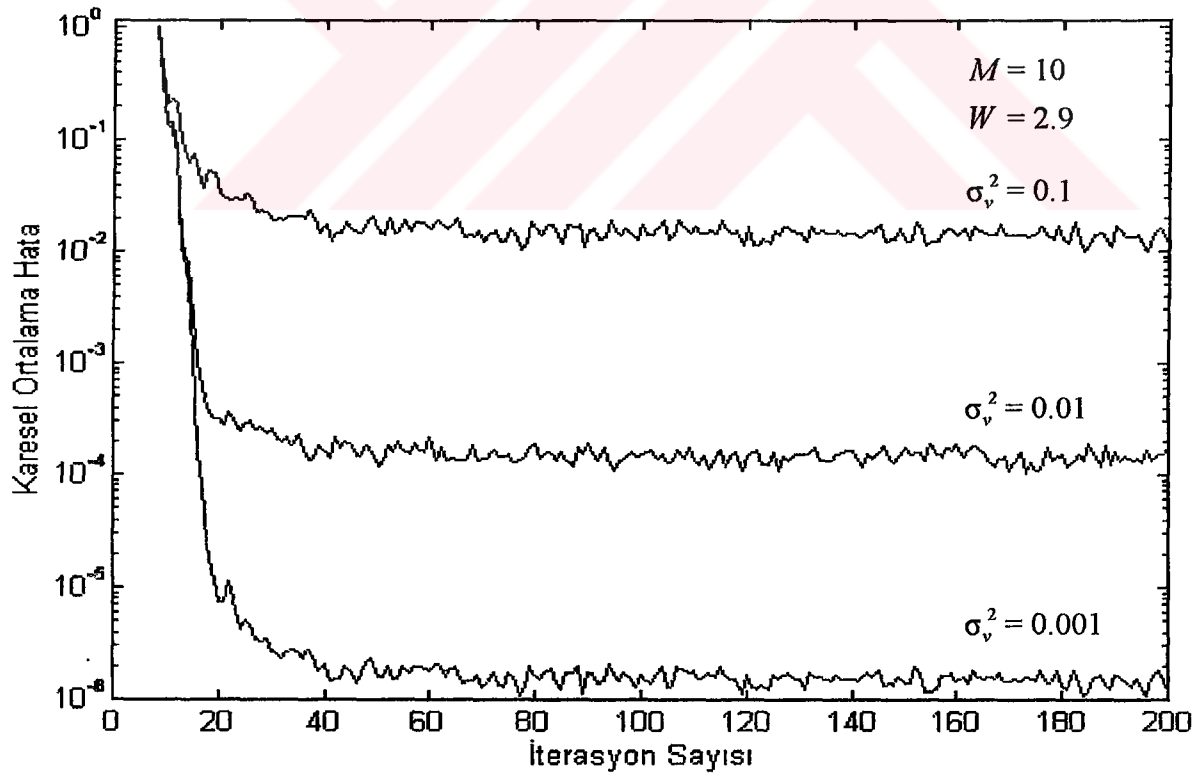
Şekil 6.7 (a)



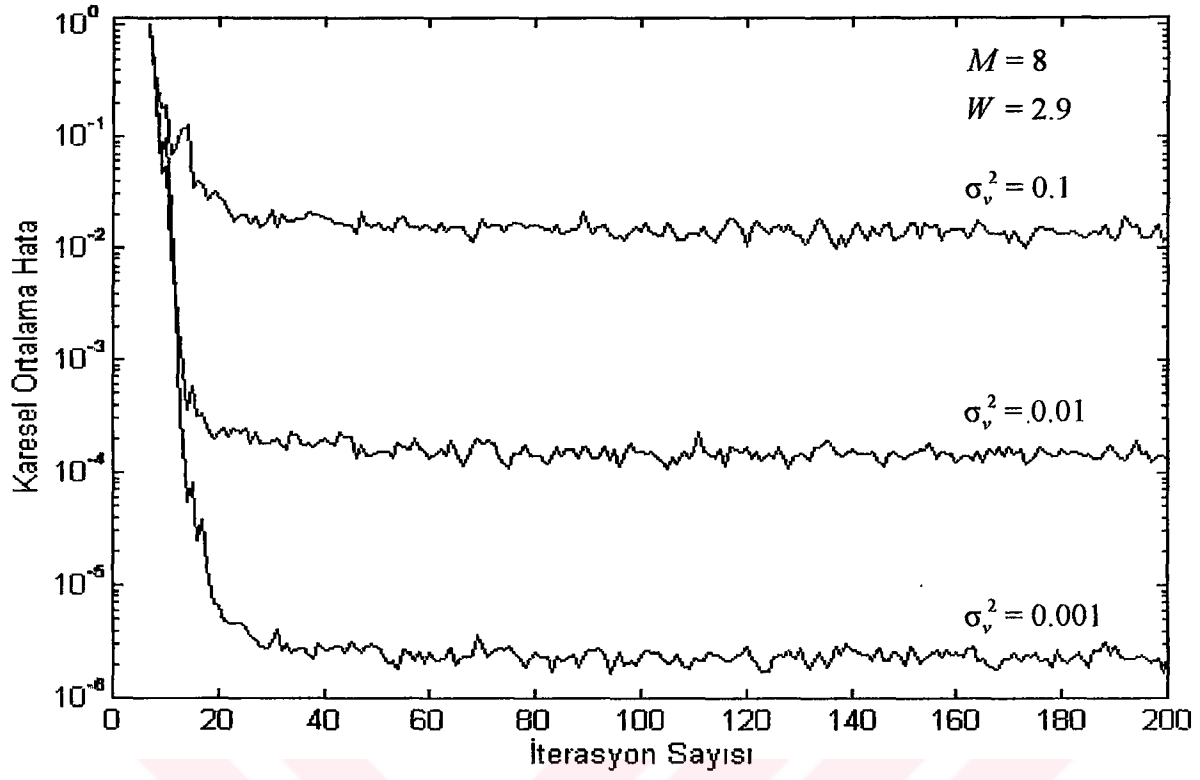
Şekil 6.7 (b)



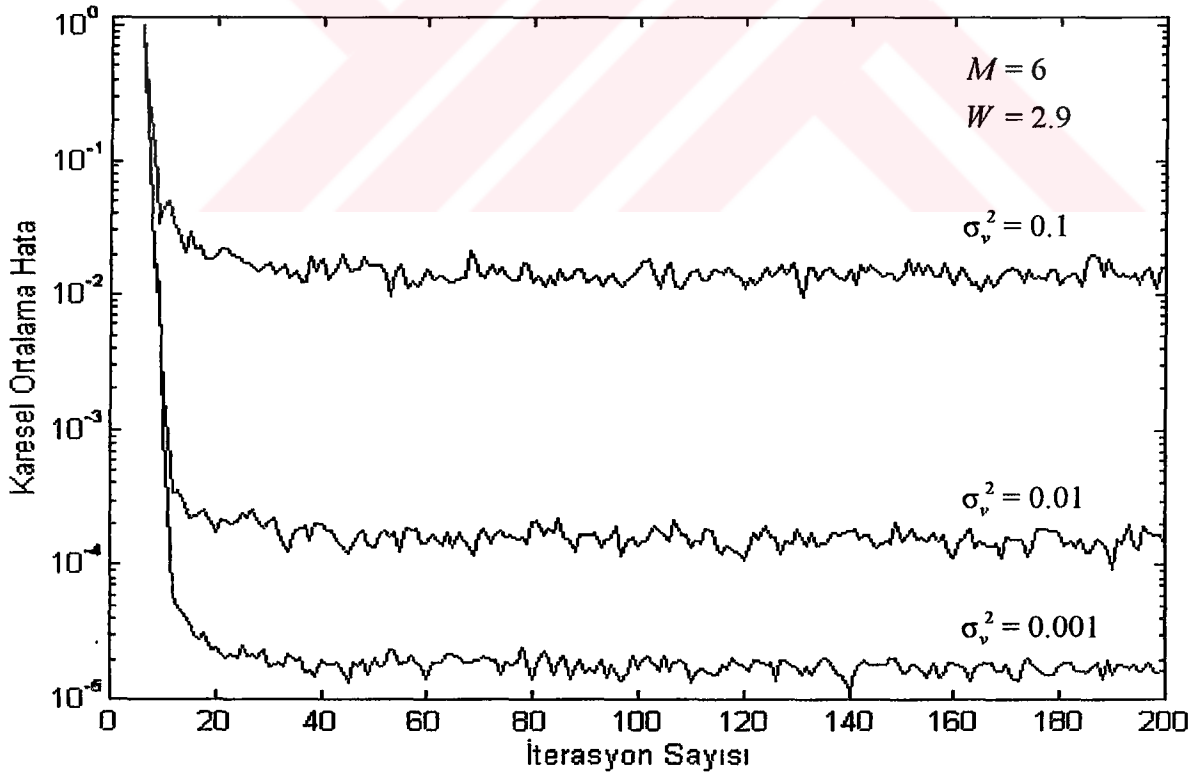
Şekil 6.7 (c)



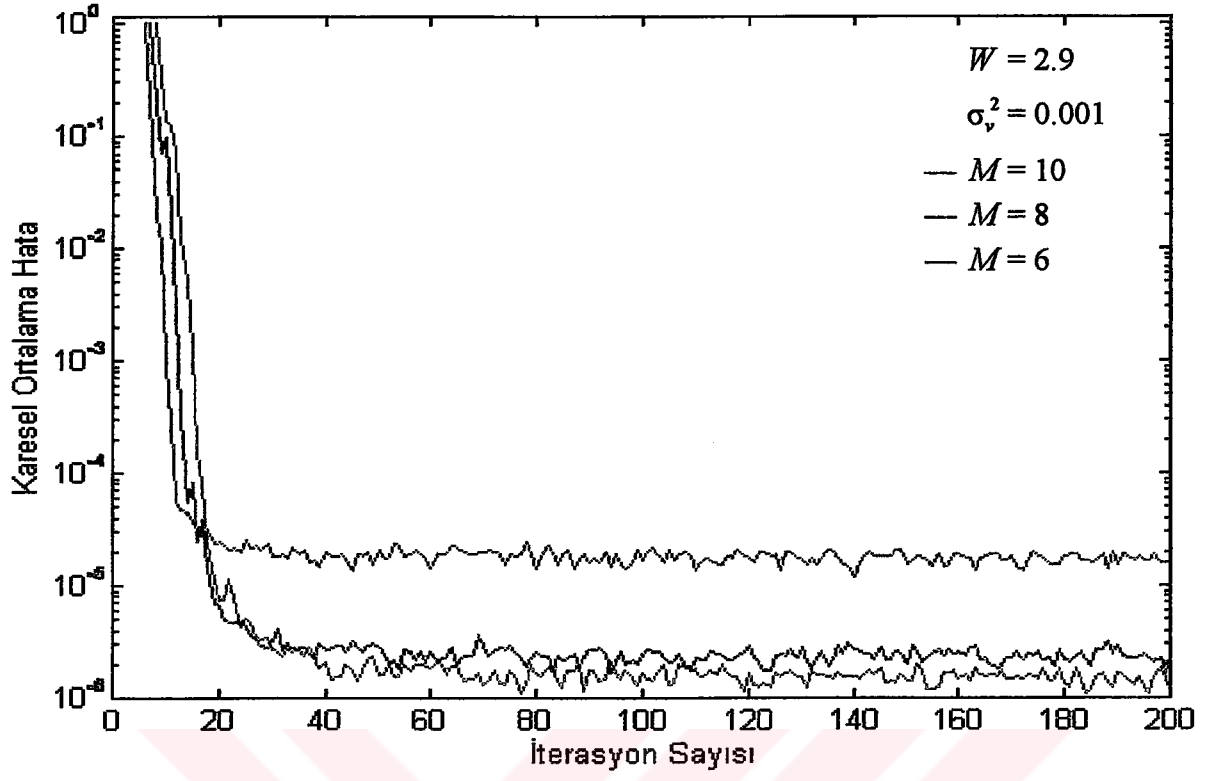
Şekil 6.8 (a)



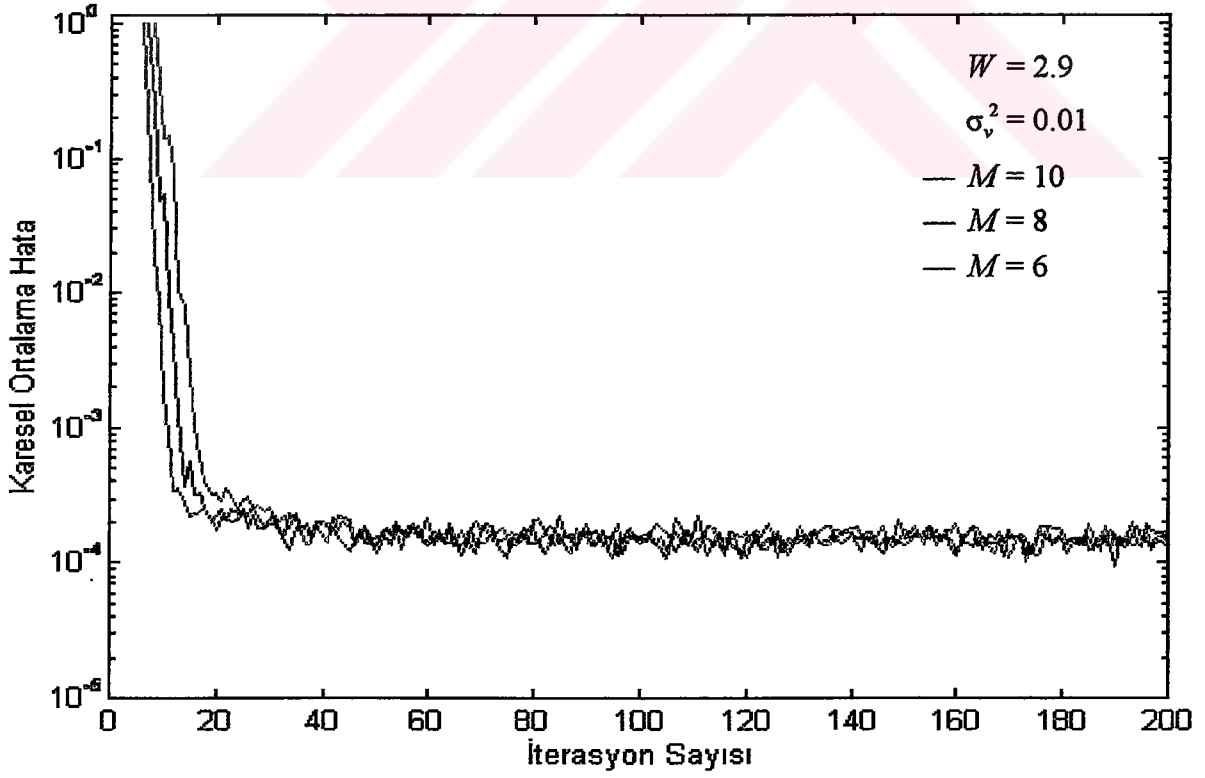
Şekil 6.8 (b)



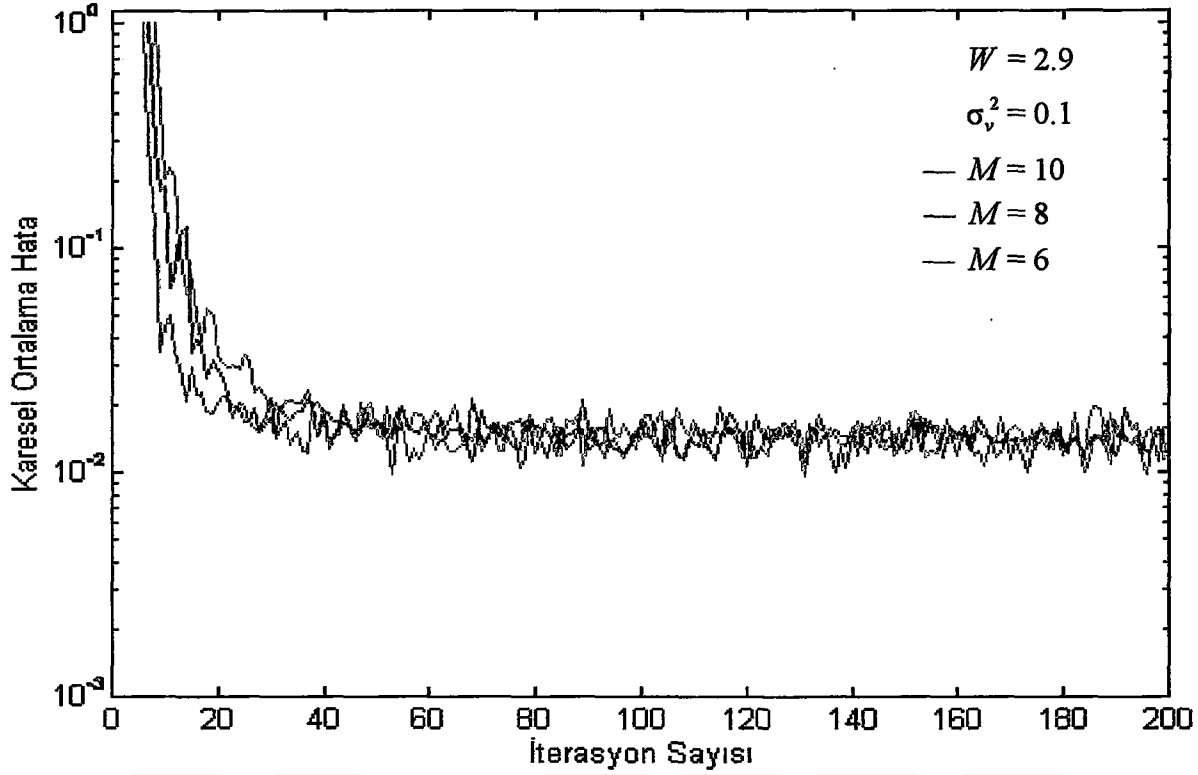
Şekil 6.8 (c)



Şekil 6.9 (a)



Şekil 6.9 (b)



Şekil 6.9 (c)

Kullandığımız uyarlamalı dengeleyici, RLSL algoritmasına dayalı birleşik süreç kestirici (Şekil 6.4) olduğuna göre; kestirim hatasının karesel ortalama değerine ait bu eğrilerin yakınsama analizini yaparak, RLSL algoritmasının karesel ortalama anlamda yakınsaması hakkında bilgi edinilebilir.

Bir algoritmanın karesel ortalama bakımından yakınsaması için gereken iterasyon sayısı az ise, o algoritmanın yakınsama hızı yüksek demektir. Bu aynı zamanda, o algoritmanın içinde çalıştığı ortamın koşullarını kısa bir süre içinde öğrenip, kalıcı halde çalışmaya geçebildiği anlamına da gelmektedir. Eğer algoritmanın yakınsaması için gereken iterasyon sayısı fazla ise, o algoritmanın yakınsama hızı düşüktür. Bu da, o algoritmanın uzun bir öğrenme süresinden sonra kalıcı halde çalışmaya başlayabildiği anlamına gelmektedir.

Şekil 6.8 (a), (b) ve (c) incelendiğinde, RLSL algoritmasının yakınsama süresinin M değeri ile birlikte azaldığı görülür. Örneğin; $\sigma_v^2 = 0.001$ gürültü koşulunda $M = 10$ için yaklaşık 40-45 iterasyon sonunda yakınsarken, $M = 8$ için yaklaşık 35-40 iterasyon sonunda ve $M = 6$ için ise yaklaşık 30-35 iterasyon sonunda yakınsamaktadır. Diğer yandan, Şekil 6.9 (a), (b)

ve (c) incelendiğinde ise, $M = \{10, 8, 6\}$ eğrileri arasındaki yakınsama süresi farklılığının, giderek azalmakta olan işaret-gürültü oranı karşısında ortadan kalktığı ve en sonunda, $\text{SNR} = 10 \text{ dB}$ ($\sigma_v^2 = 0.1$) için her üç eğrinin hemen hemen aynı sürede yakınsadıkları görülmektedir. Bu iki gözlem, şu şekilde yorumlanabilir:

1. İşaret-gürültü oranının yüksek olması durumu (30 dB)

RLSL algoritmasının yakınsama hızı, oldukça yüksektir ve M değeri ile ters orantılı bir şekilde değişmektedir.

2. İşaret-gürültü oranının düşük olması durumu (10 dB)

RLSL algoritmasının yakınsama hızı, bu durumda bile yüksek sayılır. Sadece, M değerinin değişimine karşı olan hassasiyeti ortadan kalkmakta ve her M değeri için hemen hemen aynı sürede yakınsamaktadır.

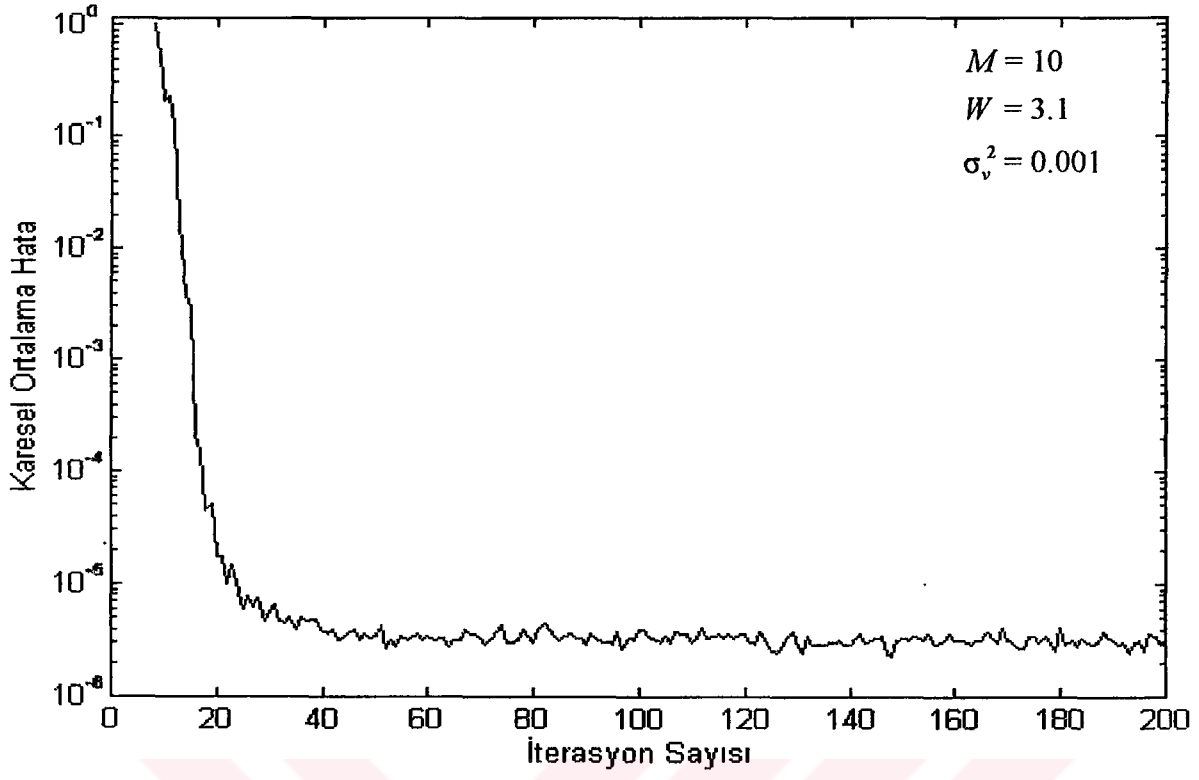
Bu yaptığımız yorumları Bölüm 5.2.5.3'te RLS algoritmasının yakınsama hızı için elde edilen teorik sonuçlarla karşılaştırdığımızda, RLS ve RLSL algoritmalarının yakınsama özelliklerinin aynı olduğunu görürüz.

Karesel ortalama hatanın belli bir değere yakınsamasına kadar olan dönem, algoritmanın içinde çalıştığı ortamı *öğrenme dönemi* olarak; yakınsadığı andan itibaren olan dönem ise, algoritmanın *kalıcı halde çalışma dönemi* olarak tanımlanır. Yukarıda yapılan analizde, RLSL algoritmasının yakınsama hızı üzerinde durulmuş ama yakınsadığı değere değinilmemiştir. Oysa, karesel ortalama hatanın ulaştığı bu *kalıcı hal değeri*, algoritmanın *hata başarımının* bir göstergesidir. Çünkü, bu değer ne kadar küçük oluyorsa, algoritma da istenen yanıtın kestirimi $\hat{d}(n)$ 'i istenen yanıt $d(n)$ 'e o kadar çok benzetebiliyor demektir. Örneğin; $M = 10$, $W = 2.9$ ve $\sigma_v^2 = 0.001$ için (Şekil 6.5 (a)) RLSL algoritması, 0.0000015 gibi sıfır sayılabilecek kadar küçük bir karesel ortalama hata ile kalıcı olarak çalışmaktadır. Dolayısıyla, bu koşullar altında üreteceği kestirim de, istenen yanıtla oldukça çok benzecektir.

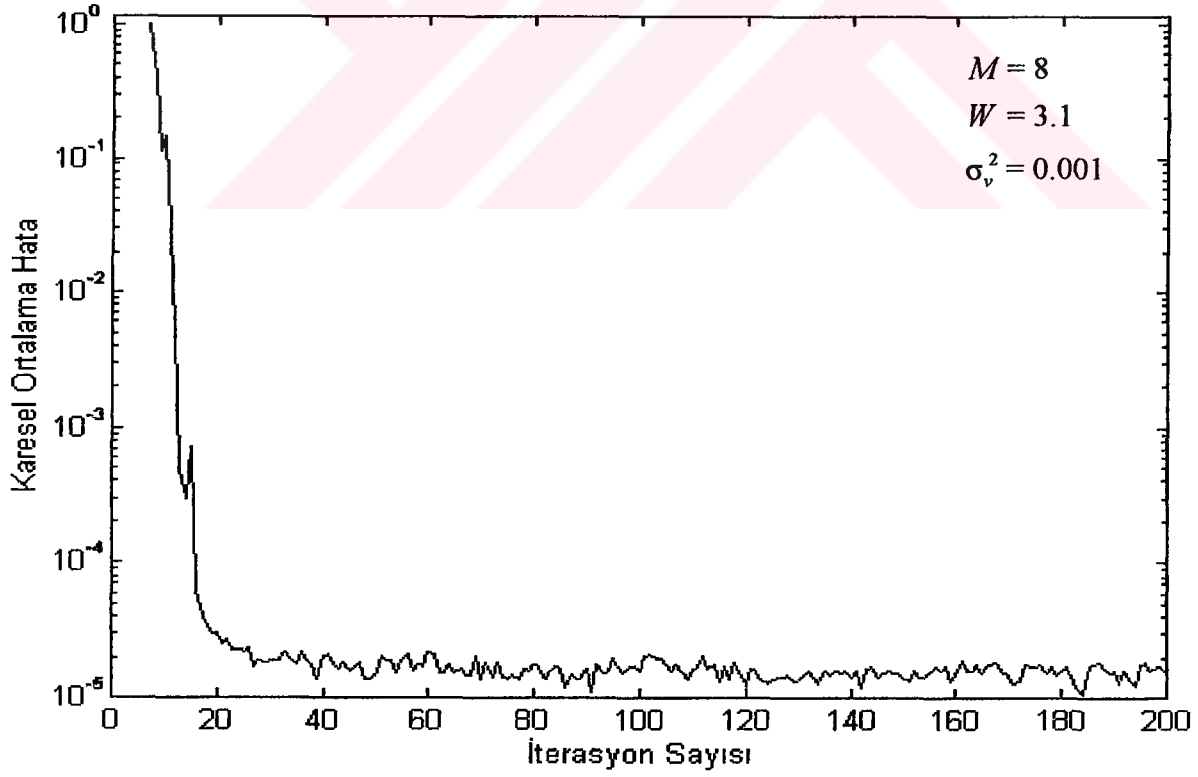
Şekil 6.8 (a), (b) ve (c) incelendiğinde, karesel ortalama hatanın kalıcı hal değerinin, gürültünün değişintisi ile birlikte azalmakta olduğu görülür. Kanalin yarattığı bozulma ve kafes filtre modül sayısı aynı olmasına rağmen, gürültüdeki artış algoritmanın hata başarımını oldukça belirgin bir şekilde kötüleştirmektedir. Şekil 6.9 (a), (b) ve (c)'de, aynı gürültü değişintisine karşı farklı M değerleriyle elde edilen karesel ortalama hatalar

karşılaştırılmaktadır. Şekil 6.9 (a)'dan, kafes filtre modül sayısı arttırıldıkça, algoritmanın hata başarımının da arttığı görülmektedir; ancak, Şekil 6.9 (b) ve (c)'den görüldüğü gibi, giderek artan gürültü değişintisi karşısında kafes filtre modül sayısını arttırmak da etkili olmamakta ve tüm eğriler aynı değere yakınsamaktadırlar. Buna göre, kafes filtre modül sayısını arttırmak, algoritmanın hata başarımını iyileştirmektedir; ama bunu yaparken, gürültü değişintisi de göz önünde bulundurulmalıdır.

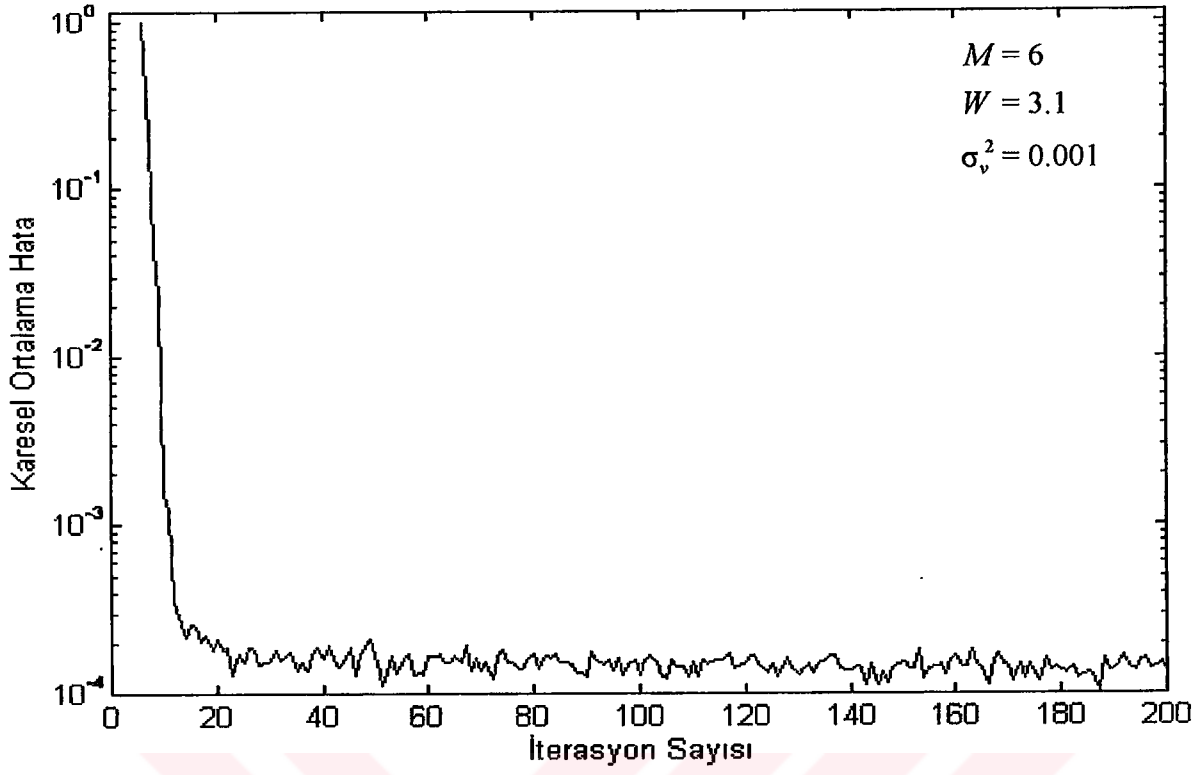




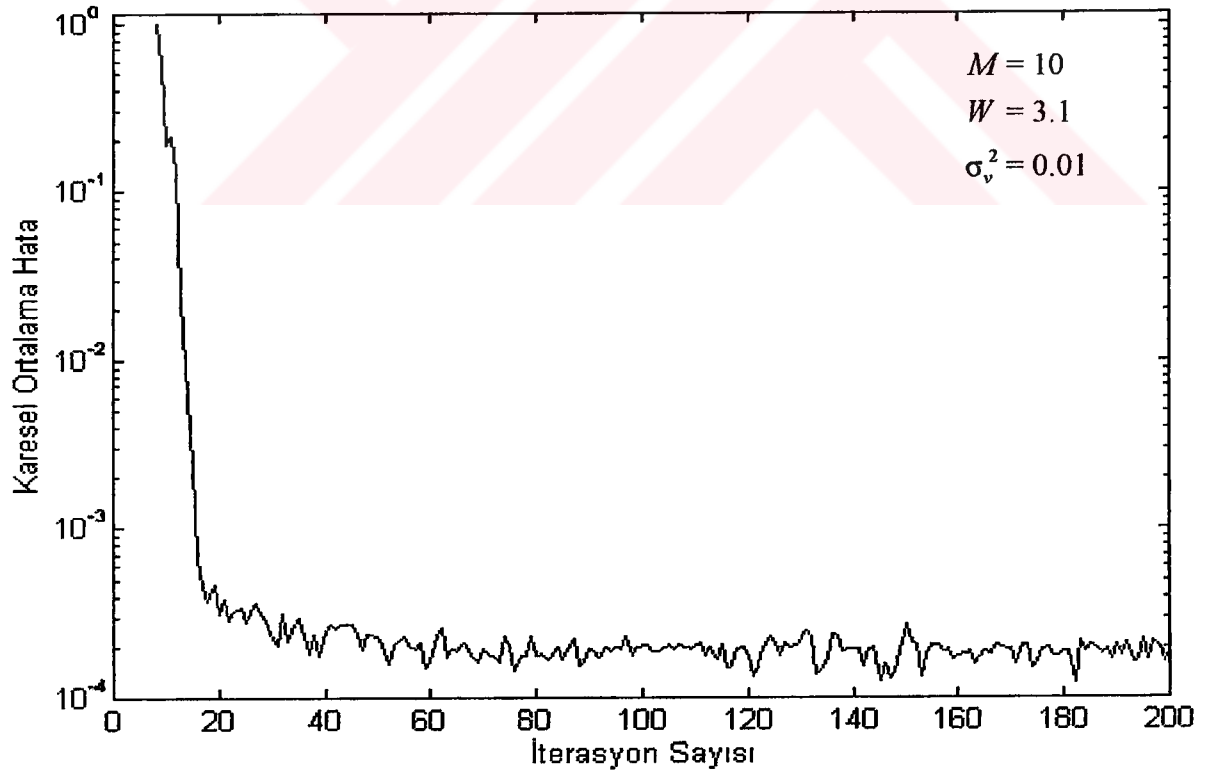
Şekil 6.10 (a)



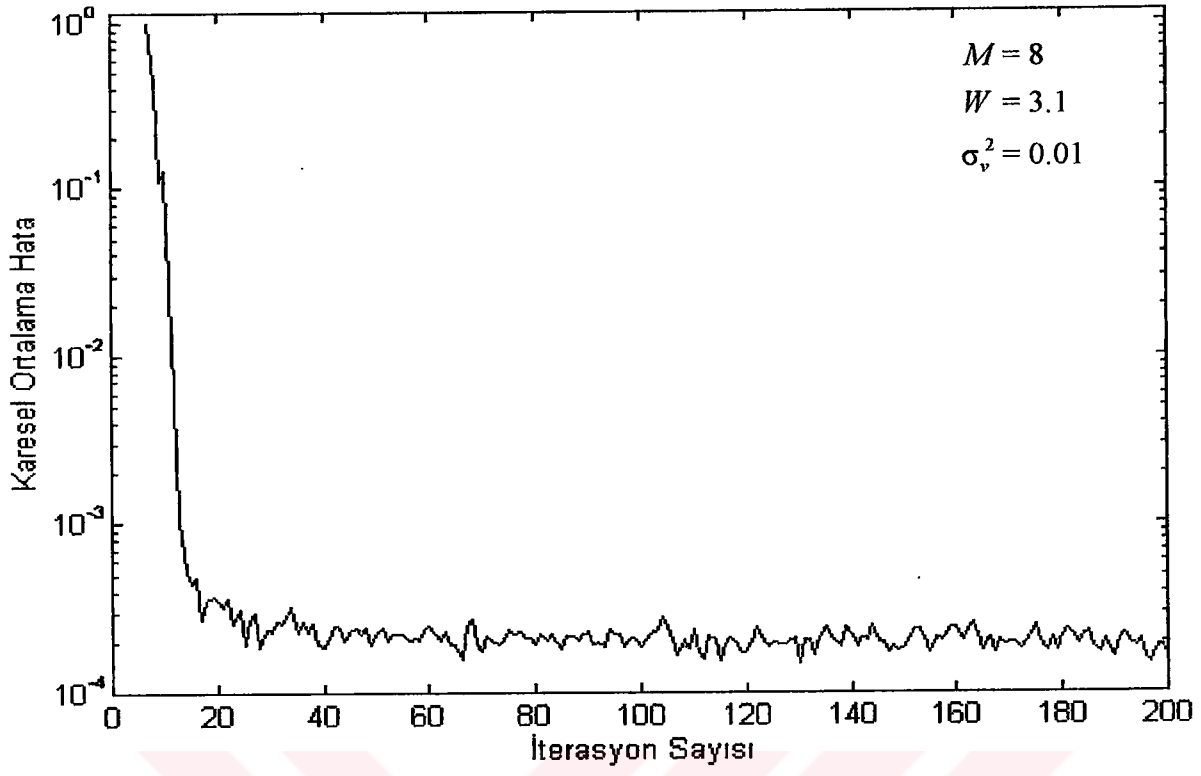
Şekil 6.10 (b)



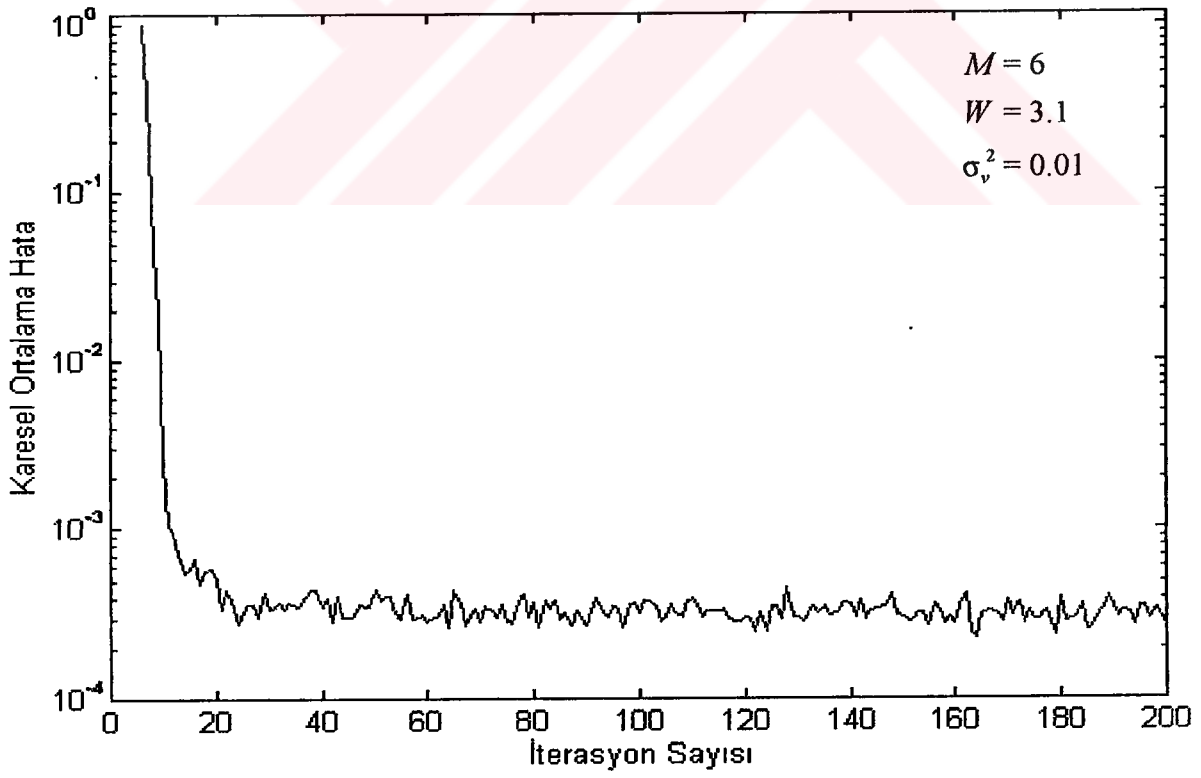
Şekil 6.10 (c)



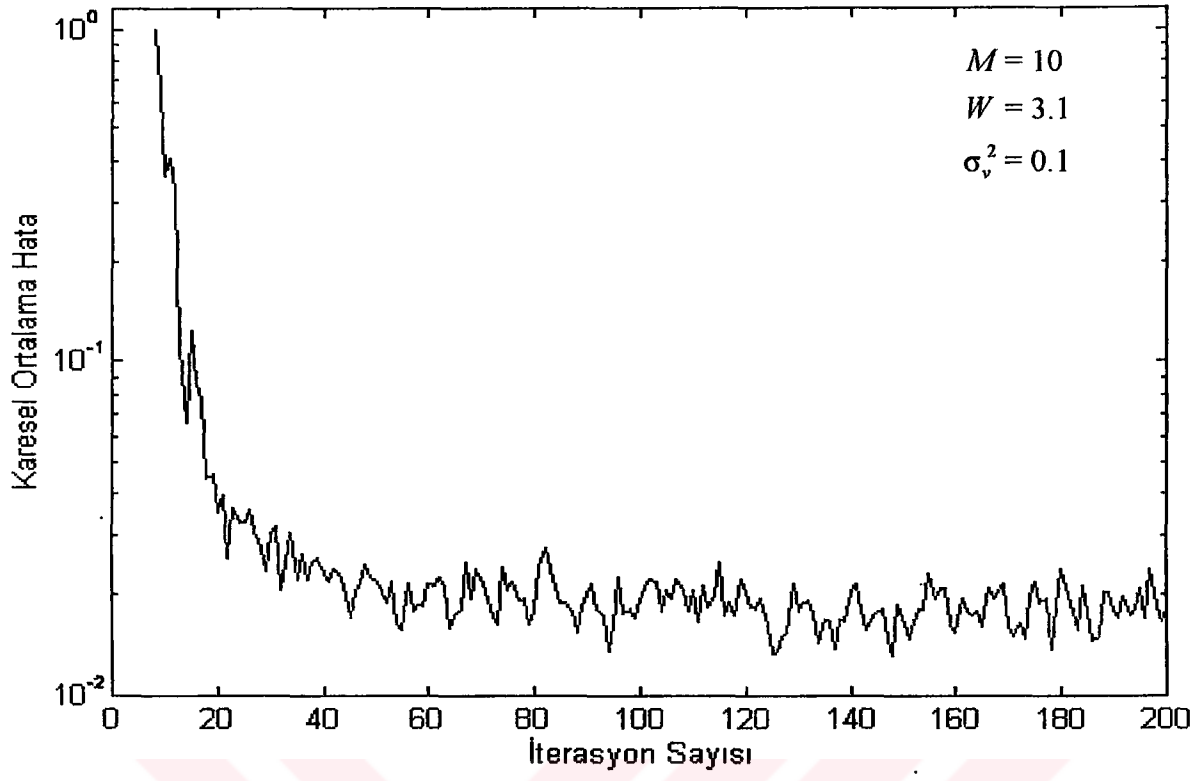
Şekil 6.11 (a)



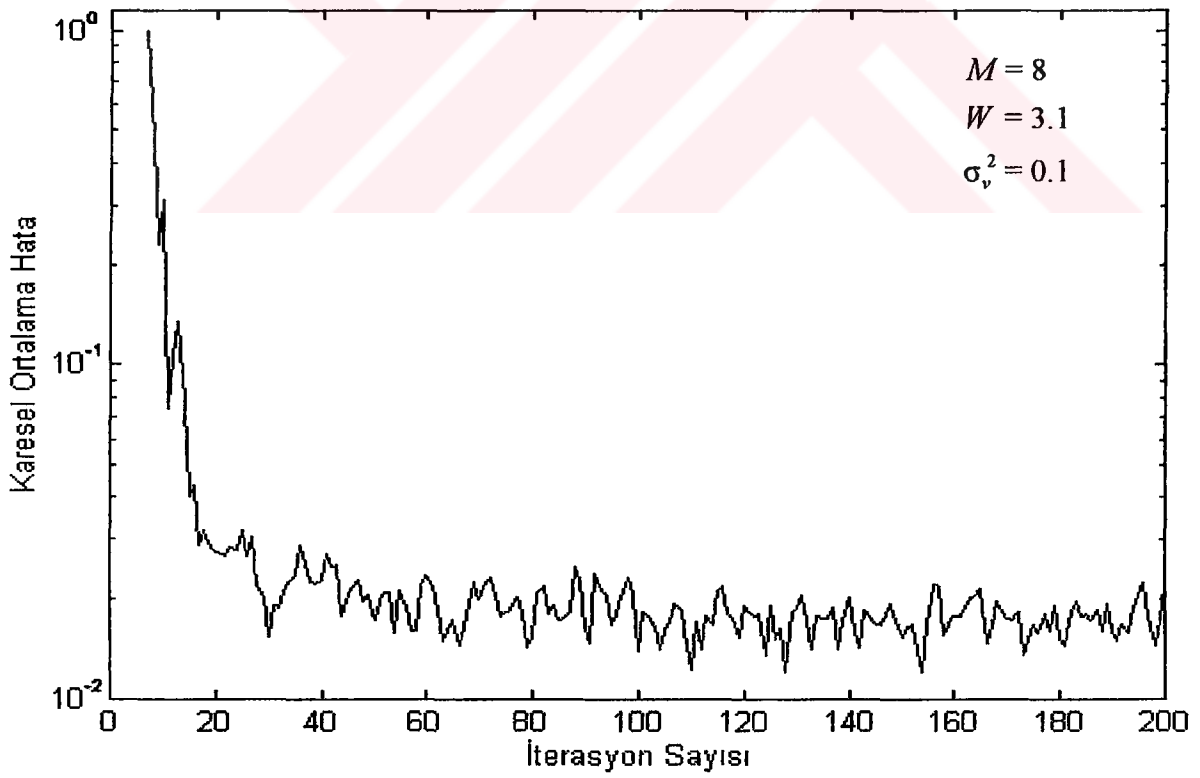
Şekil 6.11 (b)



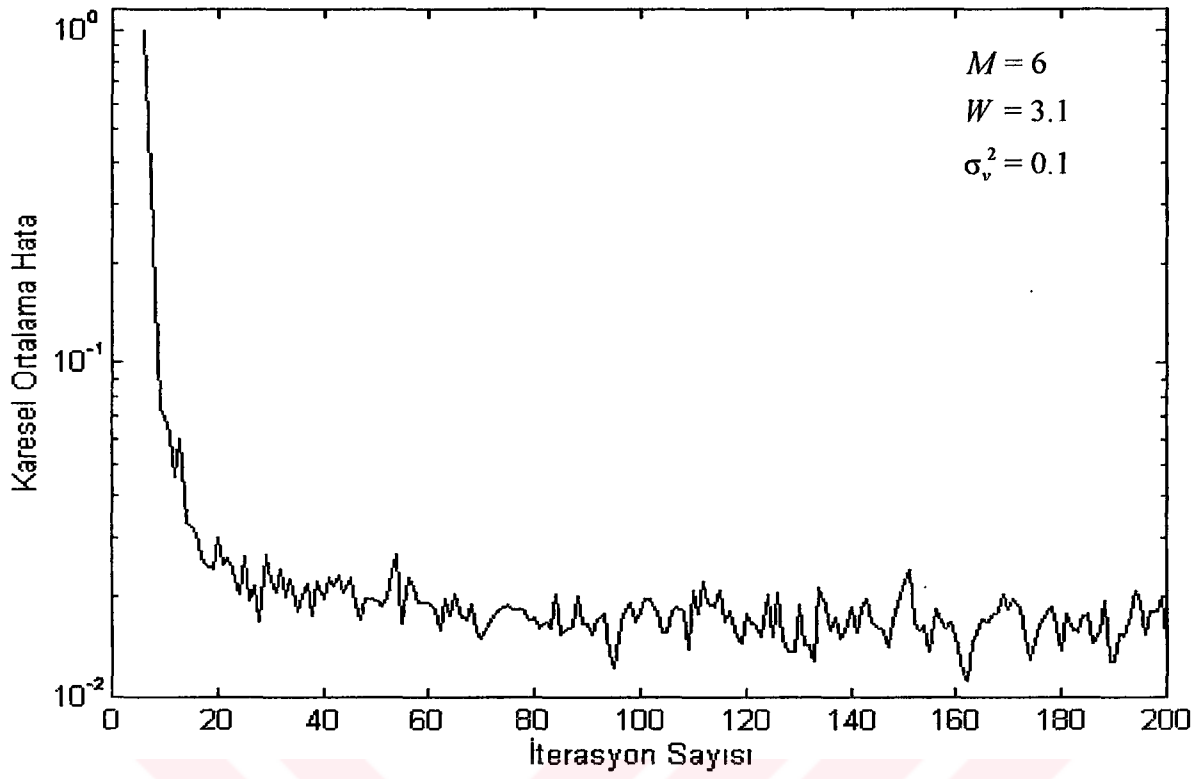
Şekil 6.11 (c)



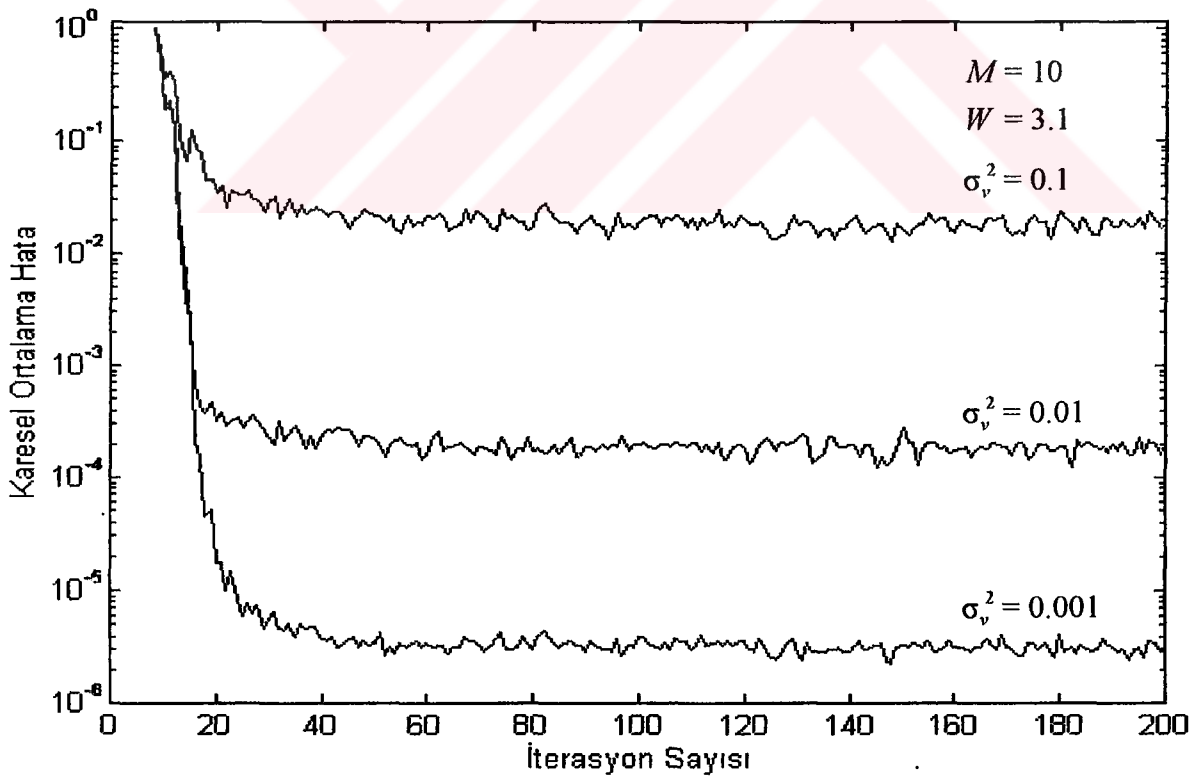
Şekil 6.12 (a)



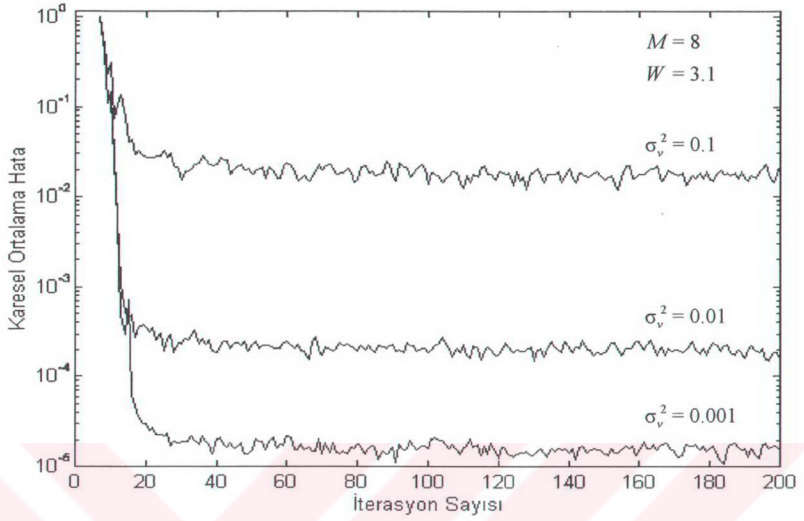
Şekil 6.12 (b)



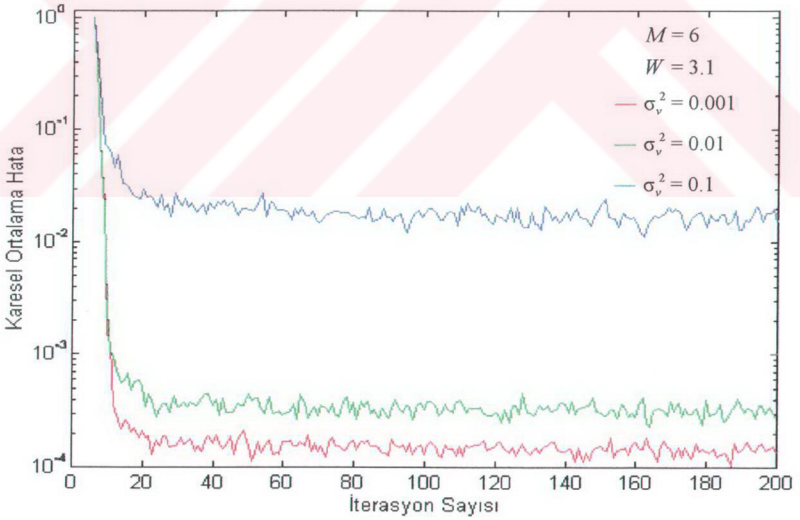
Şekil 6.12 (c)



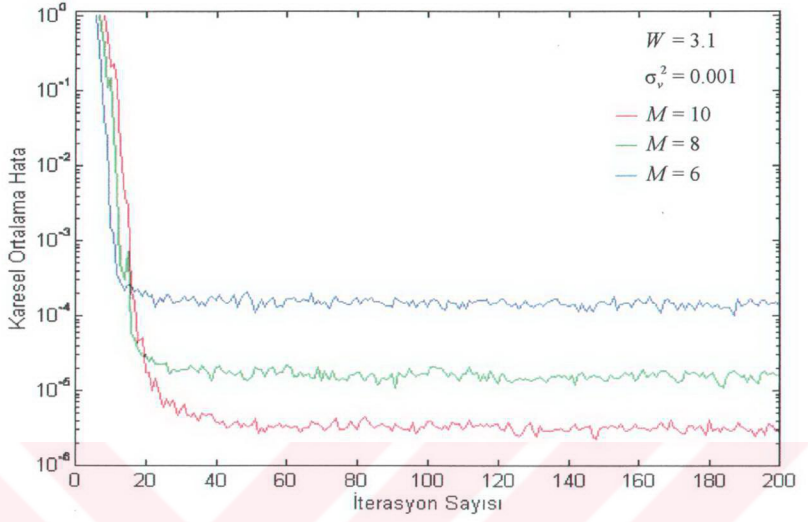
Şekil 6.13 (a)



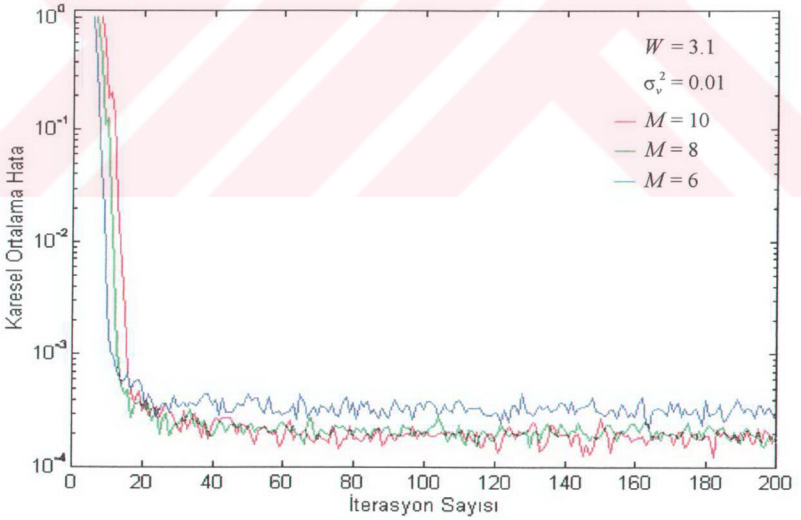
Şekil 6.13 (b)



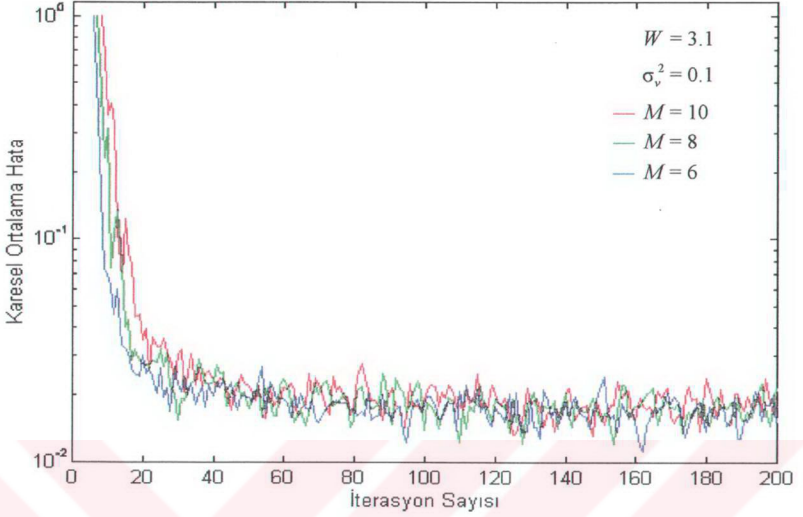
Şekil 6.13 (c)



Şekil 6.14 (a)



Şekil 6.14 (b)

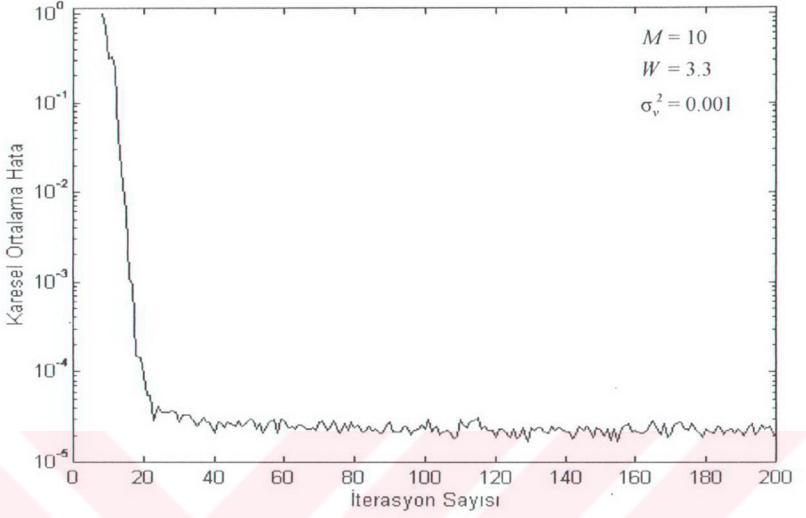


Şekil 6.14 (c)

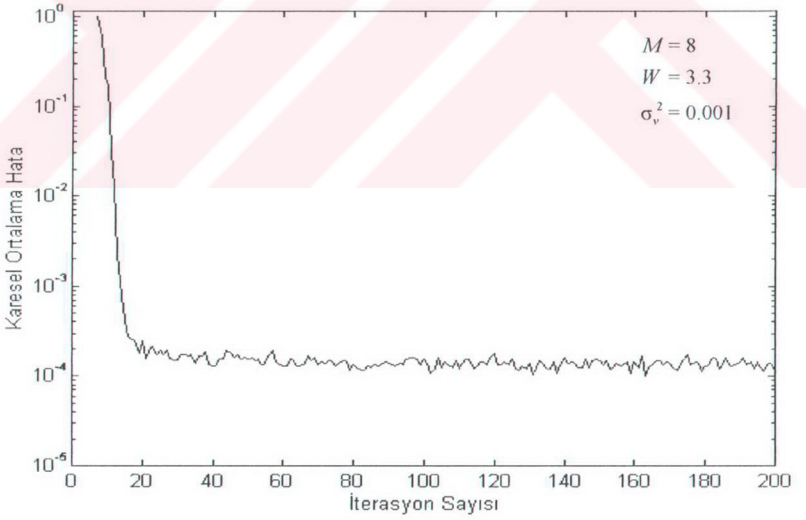
İlk kanal modeline ait öğrenme eğrileri için yaptığımız yorumlar, ikinci kanal modeline ait bu öğrenme eğrileri için de aynen geçerli olmaktadır. Özellikle, yakınsama hızı için yaptığımız yorumlar buradaki eğriler tarafından da doğrulanmaktadır. Buradaki eğriler ile ilk kanal modeline ait eğriler arasındaki tek fark, karesel ortalama hatanın kalıcı hal değerinden kaynaklanmaktadır. Bunun da tek sebebi olabilir: W parametresindeki artış. Daha önce de anlatıldığı gibi, W parametresi direkt olarak, yükseltilmiş kosinüs kanalının yarattığı bozulmanın genliğini ve dolaylı olarak da, dengeleyicinin tap girişlerine ait ilişki matrisinin özdeğer yayılımını kontrol etmektedir. Bu iki değer, W parametresinin artmasıyla birlikte artmakta ve azalmasıyla birlikte de azalmaktadır. Bu nedenle, W parametresinde meydana gelen bir artışın, RLSL algoritmasının hata başarımını düşürmesi beklenir.

Şekil 6.9 (a) ile Şekil 6.14 (a) ve Şekil 6.9 (b) ile Şekil 6.14 (b) birlikte incelendiğinde, M değerinin karesel ortalama hata üzerinde ne derece etkin bir rol oynadığı ortaya çıkmaktadır. Başlangıçta, kanal bozulması çok düşük olduğu için ($W = 2.9$), farklı M değerlerine rağmen birbirine oldukça yakın karesel ortalama hatalar elde edilmekte (Şekil 6.9 (a)) ve hatta, $\sigma_v^2 = 0.01$ koşulunda bile, tüm M değerleri için aynı karesel ortalama hata kalıcı hal

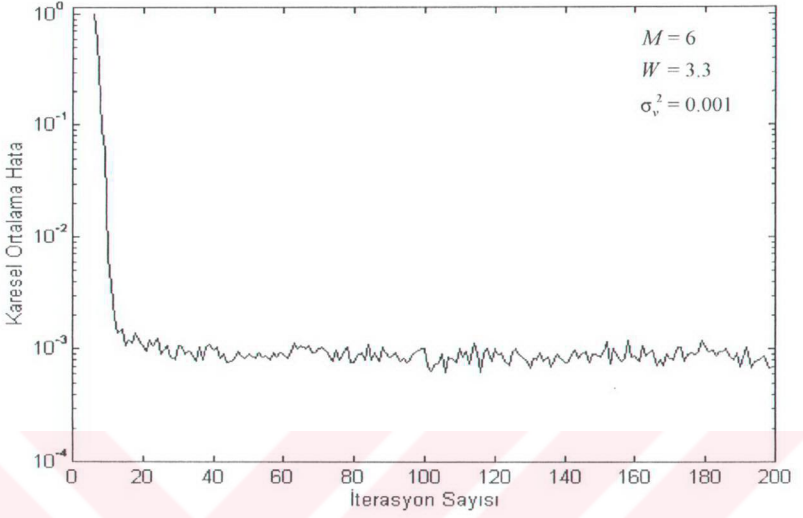
değerine erişilmektedir (Şekil 6.9 (b)). Bu durumda şöyle düşünülebilir: mevcut ortam koşulları altında aynı karesel ortalama hatayı elde ettiklerine göre, $M = 10$ değerli bir dengeleyici yerine $M = 8$ veya $M = 6$ değerli bir dengeleyici kullanmak daha ekonomiktir. İşte bu noktada, kanal bozulmasında bir artış olabileceği göz önünde bulundurulsa: M değeri büyük olan bir dengeleyici ile elde edilecek hata başarımının, M değeri küçük olan bir dengeleyici ile elde edilecek hata başarımından daha iyi olacağı ortaya çıkar (Şekil 6.14 (a) ve (b)). Öyleyse, artan W değerine karşılık alınabilecek en iyi önlem, M değerini arttırmaktır. Bu öneriye verilebilecek en güzel ispat da, Şekil 6.9 (b), 6.14 (b), 6.19 (b) ve 6.24 (b)'de yapılan karşılaştırmalardır. Öte yandan, kanal bozulması ne olursa olsun, $\hat{\sigma}_v^2 = 0.1$ koşulunda sonuç hiç değişmemekte ve her farklı M değeri için aynı karesel ortalama hataya ulaşılmaktadır. Aslında bu durum, M değerinde sağlanan artışın algoritmanın hata başarımını ancak belli bir seviyeye kadar yükseltebileceğini ve o seviyeden sonra, M değerini arttırmanın da fazla bir şey değiştirmeyeceğini gösterir.



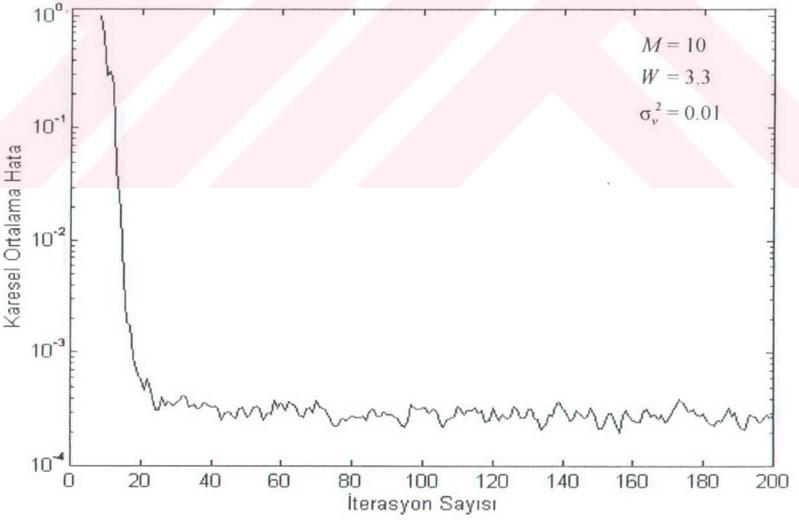
Şekil 6.15 (a)



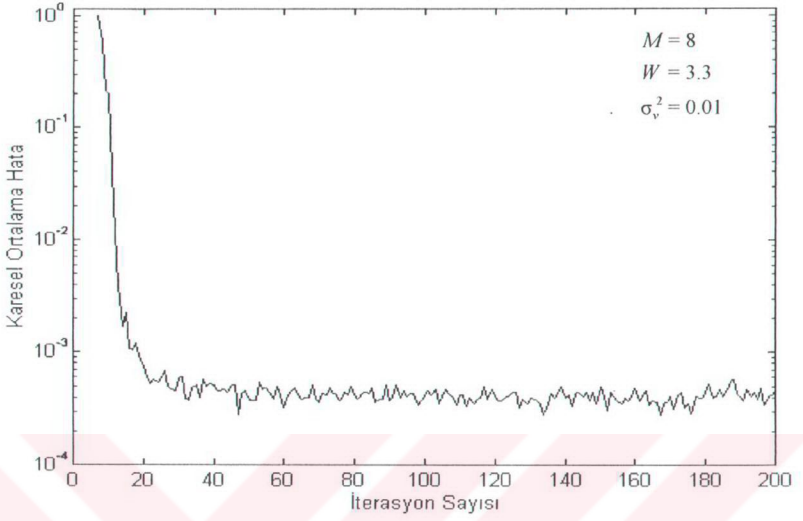
Şekil 6.15 (b)



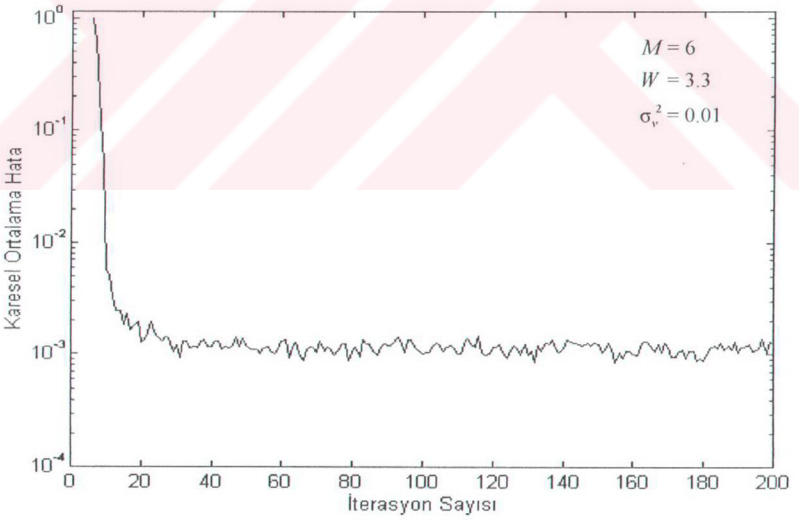
Şekil 6.15 (c)



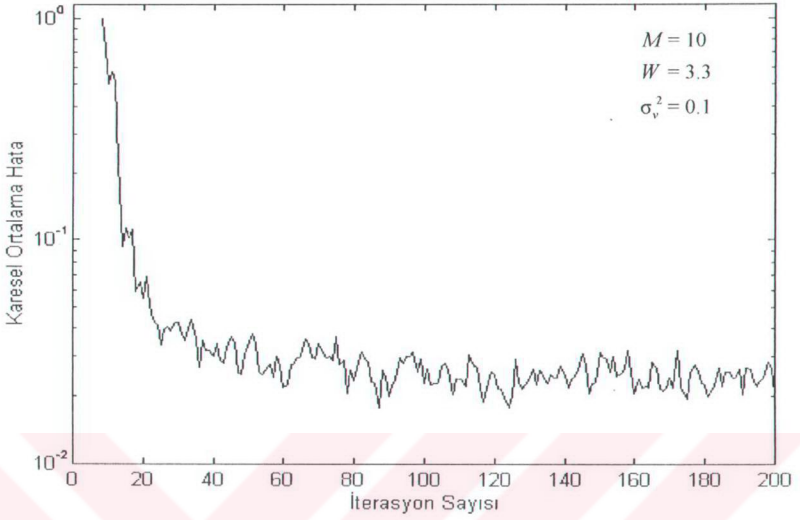
Şekil 6.16 (a)



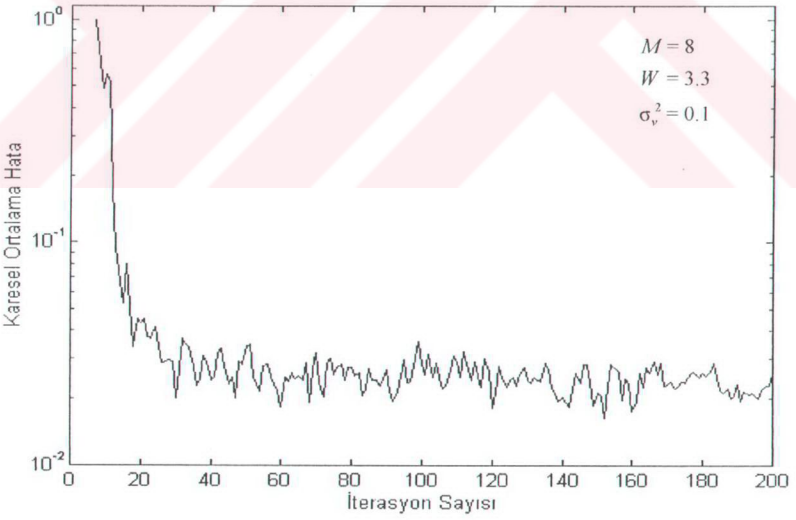
Şekil 6.16 (b)



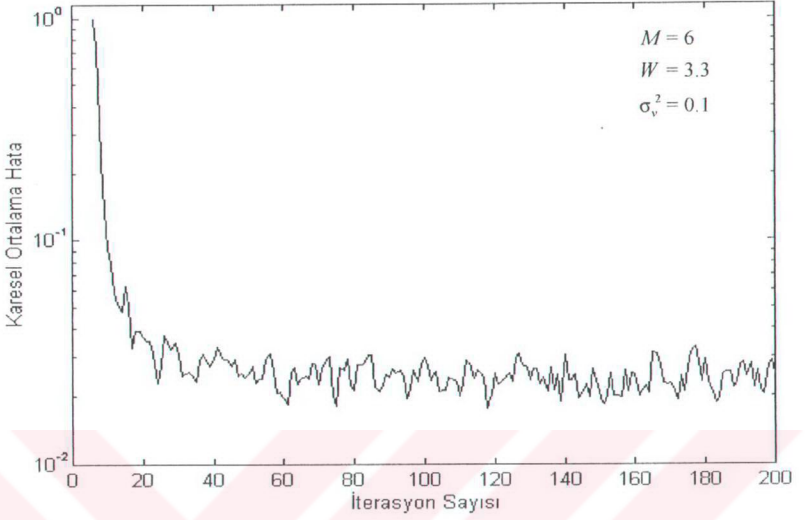
Şekil 6.16 (c)



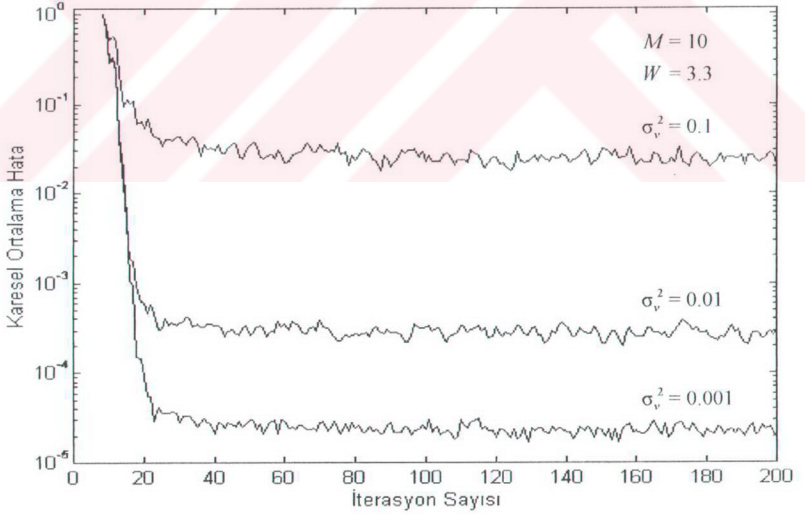
Şekil 6.17 (a)



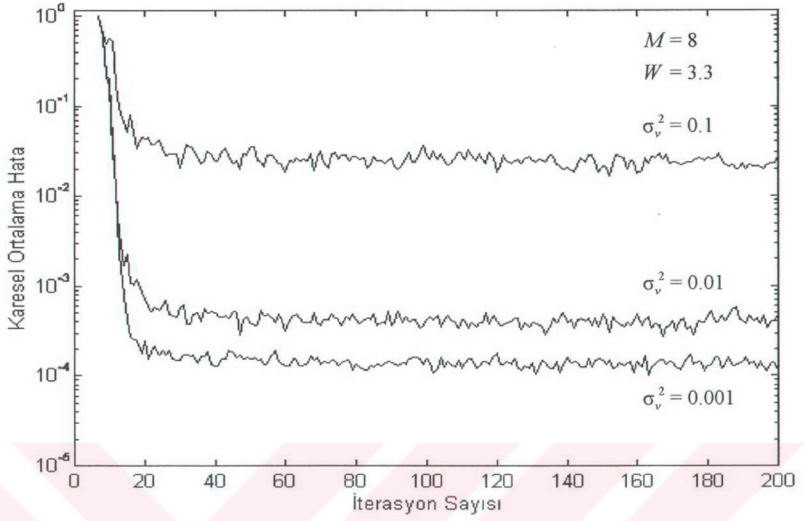
Şekil 6.17 (b)



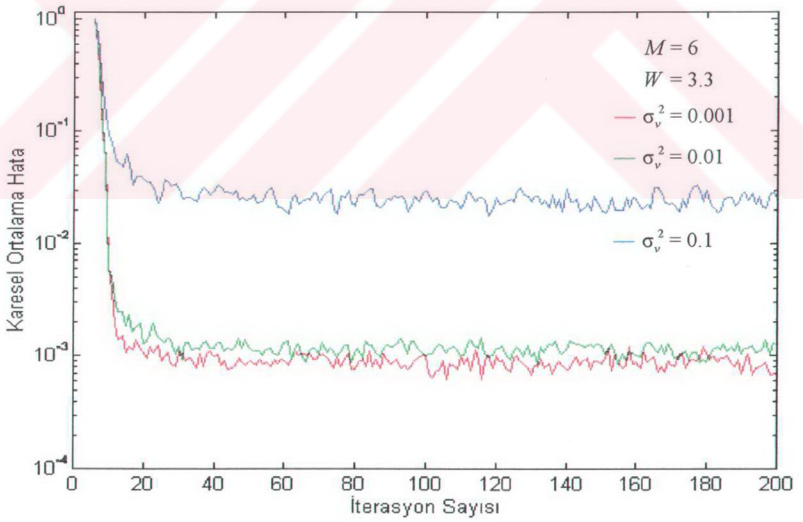
Şekil 6.17 (c)



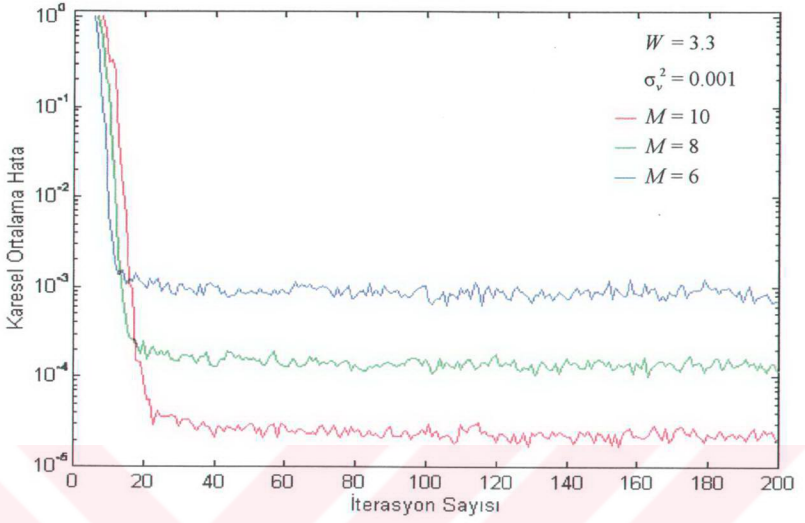
Şekil 6.18 (a)



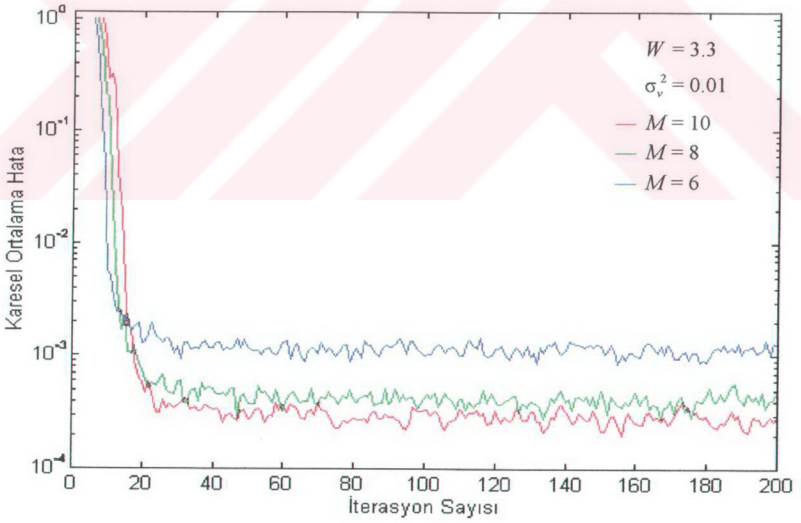
Şekil 6.18 (b)



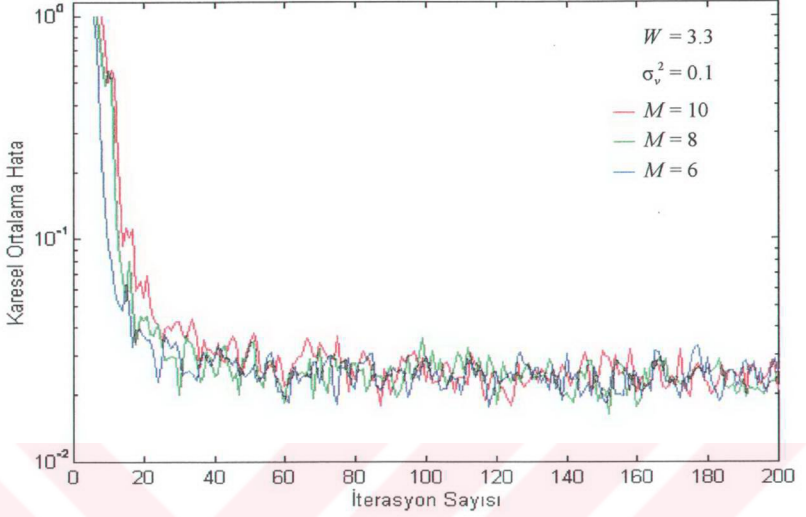
Şekil 6.18 (c)



Şekil 6.19 (a)

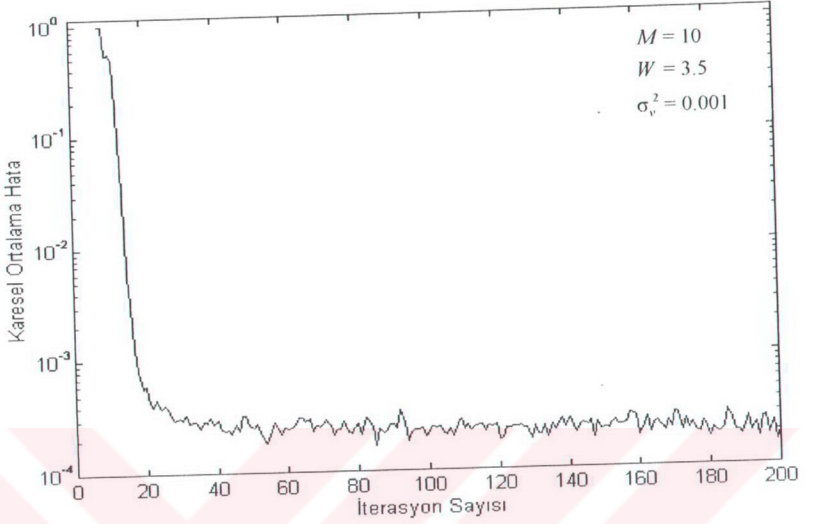


Şekil 6.19 (b)

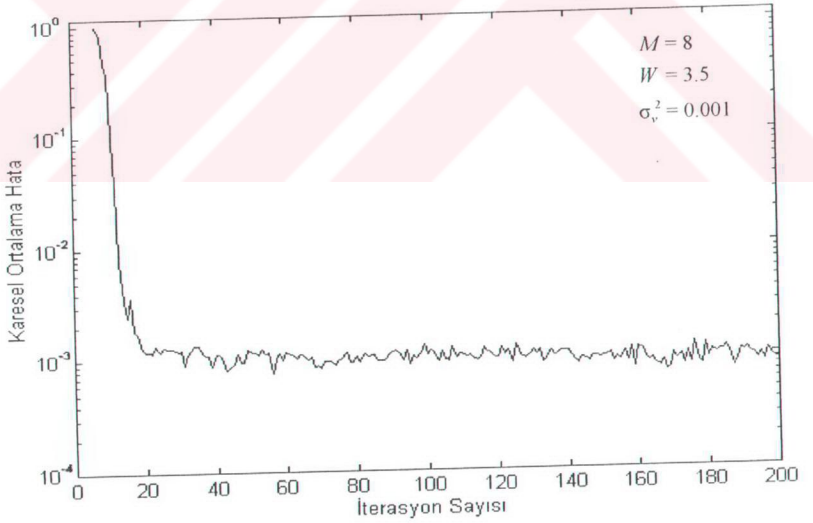


Şekil 6.19 (c)

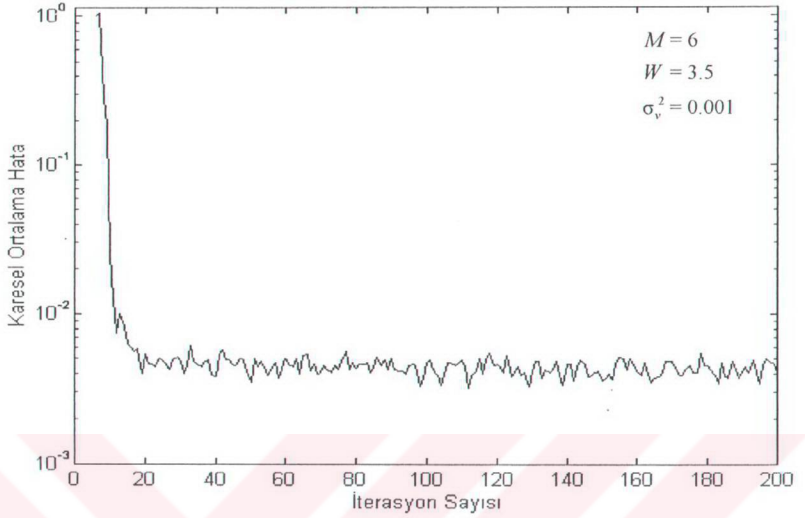
Birinci ve ikinci kanal modeline ait öğrenme eğrileri için yaptığımız yorumlar, üçüncü kanal modeline ait bu öğrenme eğrileri için de aynen geçerli olmaktadır. Özellikle, yakınsama hızı için yaptığımız yorumlar buradaki eğriler tarafından da doğrulanmaktadır. Tek fark yine, karesel ortalama hatanın kalıcı hal değerinin artmasından kaynaklanmaktadır. Algoritmanın hata başarımındaki bu düşüşün sebebi, bir önceki kanal modelinde de açıklandığı gibi, W parametresinin artmasıdır. Öte yandan, $\sigma_v^2 = 0.1$ koşulunda sonuç yine değişmemekte ve her farklı M değeri için aynı karesel ortalama hataya ulaşılmaktadır; ancak burada, kanalın yarattığı bozulmanın artması sonucu, $\sigma_v^2 = 0.1$ koşulunda her farklı M değeri için ulaşılan karesel ortalama hata kalıcı hal değeri 0.025'e yükselmiştir.



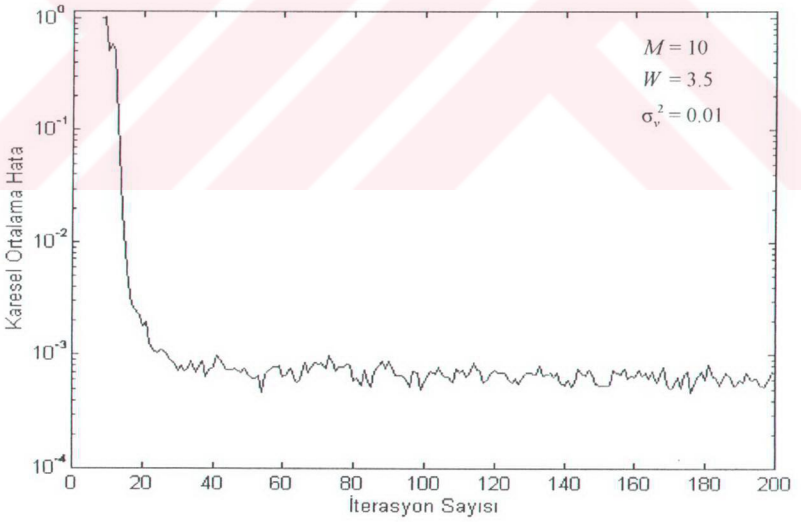
Şekil 6.20 (a)



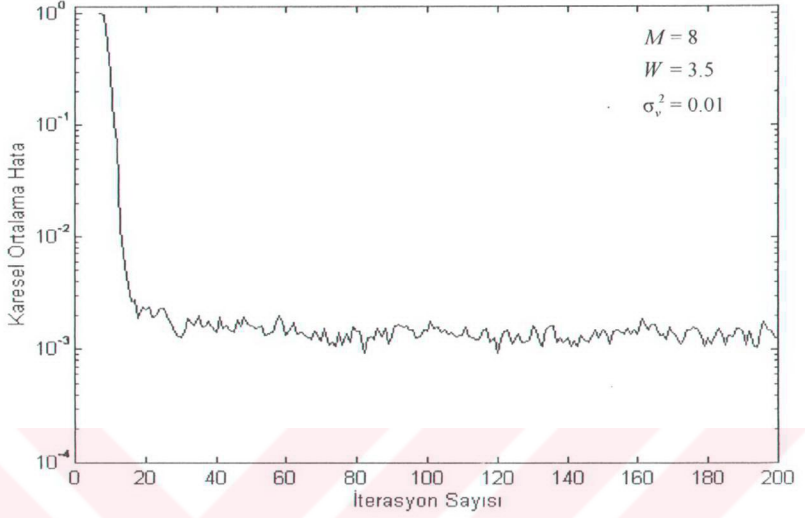
Şekil 6.20 (b)



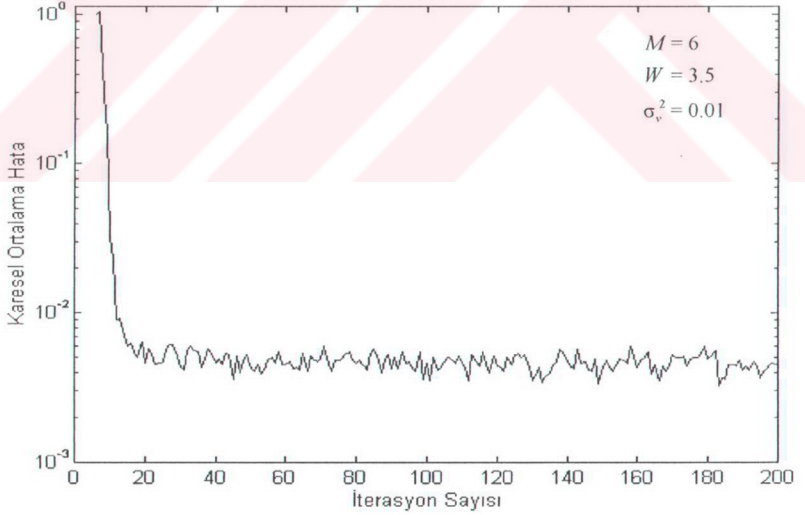
Şekil 6.20 (c)



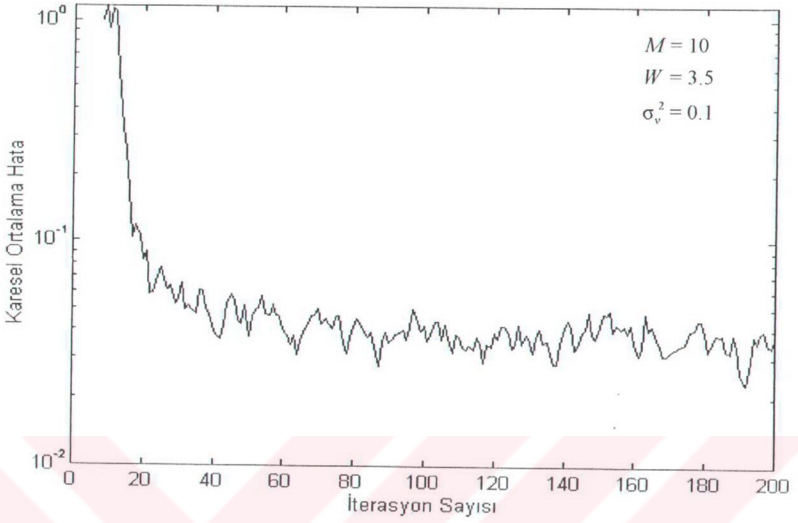
Şekil 6.21 (a)



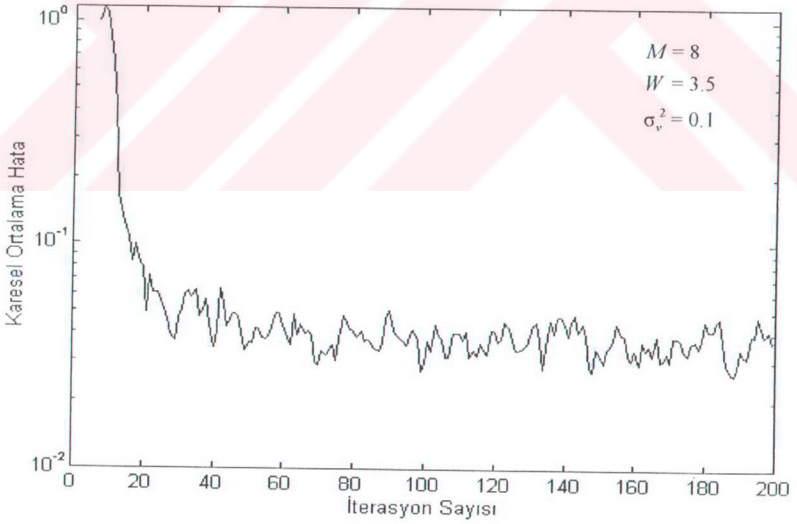
Şekil 6.21 (b)



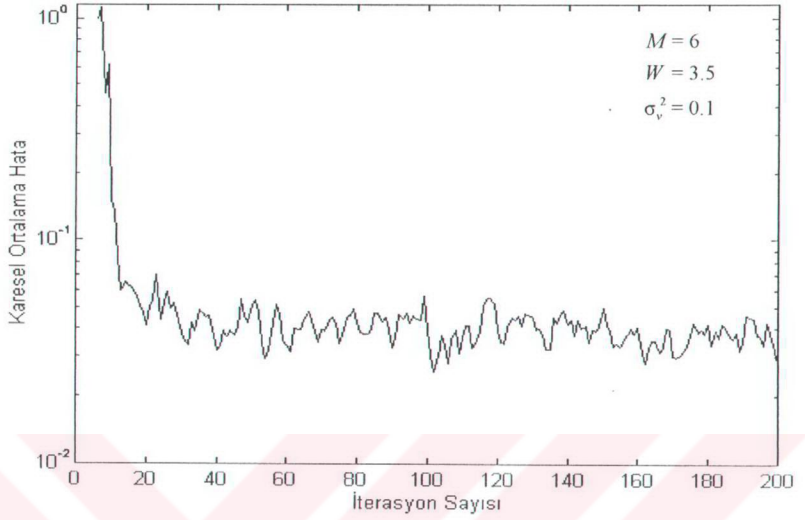
Şekil 6.21 (c)



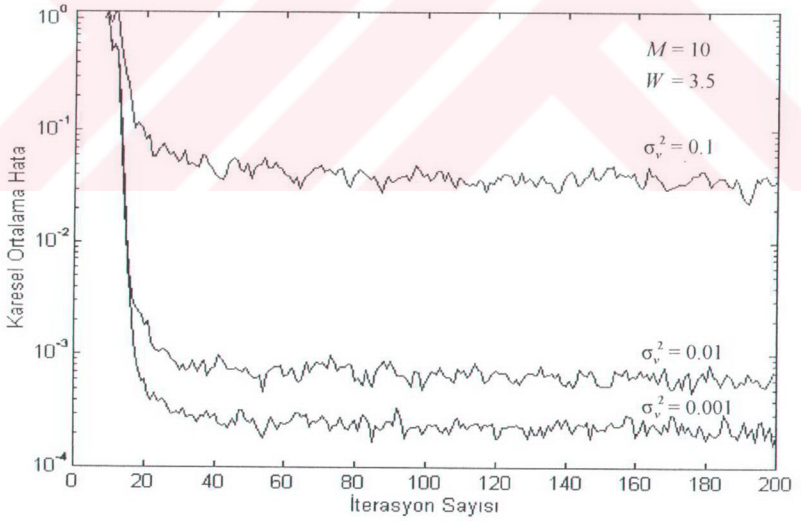
Şekil 6.22 (a)



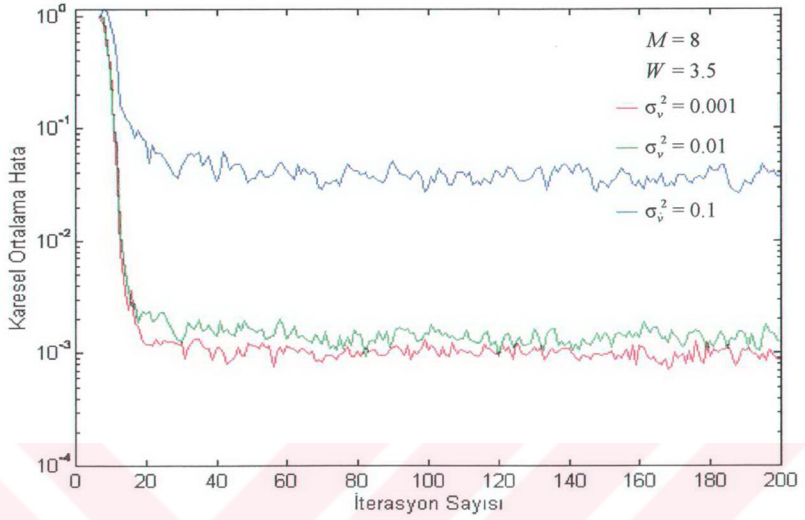
Şekil 6.22 (b)



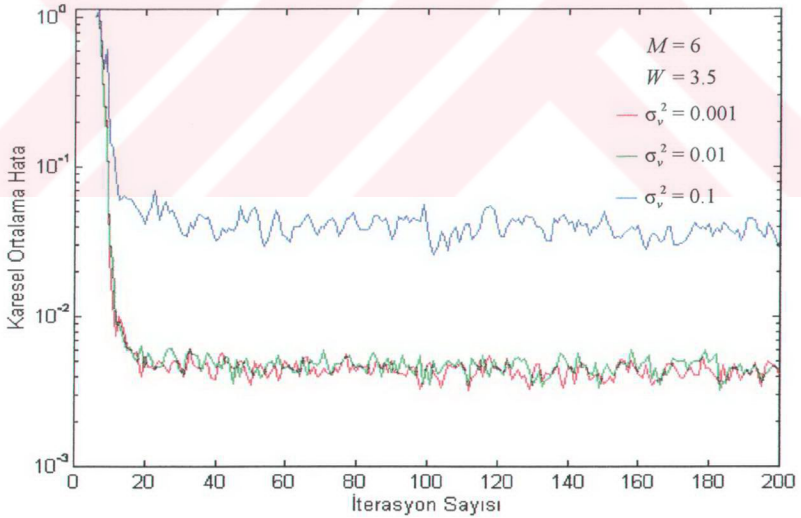
Şekil 6.22 (c)



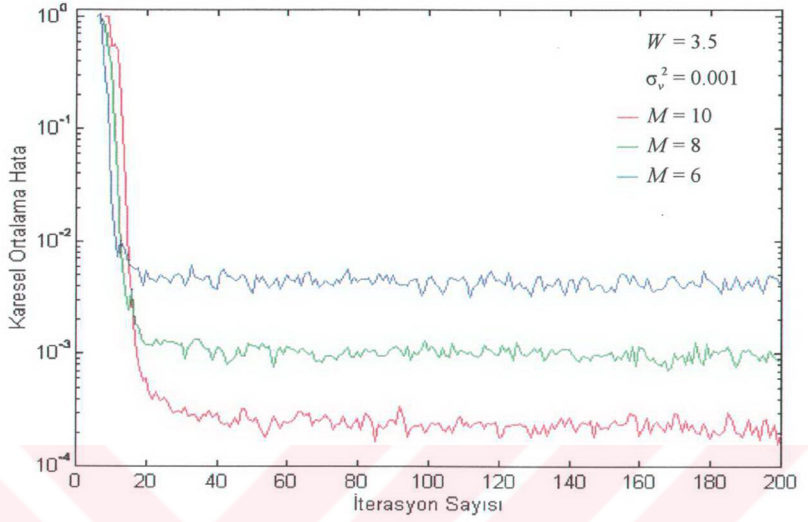
Şekil 6.23 (a)



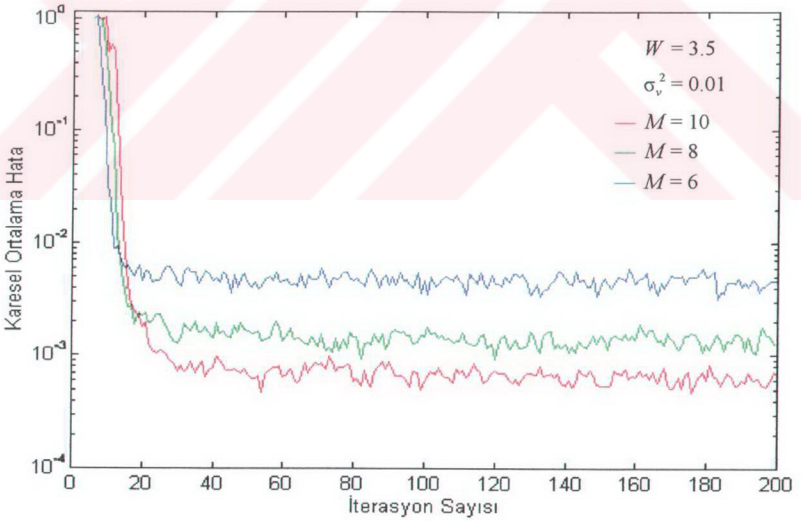
Şekil 6.23 (b)



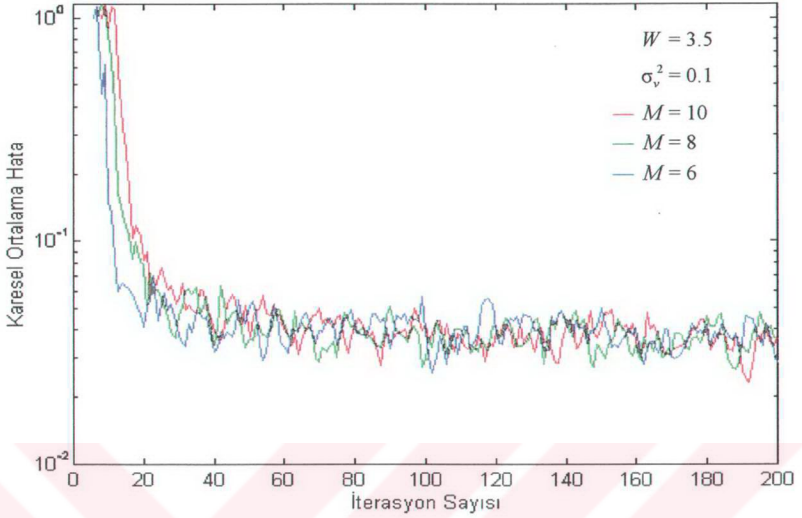
Şekil 6.23 (c)



Şekil 6.24 (a)

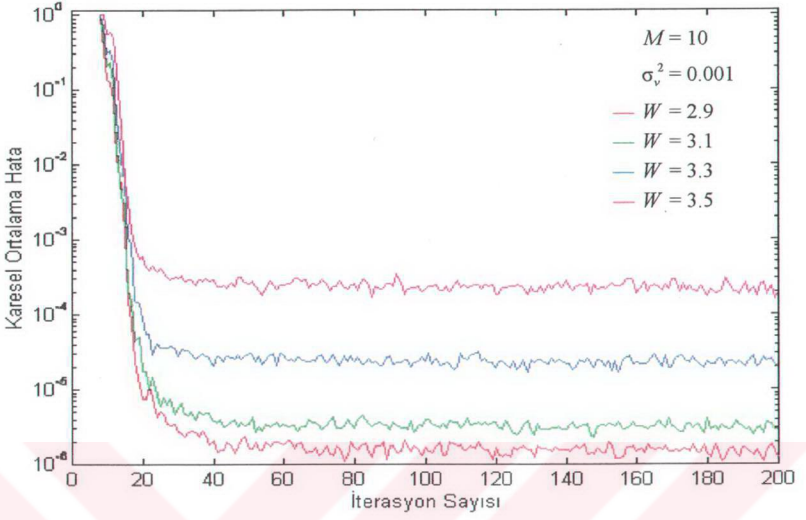


Şekil 6.24 (b)

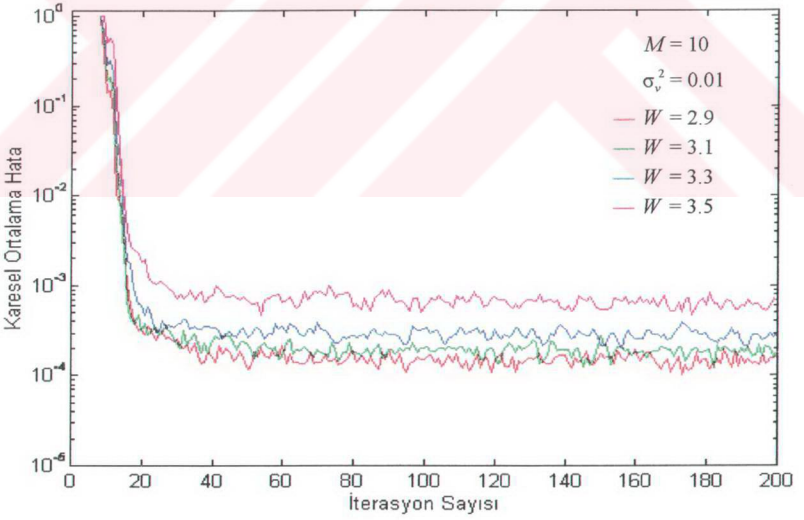


Şekil 6.24 (c)

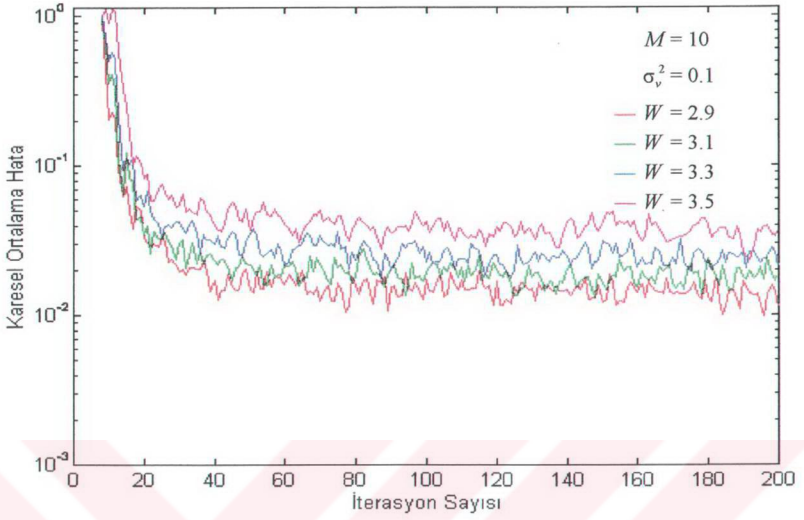
Birinci, ikinci ve üçüncü kanal modeline ait öğrenme eğrileri için yaptığımız yorumlar, dördüncü kanal modeline ait bu öğrenme eğrileri için de aynen geçerli olmaktadır. Varolan fark yine, algoritmanın hata başarımındaki düşüştür. Özellikle, $\sigma_v^2 = 0.1$ için elde edilen eğrilerde karesel ortalama hatanın kalıcı hal değeri (0.04) iyice 0.1'e yaklaşmıştır.



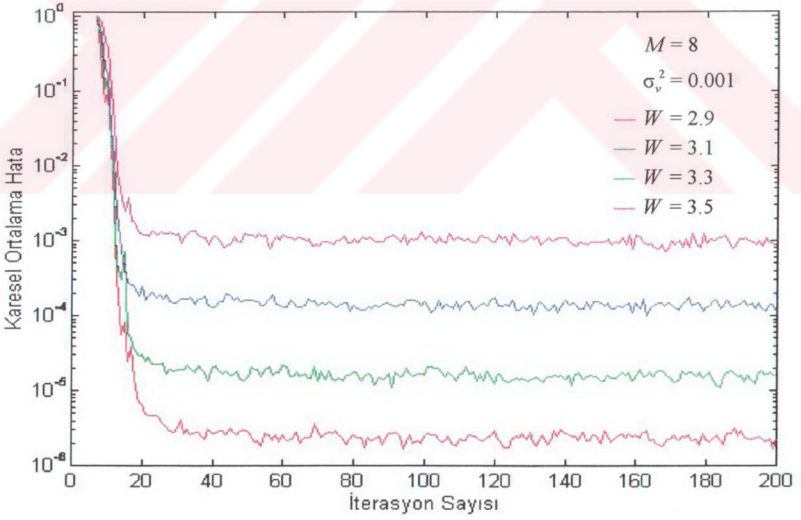
Şekil 6.25 (a)



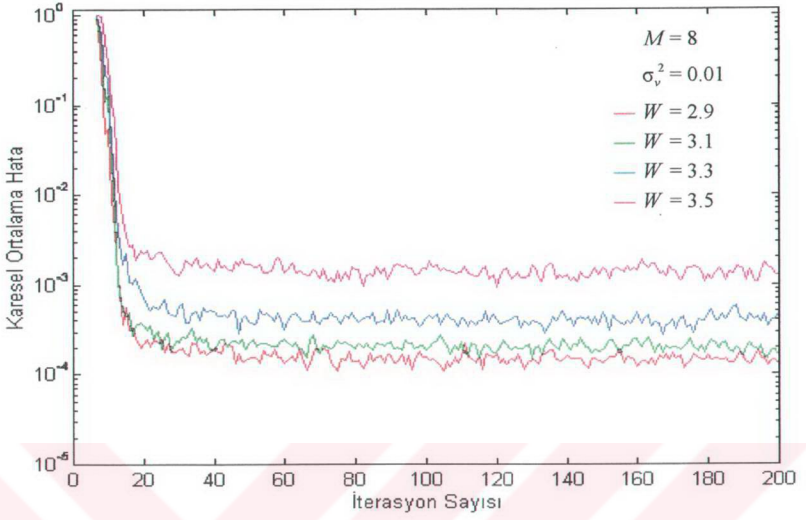
Şekil 6.25 (b)



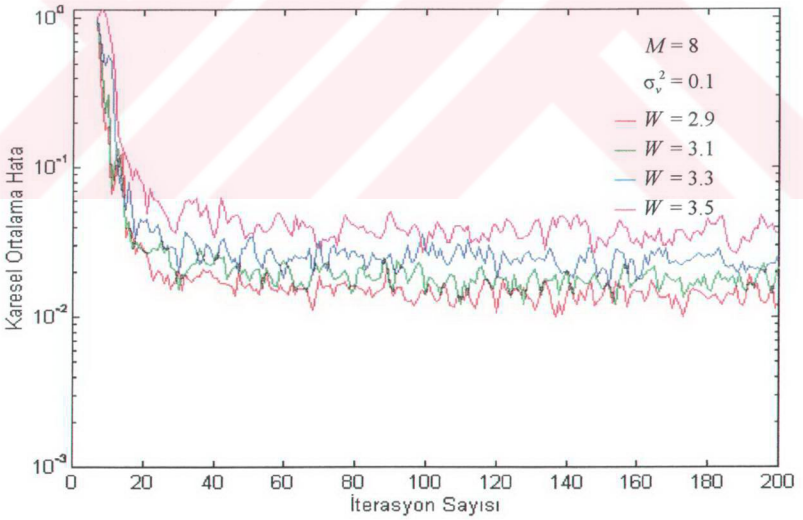
Şekil 6.25 (c)



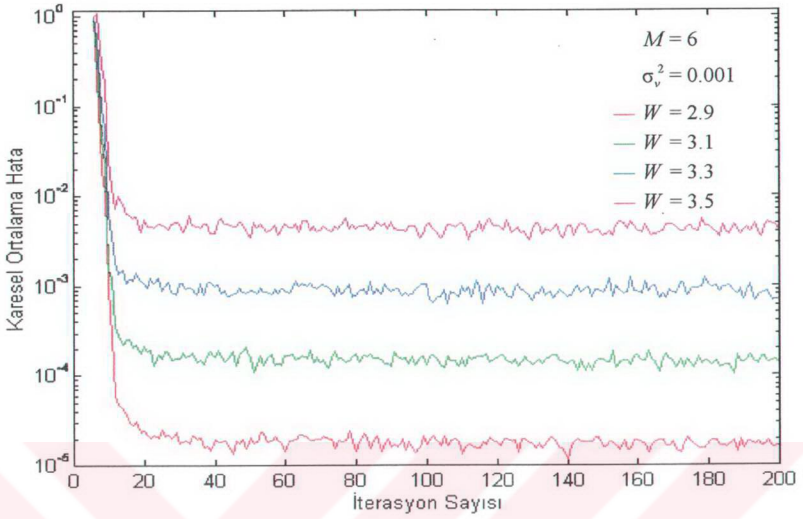
Şekil 6.26 (a)



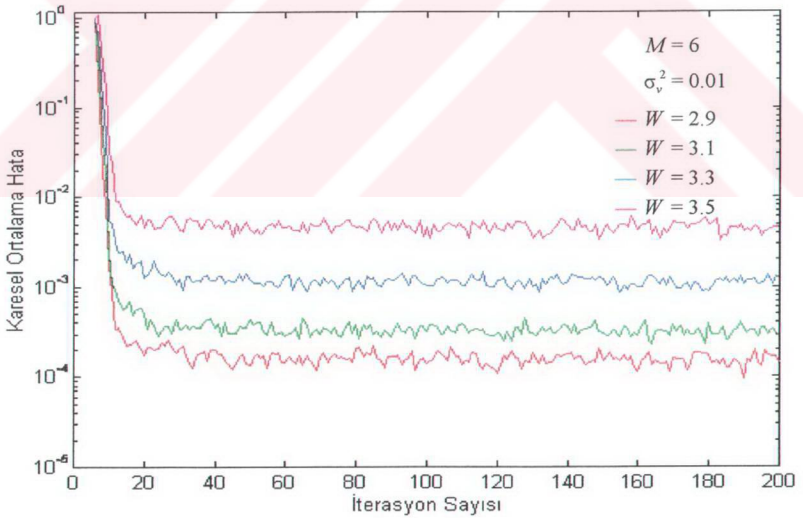
Şekil 6.26 (b)



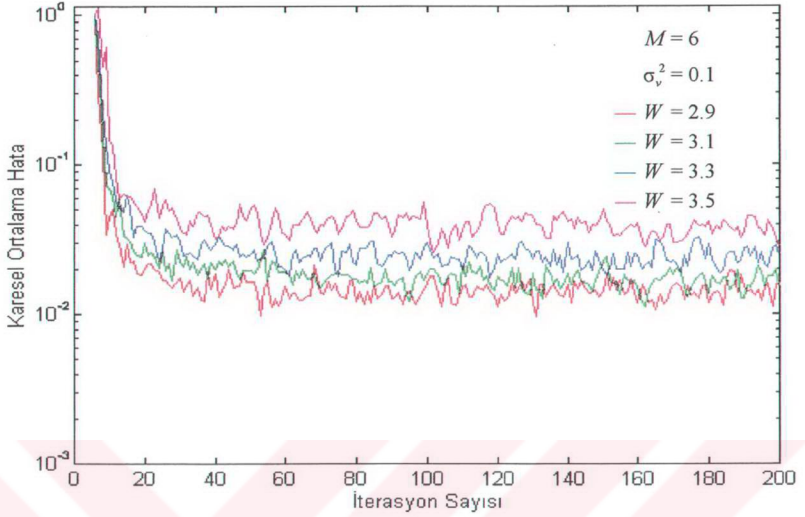
Şekil 6.26 (c)



Şekil 6.27 (a)

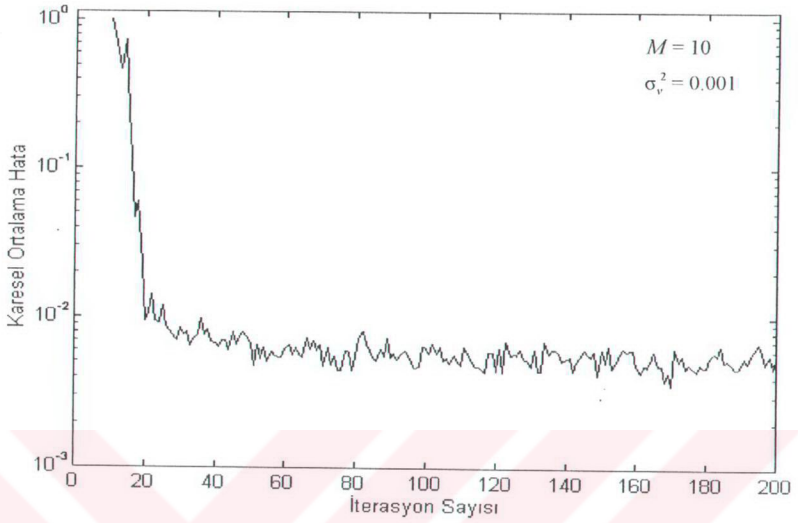


Şekil 6.27 (b)

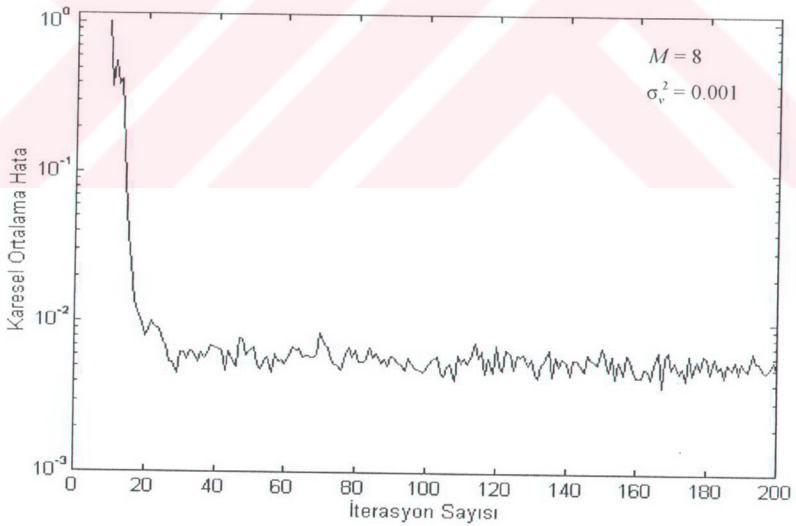


Şekil 6.27 (c)

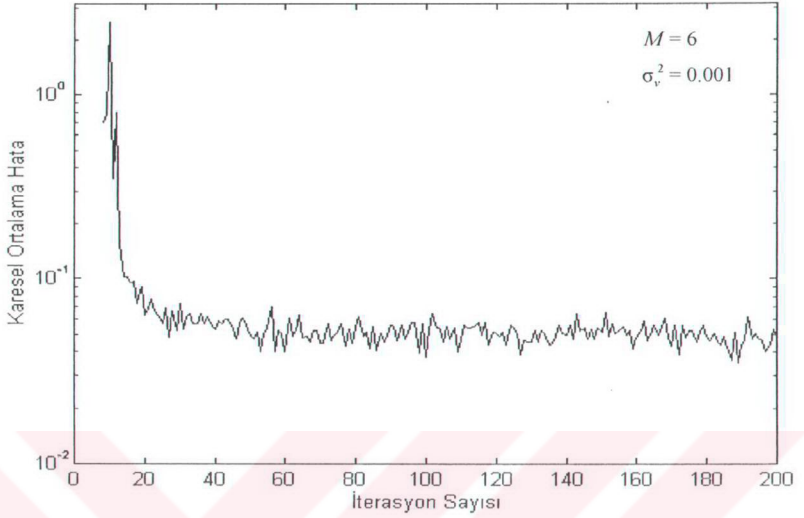
W parametresindeki değişimin karesel ortalama hata üzerindeki etkisini incelemek amacıyla, ilk dört kanal modeline ait öğrenme eğrilerinin tamamı burada, M ve σ_v^2 değerleri sabit alınarak karşılaştırılmışlardır. Şekillerden de görüldüğü gibi, RLSL algoritmasının yakınsama hızı, W parametresindeki (özdeğer yayılımındaki) değişimlere karşı duyarsızdır. W değerindeki değişime rağmen, her farklı M ve σ_v^2 sabit ikilisi için elde edilen dörtlü eğrilerdeki her dört eğrinin yakınsama süresi birbiriyle aynı olmaktadır. Örneğin: $M = 10$ ve $\sigma_v^2 = 0.001$ sabit ikilisi için elde edilen Şekil 6.25 (a)'daki her dört eğrinin yakınsama süresi birbiriyle aynı olurken, $M = 6$ ve $\sigma_v^2 = 0.001$ sabit ikilisi için elde edilen Şekil 6.27 (a)'daki dört eğrinin yakınsama süresinden farklıdır. Bu durumda, yakınsama hızını W parametresindeki değişimin değil, M ve σ_v^2 parametrelerindeki değişimin etkilediği anlaşılır. RLSL algoritması, W parametresinden tek bir yönde etkilenmektedir: karesel ortalama hatanın ulaştığı kalıcı hal değeri. Çünkü W parametresindeki artış, M veya σ_v^2 değeri ne olursa olsun, karesel ortalama hatanın kalıcı hal değerini arttırmaktadır. Öyleyse, RLSL algoritmasının hata başarımı, W parametresindeki (özdeğer yayılımındaki) değişimlere karşı duyarlıdır ve W değeri arttıkça azalmakta, azaldıkça da artmaktadır.



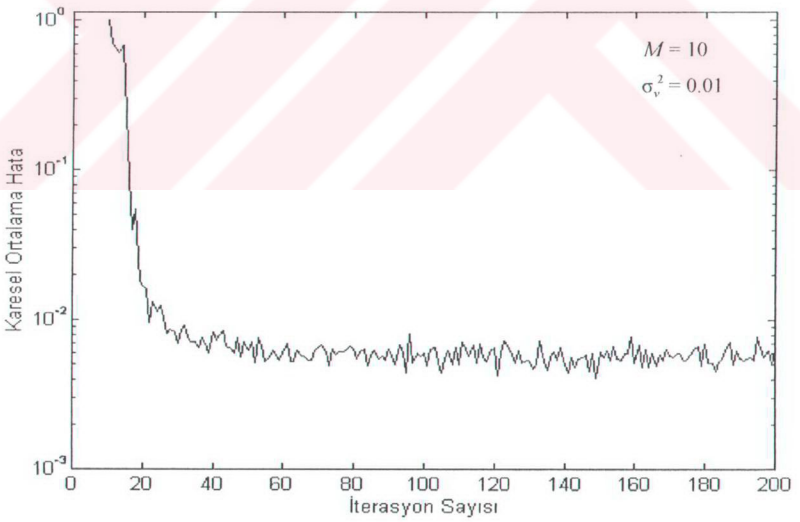
Şekil 6.28 (a)



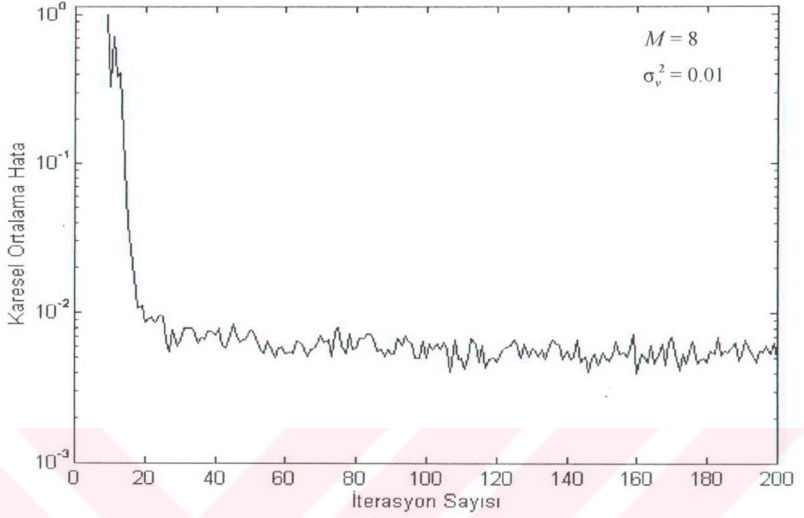
Şekil 6.28 (b)



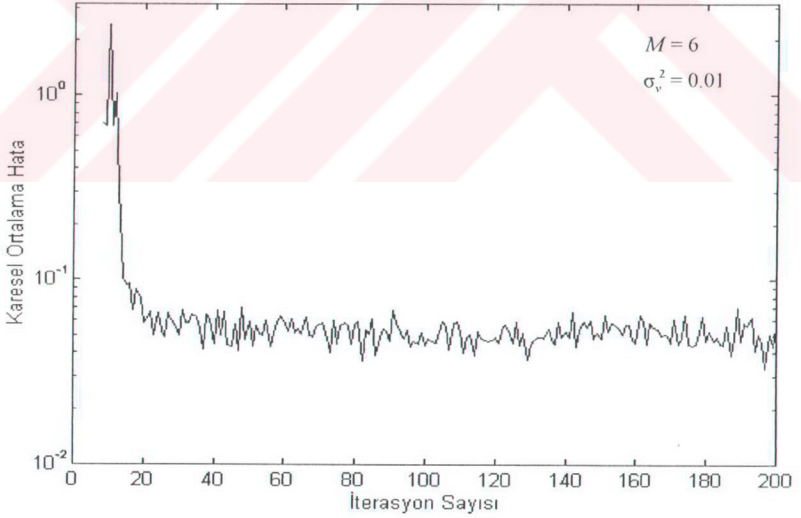
Şekil 6.28 (c)



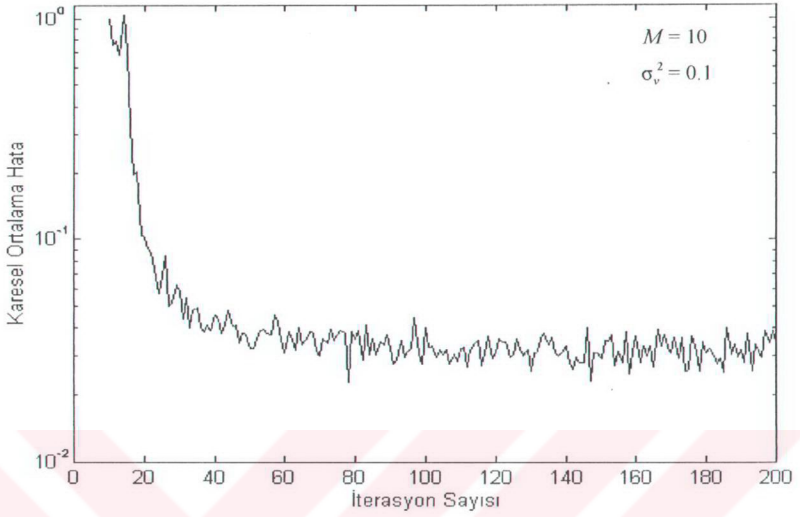
Şekil 6.29 (a)



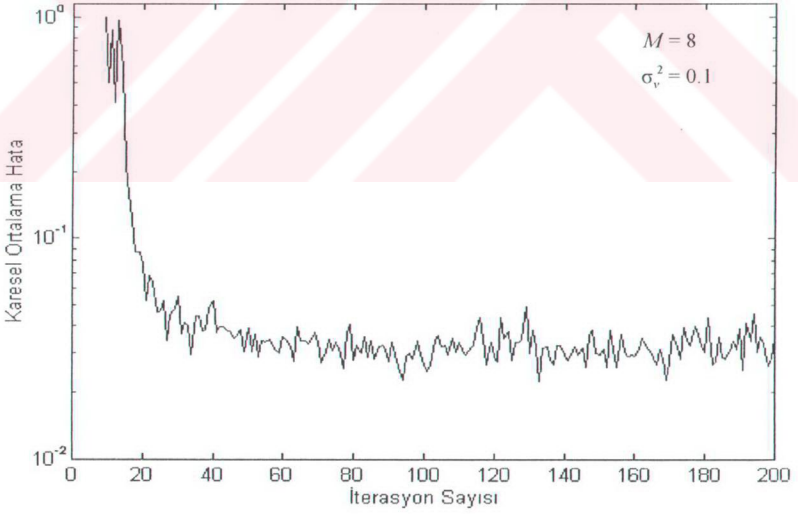
Şekil 6.29 (b)



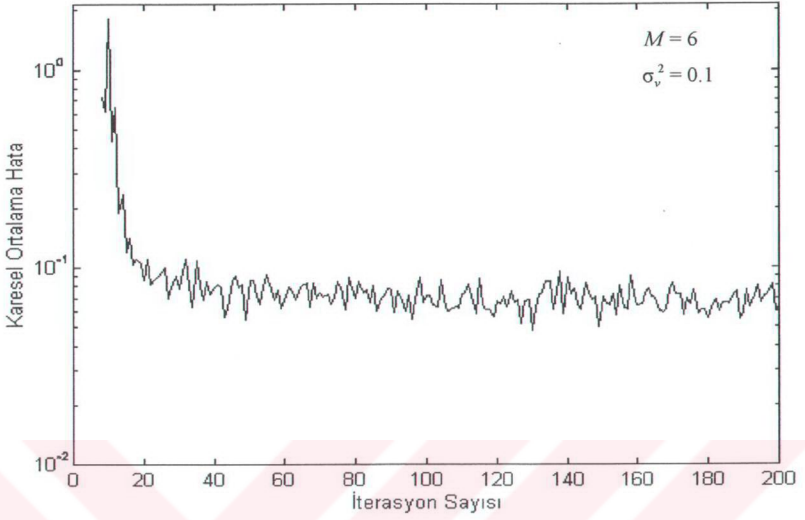
Şekil 6.29 (c)



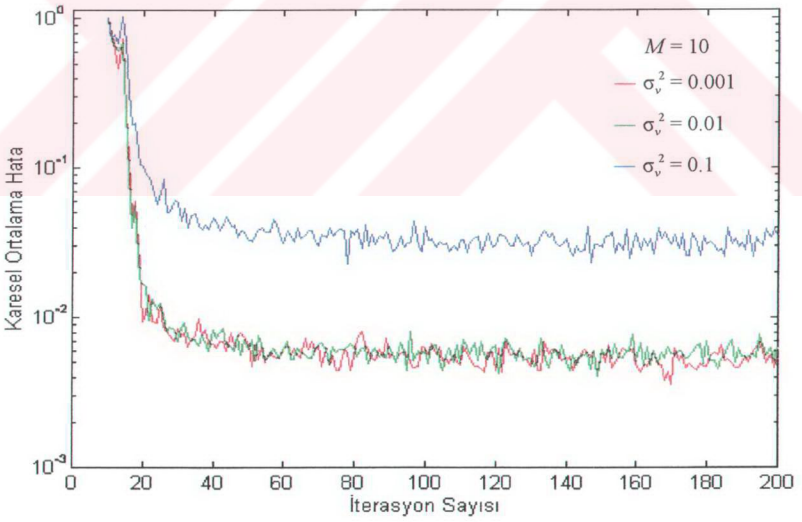
Şekil 6.30 (a)



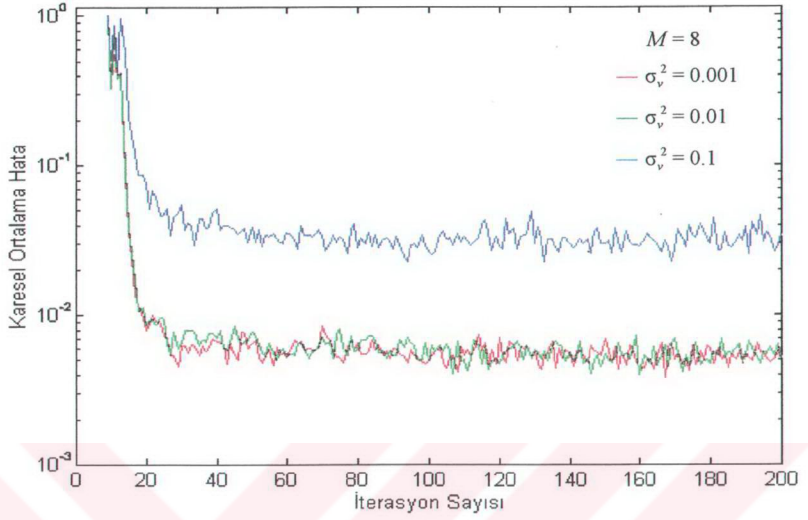
Şekil 6.30 (b)



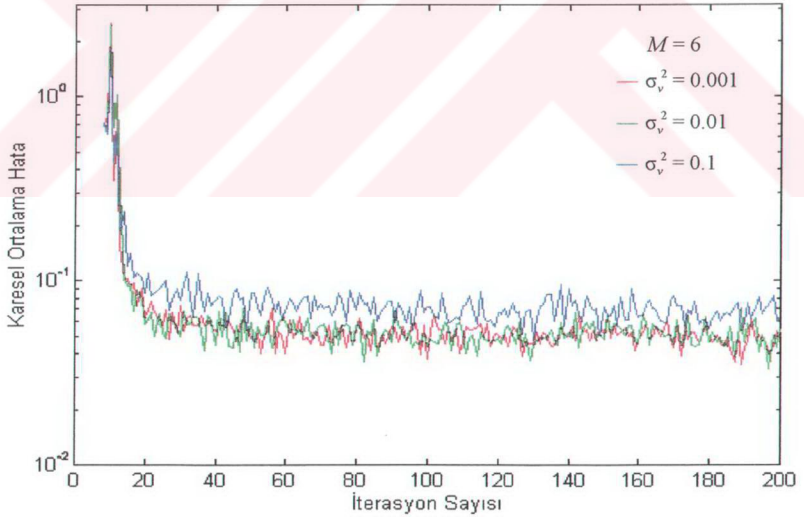
Şekil 6.30 (c)



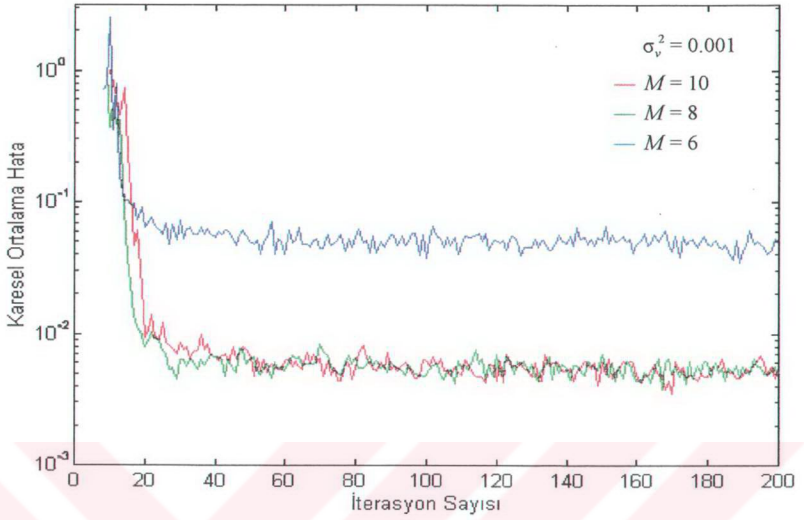
Şekil 6.31 (a)



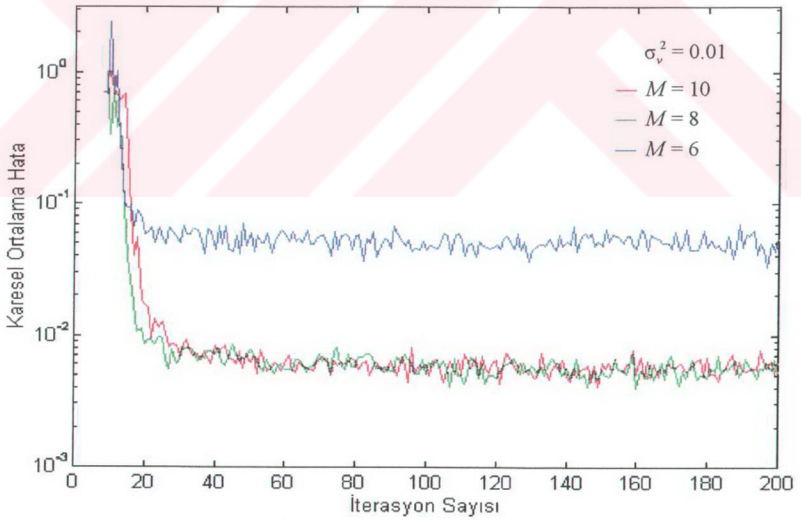
Şekil 6.31 (b)



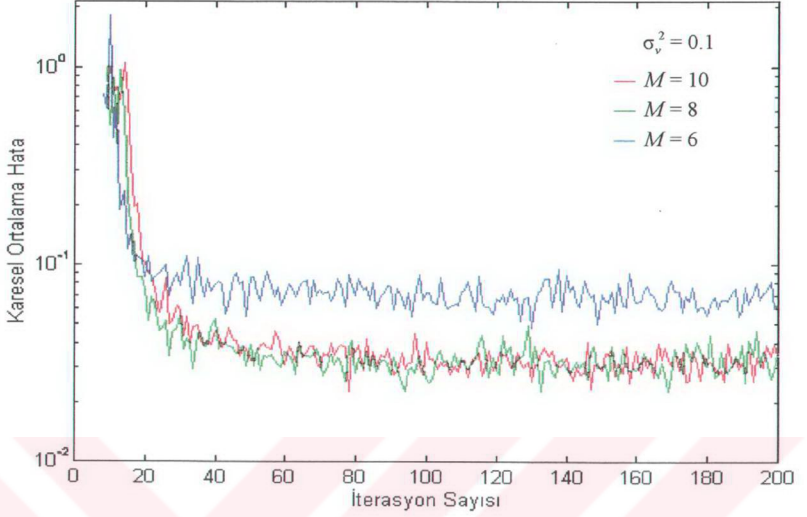
Şekil 6.31 (c)



Şekil 6.32 (a)



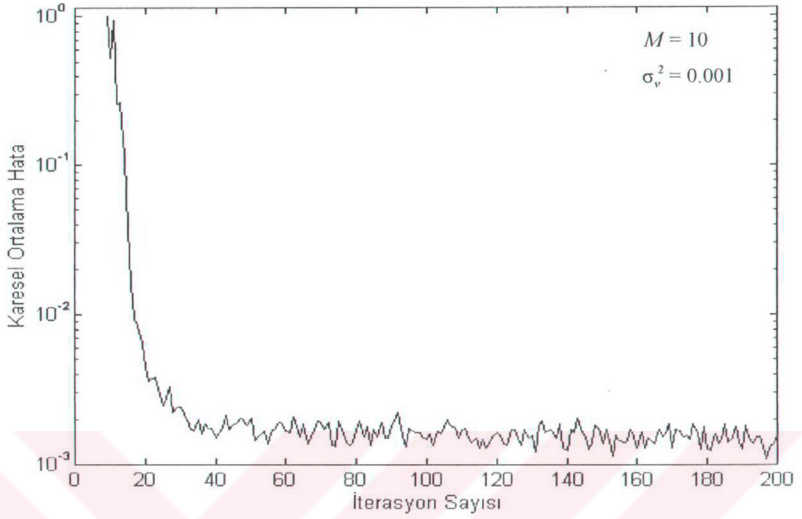
Şekil 6.32 (b)



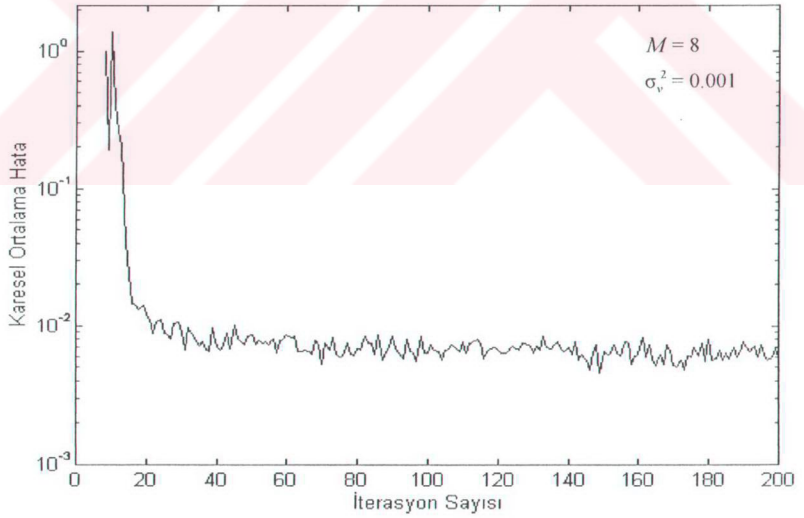
Şekil 6.32 (c)

Kendi belirlediğimiz kanal modellerinden ilki için yapılan deneylere ait bu eğrilerden de görüldüğü gibi, RLSL algoritmasının yakınsama özellikleri bu kanal modelinde de değişmemektedir. Yakınsama süresi, yüksek işaret-gürültü oranında (30 dB) M değerinin artmasıyla birlikte artarken, düşük işaret-gürültü oranlarında (20 ve 10 dB) her M değeri için aynı olmaktadır.

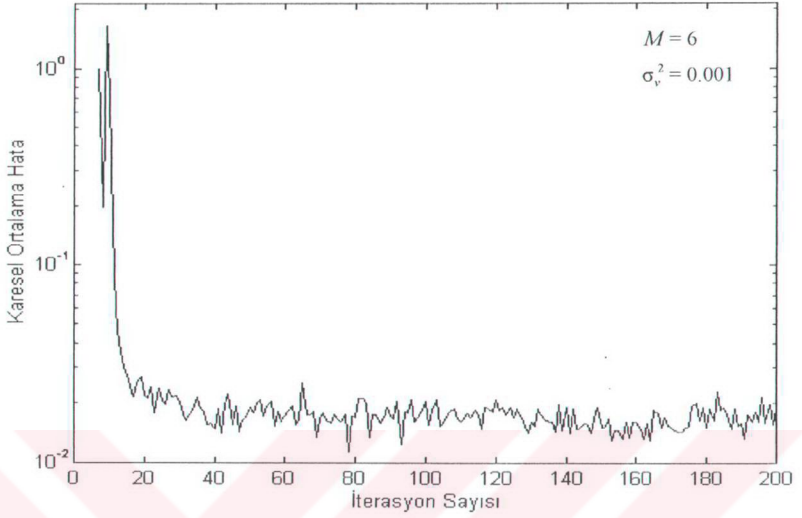
Kanalın yarattığı bozulmanın çok fazla olması nedeniyle, düşük gürültüde ($\sigma_v^2 = 0.001$) bile, $M = 10$ ve $M = 8$ için elde edilen karesel ortalama hata kalıcı hal değerleri birbirine eşit olmaktadır. Benzer durum, $\sigma_v^2 = 0.001$ ve $\sigma_v^2 = 0.01$ koşulları arasında da vardır; $\sigma_v^2 = 0.001$ ve $\sigma_v^2 = 0.01$ için elde edilen karesel ortalama hatalar, her M değerinde birbirleriyle aynı olmaktadır. Farklı σ_v^2 değerleri kullanılmasına rağmen, $M = 6$ için elde edilen karesel ortalama hataların hem yüksek değerli ve hem de birbirine bu kadar çok yakın olmasından bir tek şey anlaşılır: RLSL algoritması, $M = 6$ için bu kanal modelini dengeleyememektedir.



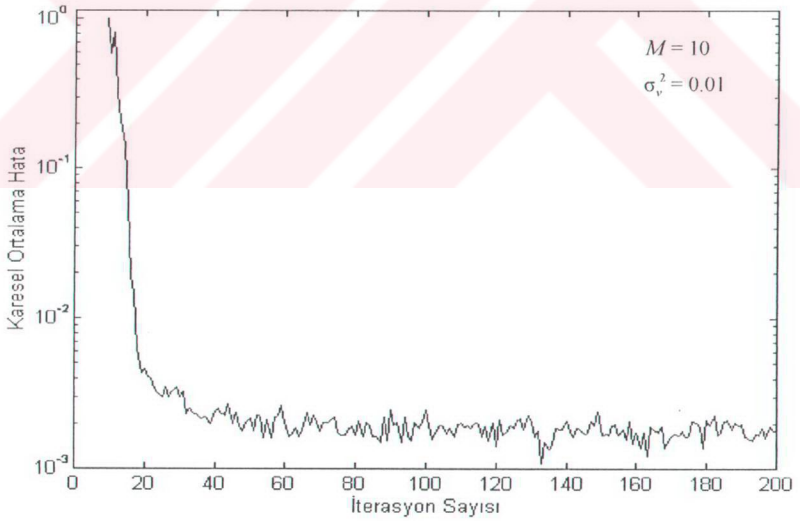
Şekil 6.33 (a)



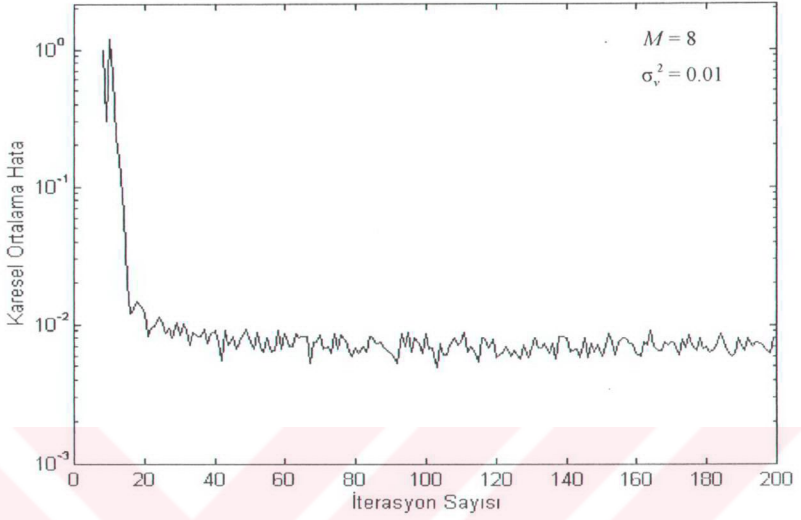
Şekil 6.33 (b)



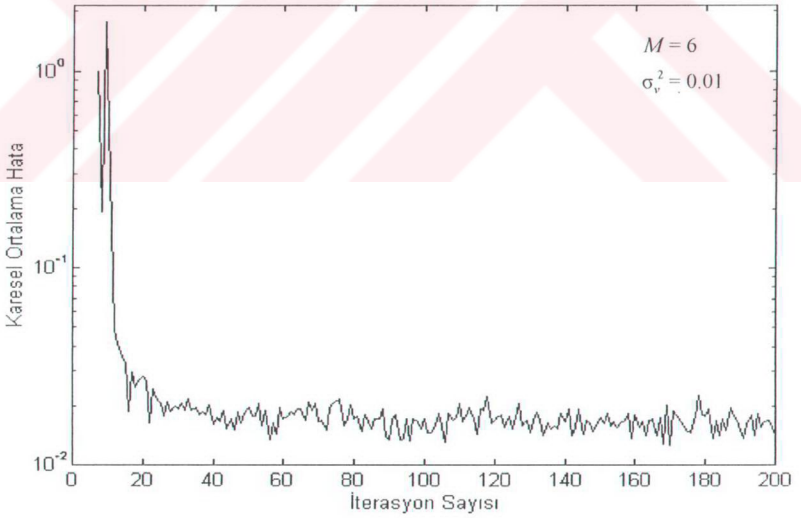
Şekil 6.33 (c)



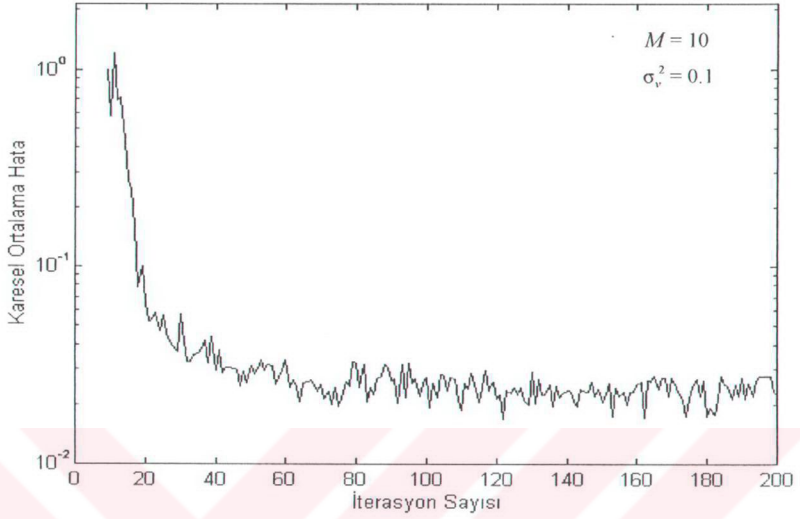
Şekil 6.34 (a)



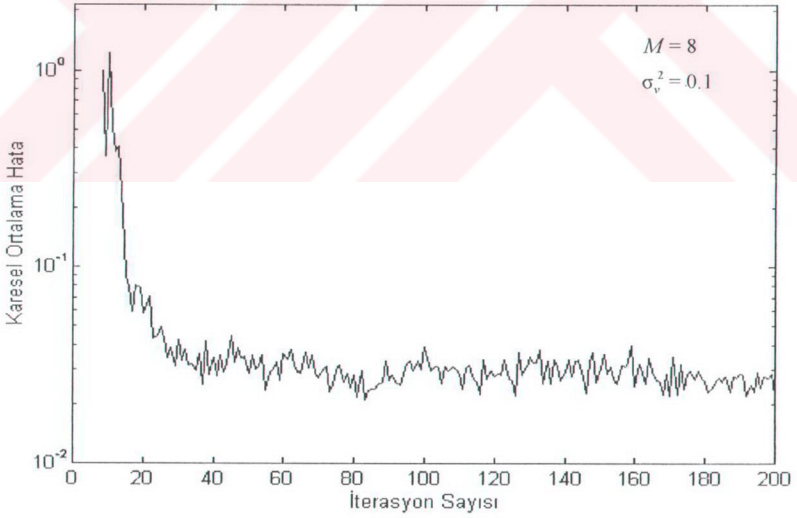
Şekil 6.34 (b)



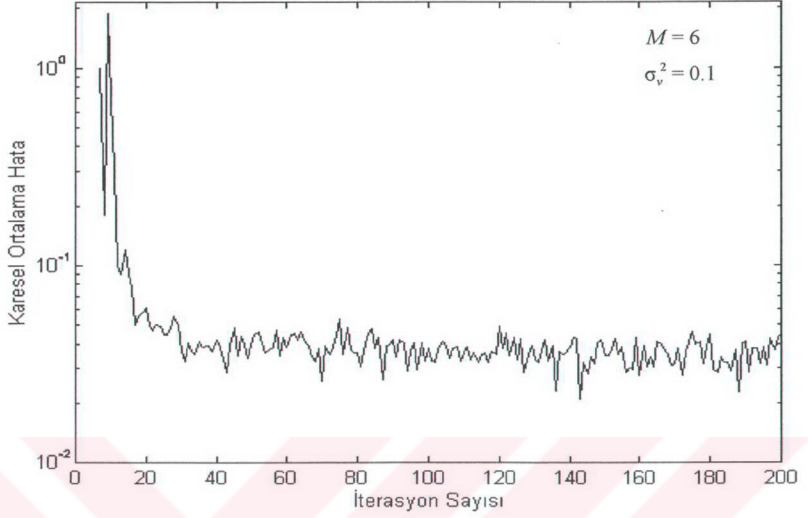
Şekil 6.34 (c)



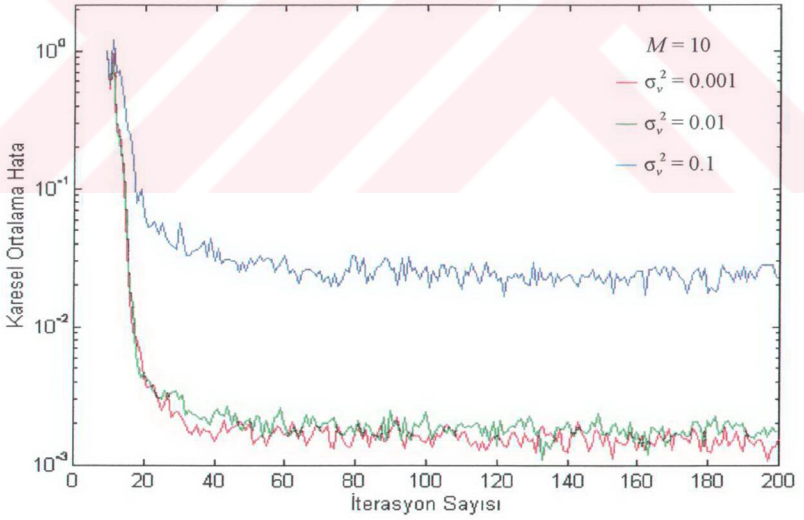
Şekil 6.35 (a)



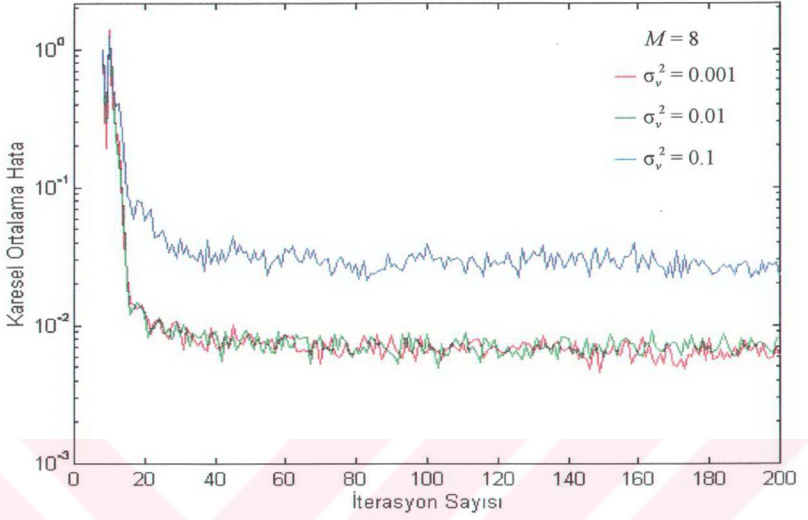
Şekil 6.35 (b)



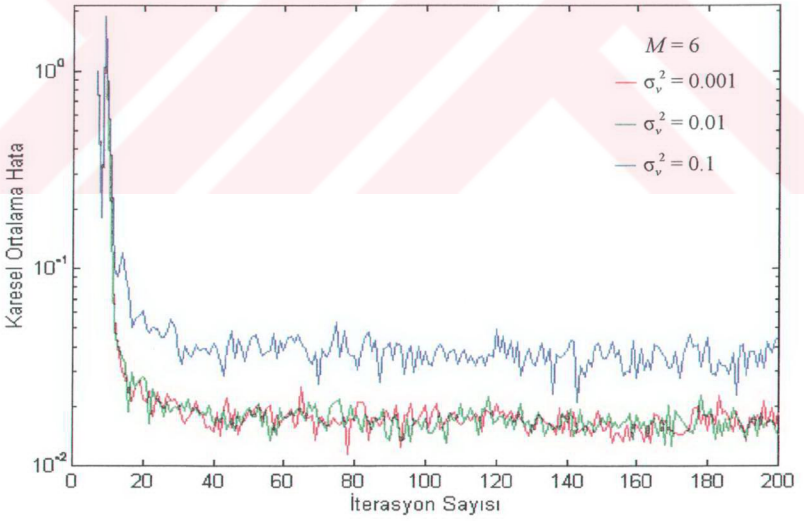
Şekil 6.35 (c)



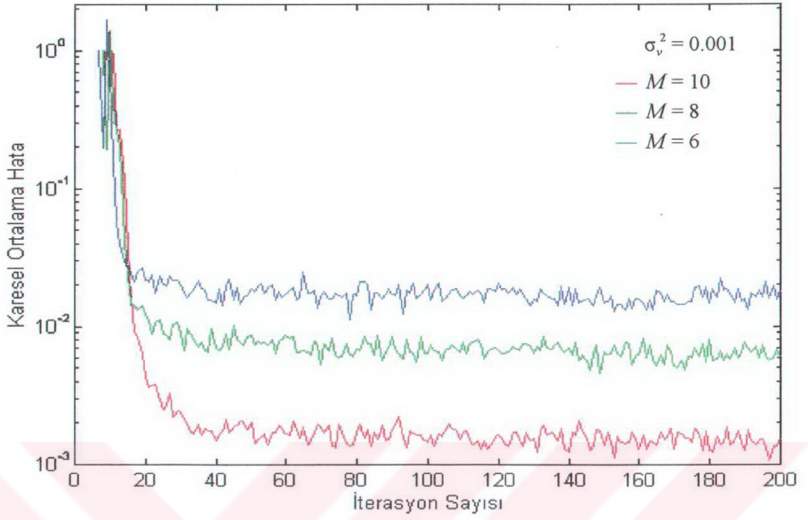
Şekil 6.36 (a)



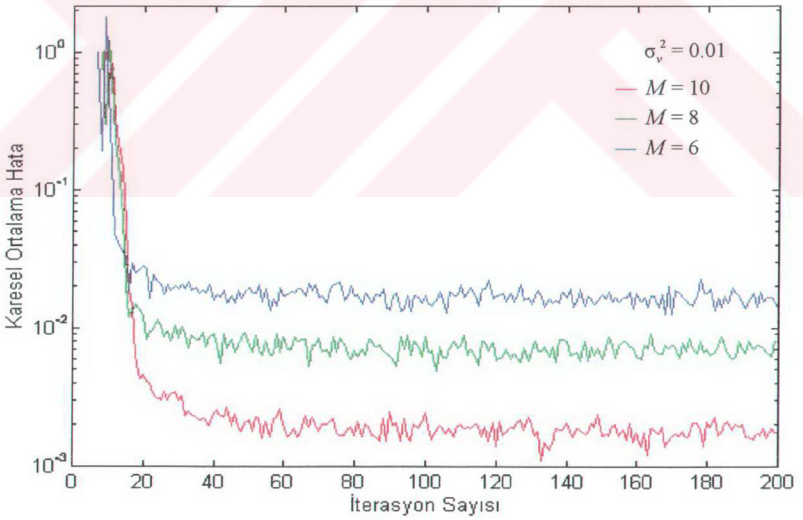
Şekil 6.36 (b)



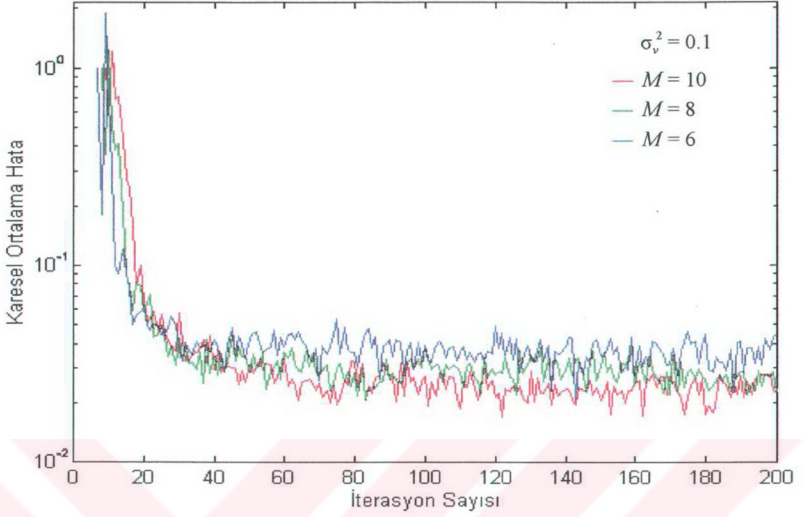
Şekil 6.36 (c)



Şekil 6.37 (a)



Şekil 6.37 (b)



Şekil 6.37 (c)

Kendi belirlediğimiz kanal modellerinden ikincisi için yapılan deneylere ait bu eğrilerden de görüldüğü gibi, RLSL algoritması aynı yakınsama özelliklerini bu kanal modelinde de sürdürmektedir.

Bu kanal modelinin ilk belirlediğimiz kanal modelinden farkı, yarattığı bozulmanın daha az oluşudur. Bunu, $M = 6$ için elde edilen karesel ortalama hata kalıcı hal değerlerinin daha düşük olmasından kolayca anlayabiliriz. Buna karşın, $\sigma_v^2 = 0.001$ ve $\sigma_v^2 = 0.01$ için elde edilen karesel ortalama hatalar, her M değerinde birbirleriyle yine aynı olmaktadır.

6.3.2 $|W(w)|$, $|H(w)|$ ve $|1/H(w)|$ eğrileri

Bu bölümde, RLSL algoritmasına dayanan $M = \{10, 8, 6\}$ değerli uyarlamalı dengeleyicilerimizin, impuls yanıtları (6.18), (6.19), (6.20), (6.21), (6.22) ve (6.23) eşitlikleriyle verilen kanal modellerini $\sigma_v^2 = \{0.001, 0.01, 0.1\}$ gürültü koşulları altında ne derece dengeleyebildiklerini, yani kendilerini ne derece bu kanalların tersine benzetebildiklerini araştıracağız. Bu da ancak, kanalın tersine ait $|1/H(w)|$ eğrisi ile uyarlamalı dengeleyicilere ait $|W(w)|$ eğrileri karşılaştırılarak öğrenilebilir. İdeal durum olarak, $|W(w)| = |1/H(w)|$ olması istenir.

İlk önce, kanal modeline ait $|H(w)|$ eğrisi tek başına verilmiş ve ardından, o kanalın tersine ait $|1/H(w)|$ eğrisi ile (aynı gürültü koşulları altında çalıştırılan) uyarlamalı dengeleyicilere ait $|W(w)|$ eğrileri karşılaştırılmıştır. Üç farklı gürültü koşulu belirlenmiş olduğundan, her bir kanal modeli için üç karşılaştırma eğrisi elde edilmiştir. Karşılaştırma eğrilerinde: kalın-düz çizgiyle çizdirilen eğri, $|1/H(w)|$ 'i; normal-düz çizgiyle çizdirilen eğri, $M = 10$ için olan $|W(w)|$ 'i; uzun-kesikli çizgilerle çizdirilen eğri, $M = 8$ için olan $|W(w)|$ 'i ve kısa-kesikli çizgilerle çizdirilen eğri ise, $M = 6$ için olan $|W(w)|$ 'i göstermektedir.

İmpuls yanıtı (6.18) eşitliğiyle verilen kanal modeli için elde edilen eğriler, Şekil 6.38'de gösterilmiştir. $|W(w)|$ eğrilerinin hepsinin $|1/H(w)|$ eğrisi ile üst üste çakışması, her üç dengeleyicinin bu kanal modelini ideal bir şekilde dengeleyebildikleri anlamına gelir. $\sigma_v^2 = 0.1$ gürültü koşulu için meydana gelen sapmalar (Şekil 6.38 (d)), ihmal edilebilecek kadar önemsizdir.

İmpuls yanıtı (6.19) eşitliğiyle verilen kanal modeli için elde edilen eğriler, Şekil 6.39'da gösterilmiştir. Bu kanal modeli de ideal bir şekilde dengelenmektedir. Ancak, artan kanal bozulması karşısında, eğrilerin giderek tam üst üste çakışmamaya başladıkları görülmektedir. Bu durum özellikle, $\sigma_v^2 = 0.1$ koşulu için elde edilen eğrilerde daha belirgindir. Nitekim, $\sigma_v^2 = 0.1$ gürültü koşulu için bu kanal modelinde ortaya çıkan sapma

(Şekil 6.39 (d)), bir önceki kanal modelinde aynı gürültü koşulu için ortaya çıkan sapmadan (Şekil 6.38 (d)) daha çoktur.

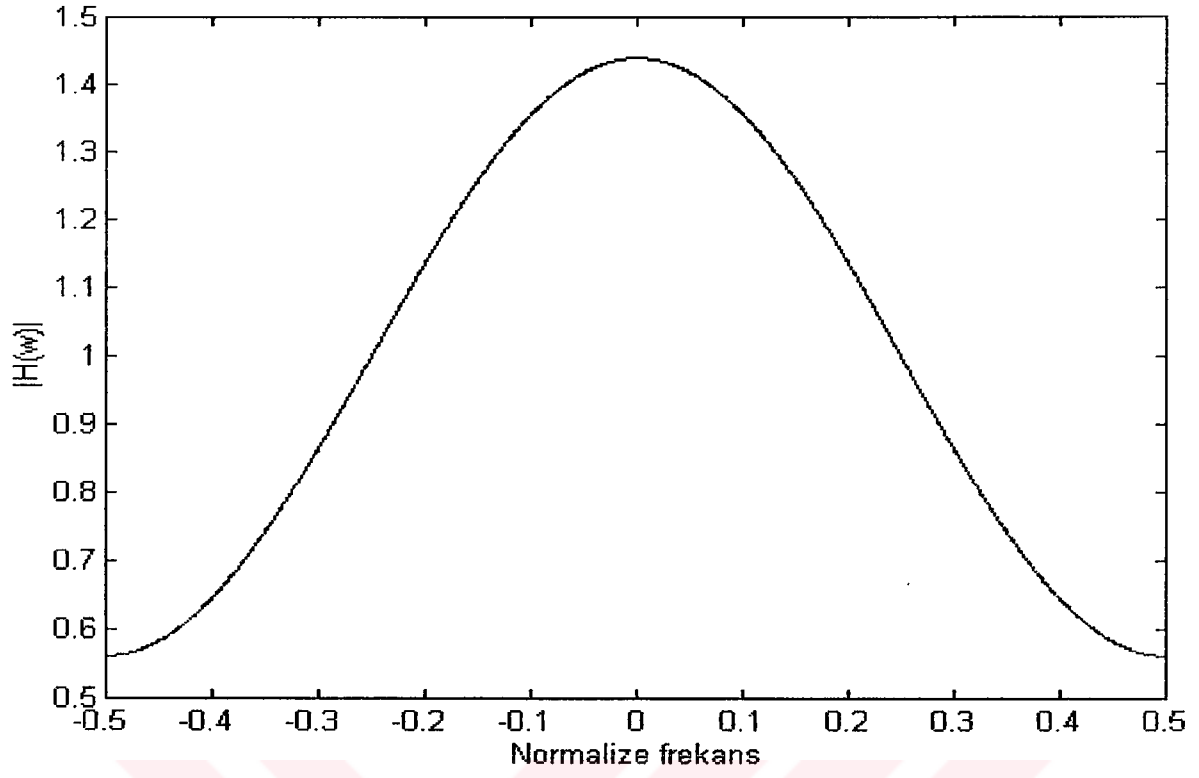
İmpuls yanıtı (6.20) eşitliğiyle verilen kanal modeli için elde edilen eğriler, Şekil 6.40'da gösterilmiştir. $M = 10$ ve $M = 8$ değerli dengeleyiciler, $\sigma_v^2 = 0.001$ ve $\sigma_v^2 = 0.01$ koşullarında ideal bir dengeleme gerçekleştirmelerine karşın, $\sigma_v^2 = 0.1$ koşulu için ideal durumdan küçük sapmalarla uzaklaşmışlardır. $M = 6$ değerli dengeleyici ise, her σ_v^2 koşulu için küçük sapmalarla ideal durumdan uzaklaşmıştır. Bu da göstermektedir ki: kanalın yarattığı bozulma fazla ise, o kanal için M değeri büyük olan bir dengeleyici kullanılması gerekir. Tüm bunlara rağmen, bu kanal modelinin başarılı bir şekilde dengelendiği söylenebilir.

İmpuls yanıtı (6.21) eşitliğiyle verilen kanal modeli için elde edilen eğriler, Şekil 6.41'de gösterilmiştir. Daha önce birbirleriyle aynı sonuca ulaşan $M = 10$ ve $M = 8$ değerli dengeleyiciler, giderek artan kanal bozulması karşısında farklı sonuçlar elde etmeye başlamışlardır. Bu da, kanal bozulmasının yüksek olması durumunda, M değerinin büyük seçilmesi gerektiğini bir kez daha vurgular. Diğer yandan, kanal bozulması ile birlikte gürültünün yarattığı bozulmanın da artması, Şekil 6.41 (d)'deki duruma sebep olur. Çünkü, $M = 10$ için bile, ideal durumdan oldukça uzaklaşmaktadır. Yine de, bu kanal modelinin kötü sayılmayacak kadar iyi bir şekilde dengelendiğini söylemek mümkündür.

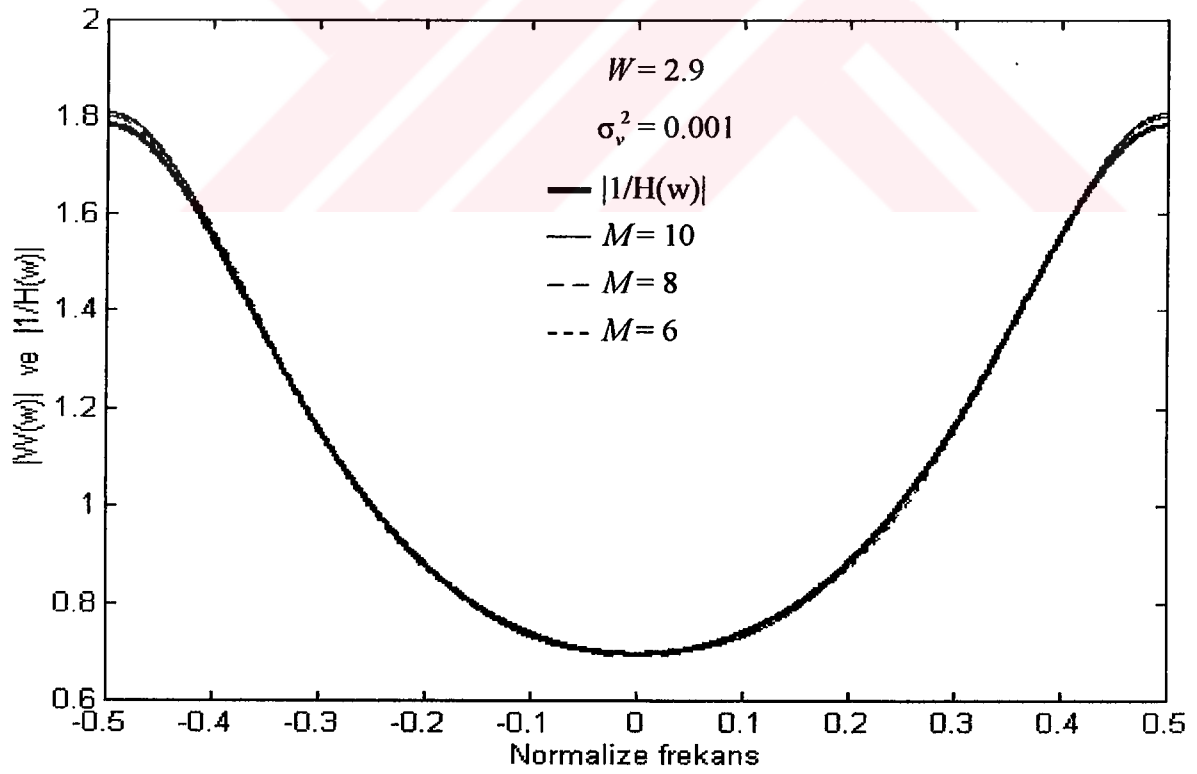
İmpuls yanıtı (6.22) eşitliğiyle verilen kanal modeli için elde edilen eğriler, Şekil 6.42'de gösterilmiştir. Hatırlanacağı gibi Bölüm 6.3.1'de, RLSL algoritmasının: $M = 6$ için bu kanal modelini dengeleyemediği; $M = 10$ ve $M = 8$ için elde edilen karesel ortalama hata kalıcı hal değerlerinin, her σ_v^2 koşulunda birbirleriyle aynı oldukları; $\sigma_v^2 = 0.001$ ve $\sigma_v^2 = 0.01$ koşulları için elde edilen karesel ortalama hata kalıcı hal değerlerinin ise, her M değerinde birbirleriyle aynı oldukları söylenmişti. Öğrenme eğrilerine bakarak çıkardığımız bu sonuçlar, buradaki eğriler tarafından da doğrulanmaktadır. $M = 6$ için elde edilen $|W(w)|$ eğrilerinin $|1/H(w)|$ eğrisinden çok farklı olması, $M = 6$ değerli uyarlamalı dengeleyicinin bu kanal modelini dengeleyemediğini gösterir. Öte yandan, $M = 10$ ve $M = 8$ için elde edilen $|W(w)|$ eğrileri, her σ_v^2 koşulunda üst üste çakışmaktadır. Bu da, $M = 10$ ve $M = 8$ değerli

uyarlamalı dengeleyicilerin, bu kanalı aynı başarıyla dengelediklerini gösterir. Kanalın yarattığı bozulmanın fazla oluşu, bu iki dengeleyiciye ait $|W(w)|$ eğrilerinin idealden uzaklaşmasına neden olmuştur. Son olarak, $\sigma_v^2 = 0.001$ koşuluna ait Şekil 6.42 (b) ile $\sigma_v^2 = 0.01$ koşuluna ait Şekil 6.42 (c) karşılaştırıldığında, bu şekillerin birbiriyle aynı olduğu görülür.

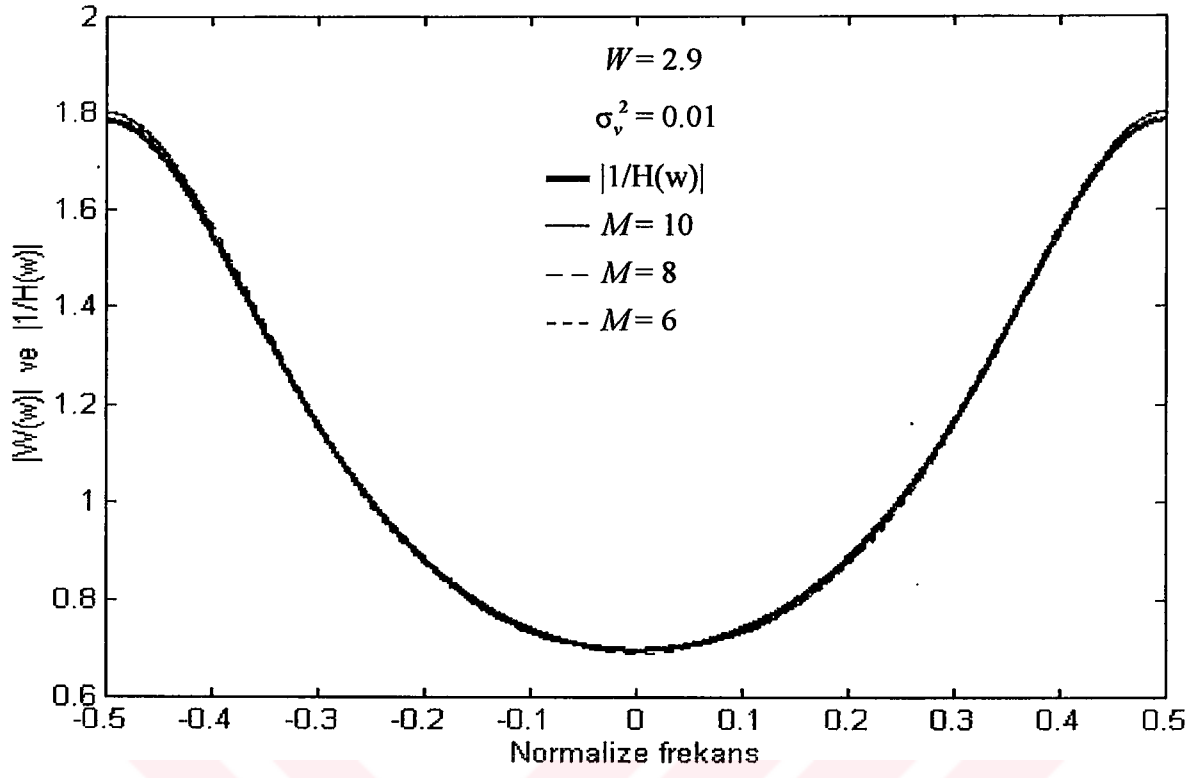
İmpuls yanıtı (6.23) eşitliğiyle verilen kanal modeli için elde edilen eğriler, Şekil 6.43’de gösterilmiştir. Hatırlanacağı gibi Bölüm 6.3.1’de, $\sigma_v^2 = 0.001$ ve $\sigma_v^2 = 0.01$ koşulları için elde edilen karesel ortalama hata kalıcı hal değerlerinin, her M değerinde birbirleriyle aynı oldukları söylenmişti. Bu sonuç, buradaki Şekil 6.43 (b) ve Şekil 6.43 (c) tarafından da doğrulanmaktadır. Çünkü her iki şekil, bir önceki kanal modelinde olduğu gibi, tepe ve alt noktalarının aldığı değerlere kadar birbirinin aynısı olmaktadır. Buna göre, bu kanal modelinde gürültünün $\sigma_v^2 = 0.001$ ’den $\sigma_v^2 = 0.01$ ’e çıkması, $M = \{10, 8, 6\}$ değerli uyarlamalı dengeleyiciler için bir şey değiştirmemektedir; dengeleyicilerin her biri, her iki gürültü koşulunda da, kanalı aynı ölçüde dengelemektedir. Bu kanal modeli için de ideal durum sağlanamamaktadır. İdeale en yakın dengeleme ise, $M = 10$ değerli dengeleyici tarafından gerçekleştirilmektedir.



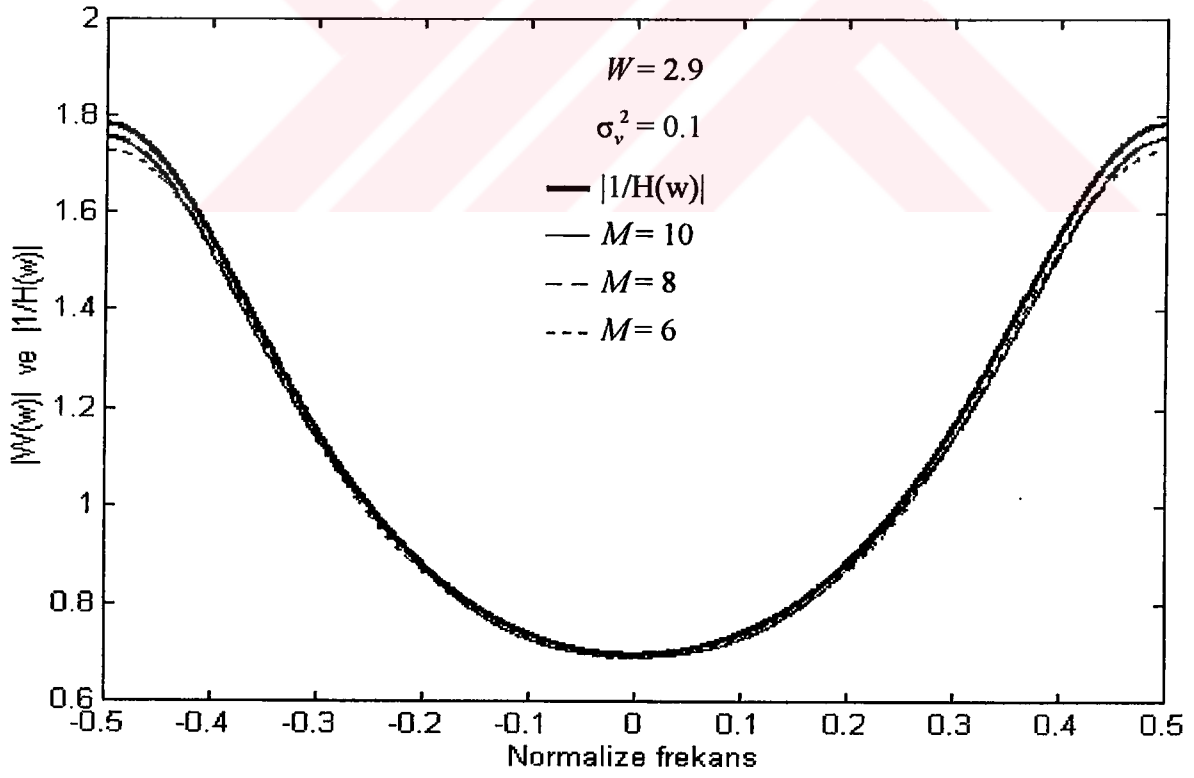
Şekil 6.38 (a)



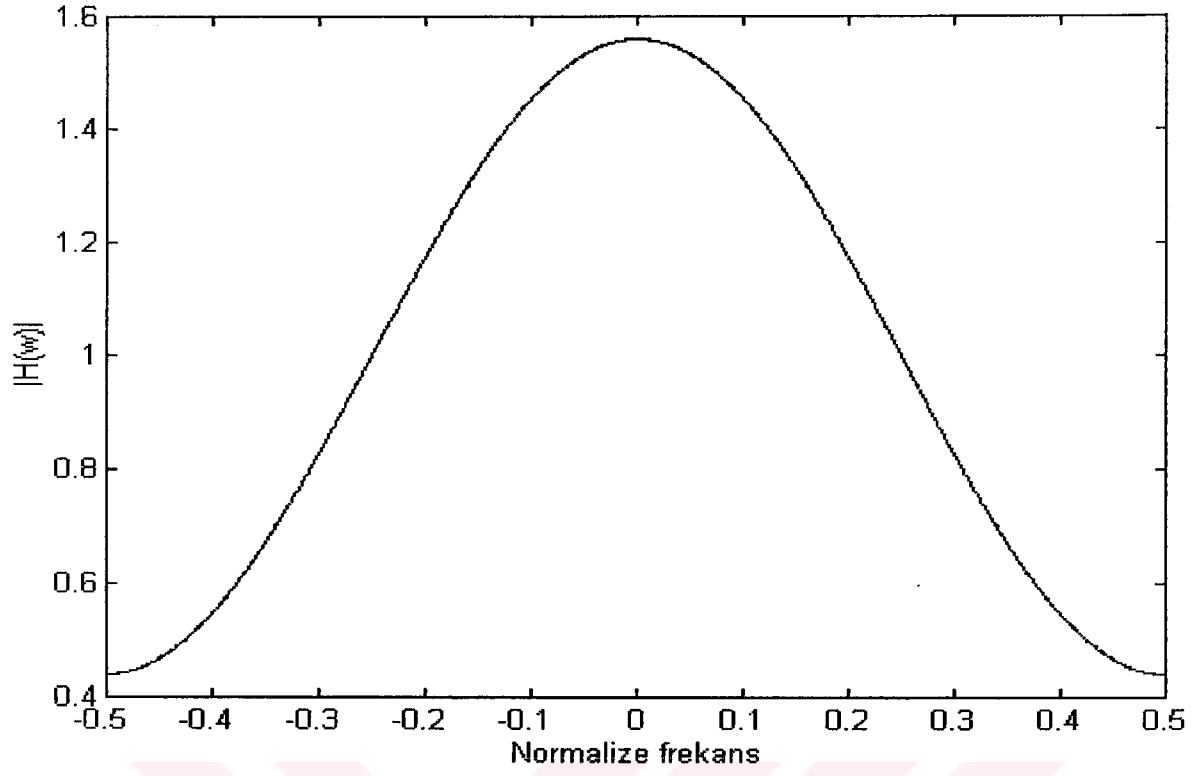
Şekil 6.38 (b)



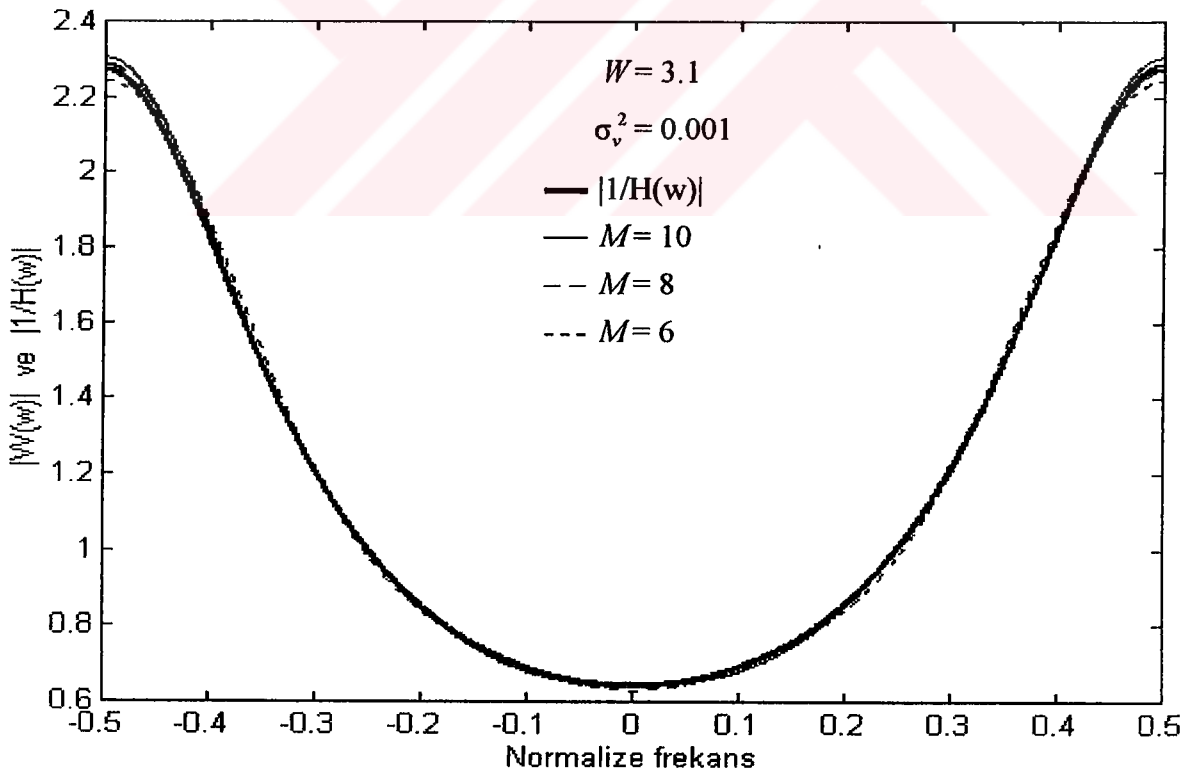
Şekil 6.38 (c)



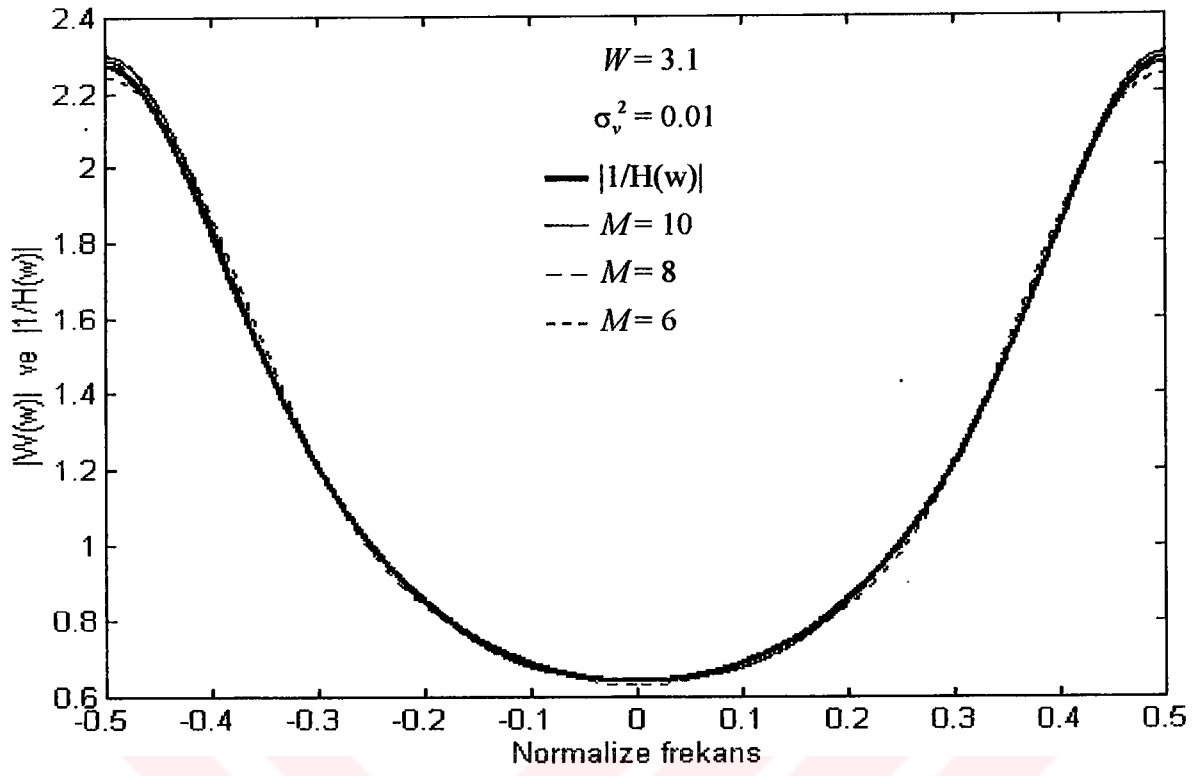
Şekil 6.38 (d)



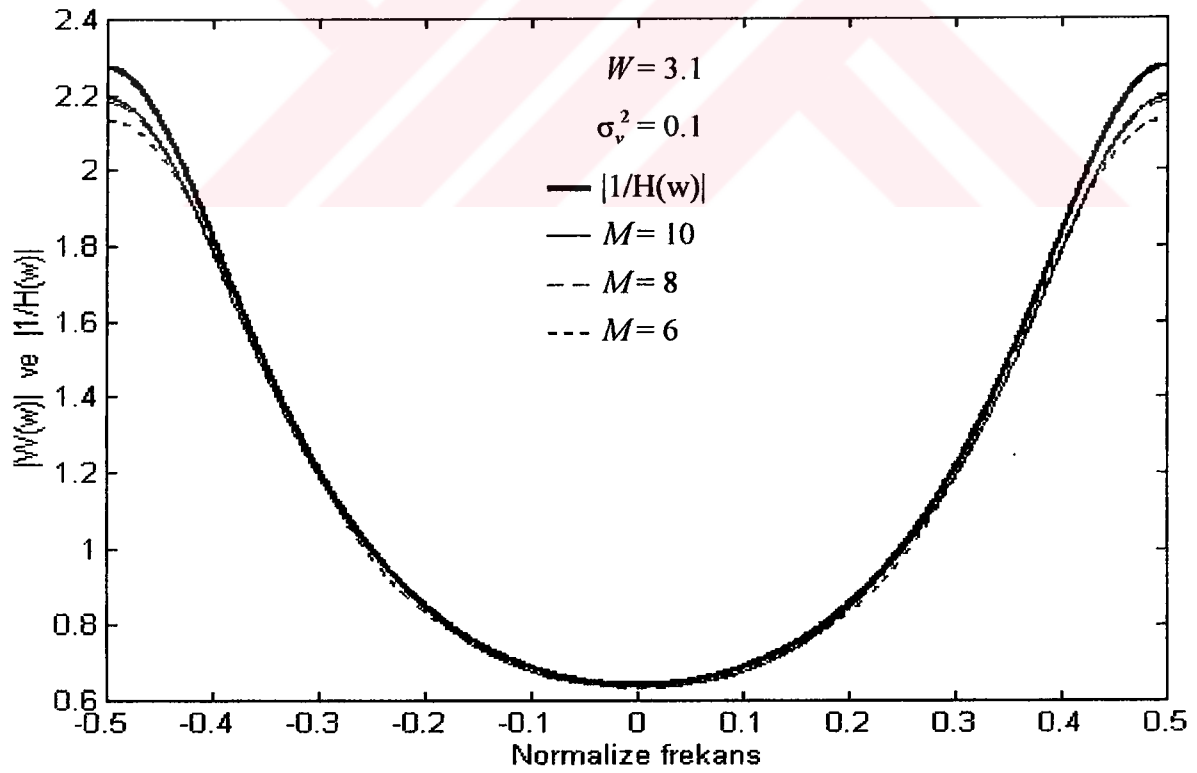
Şekil 6.39 (a)



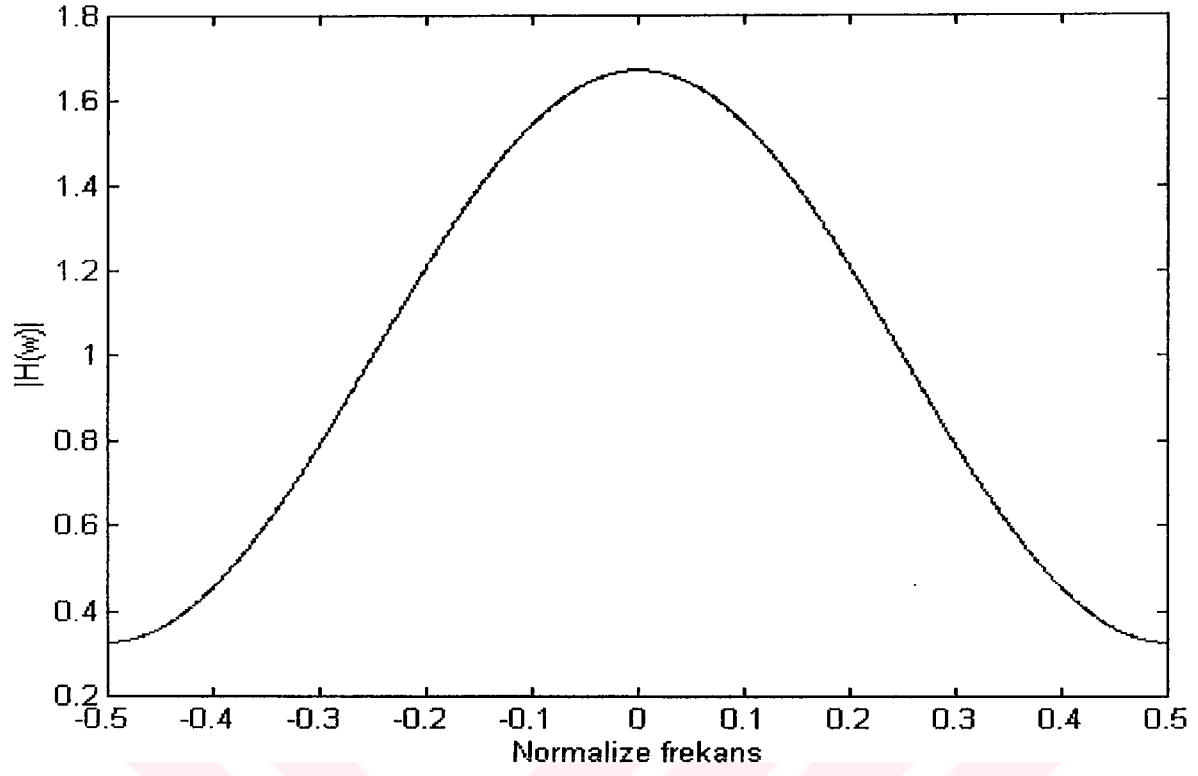
Şekil 6.39 (b)



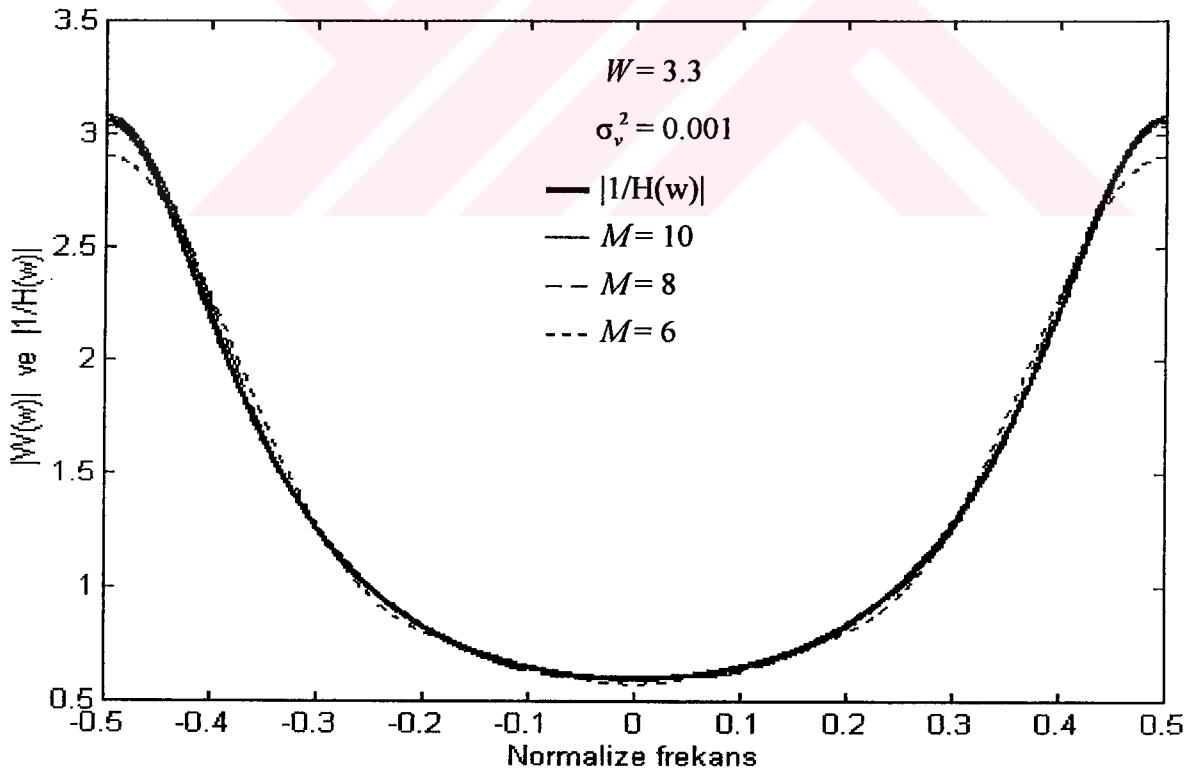
Şekil 6.39 (c)



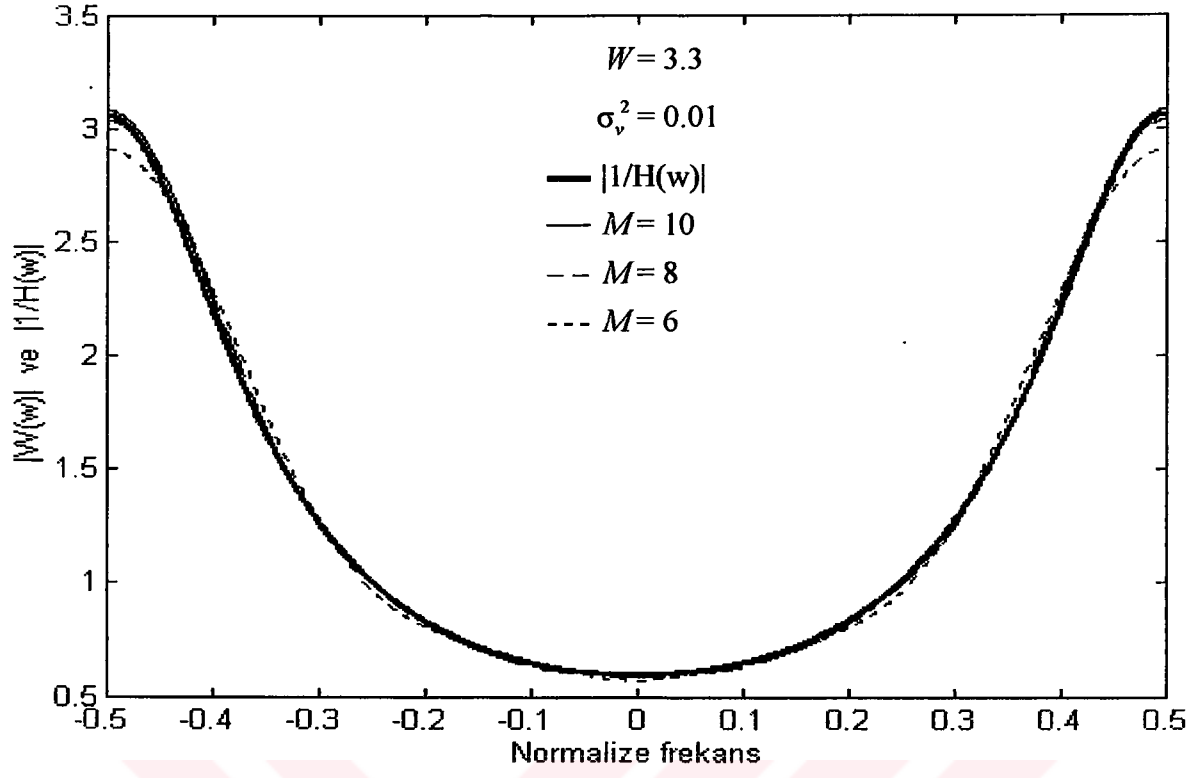
Şekil 6.39 (d)



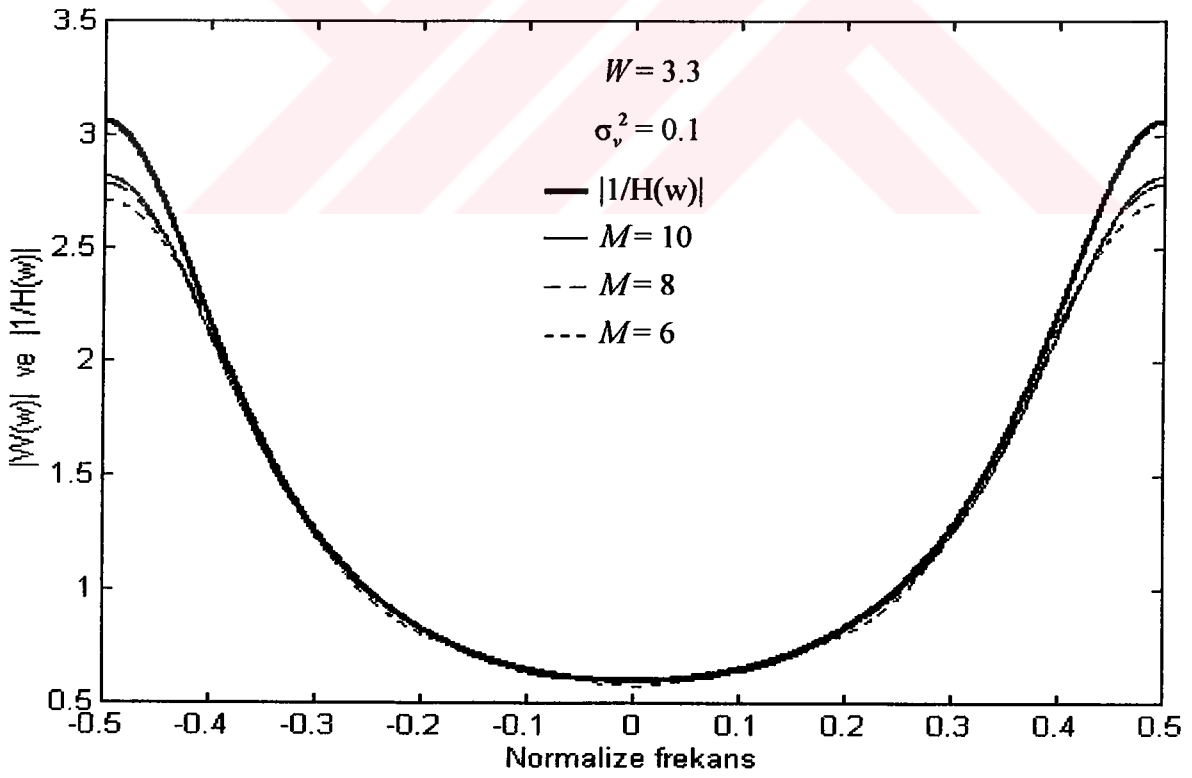
Şekil 6.40 (a)



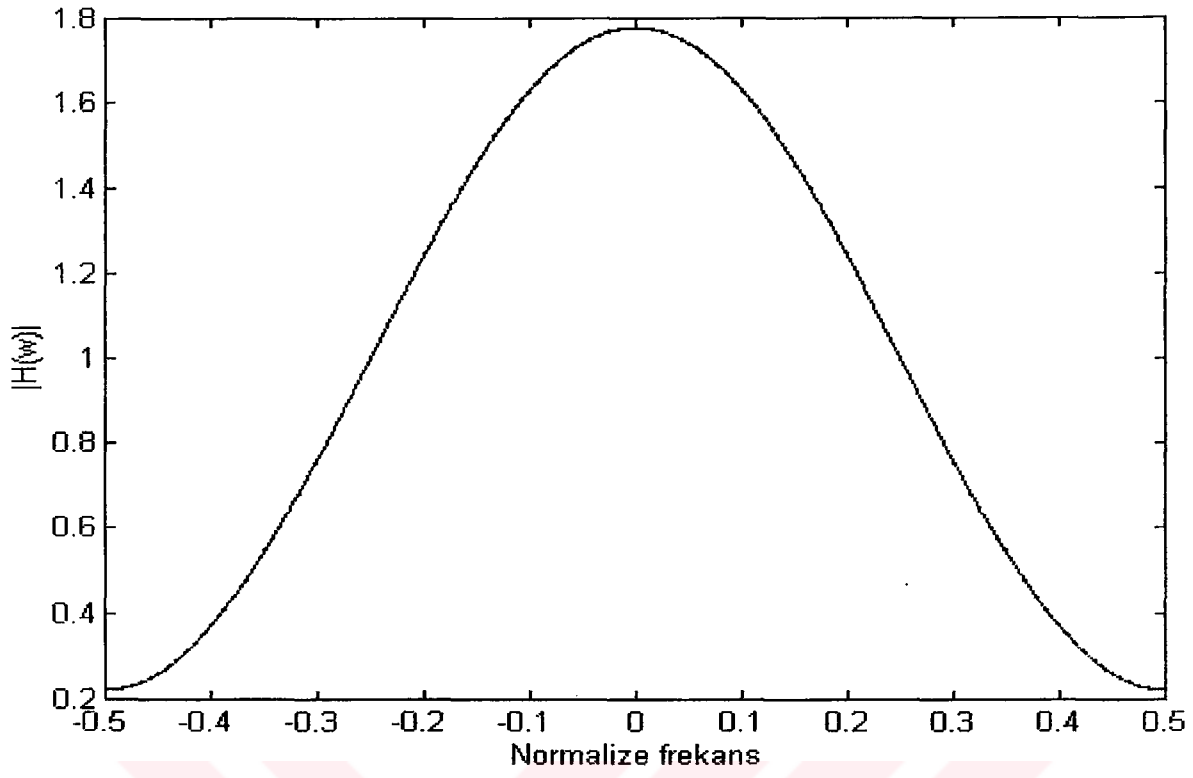
Şekil 6.40 (b)



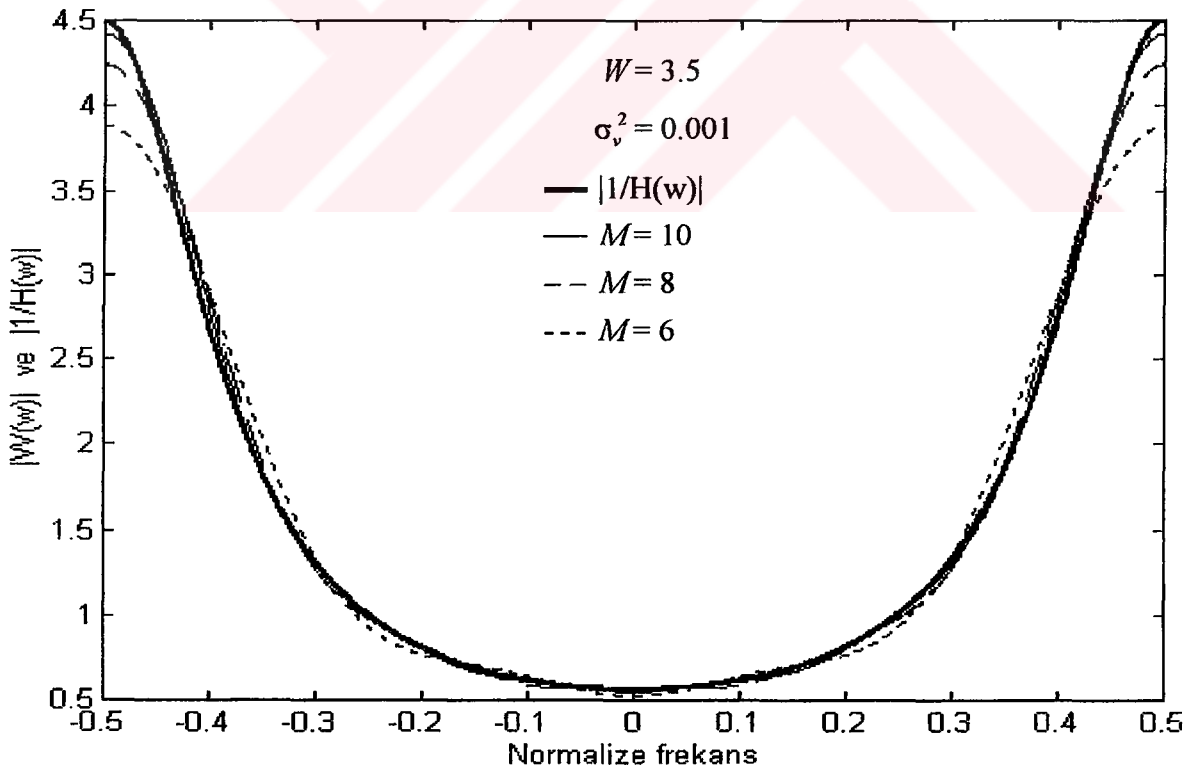
Şekil 6.40 (c)



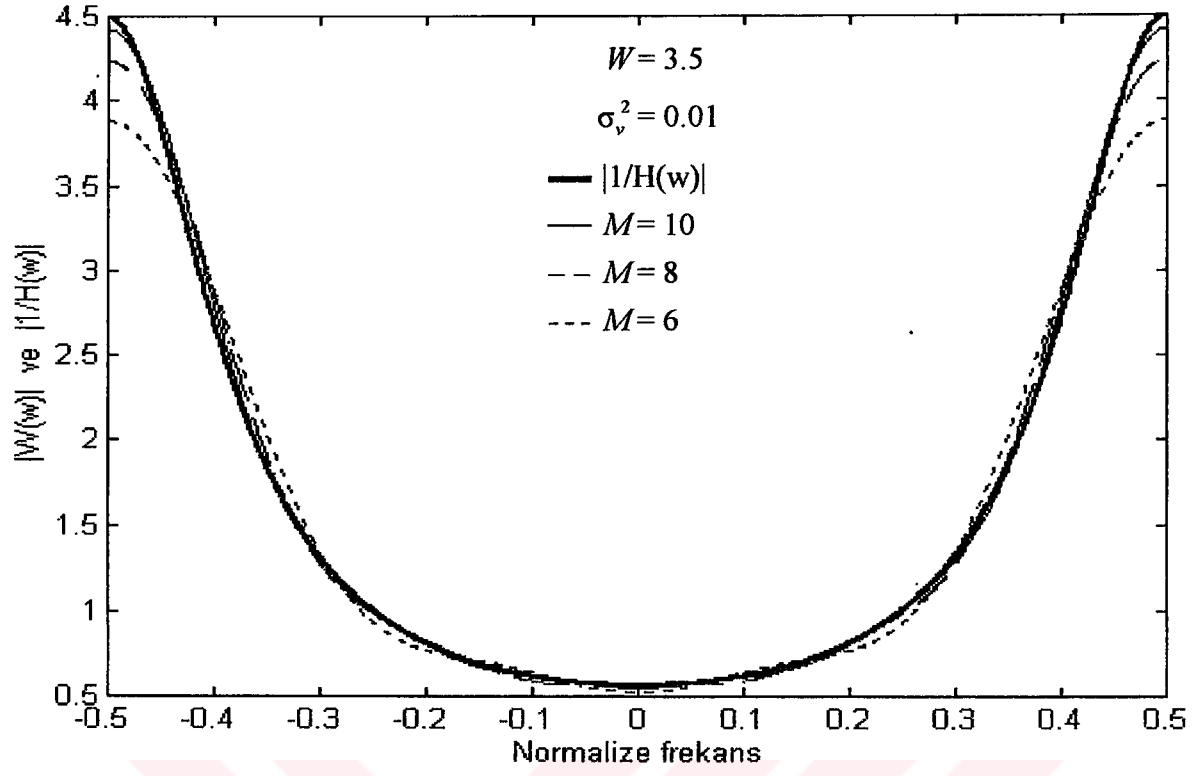
Şekil 6.40 (d)



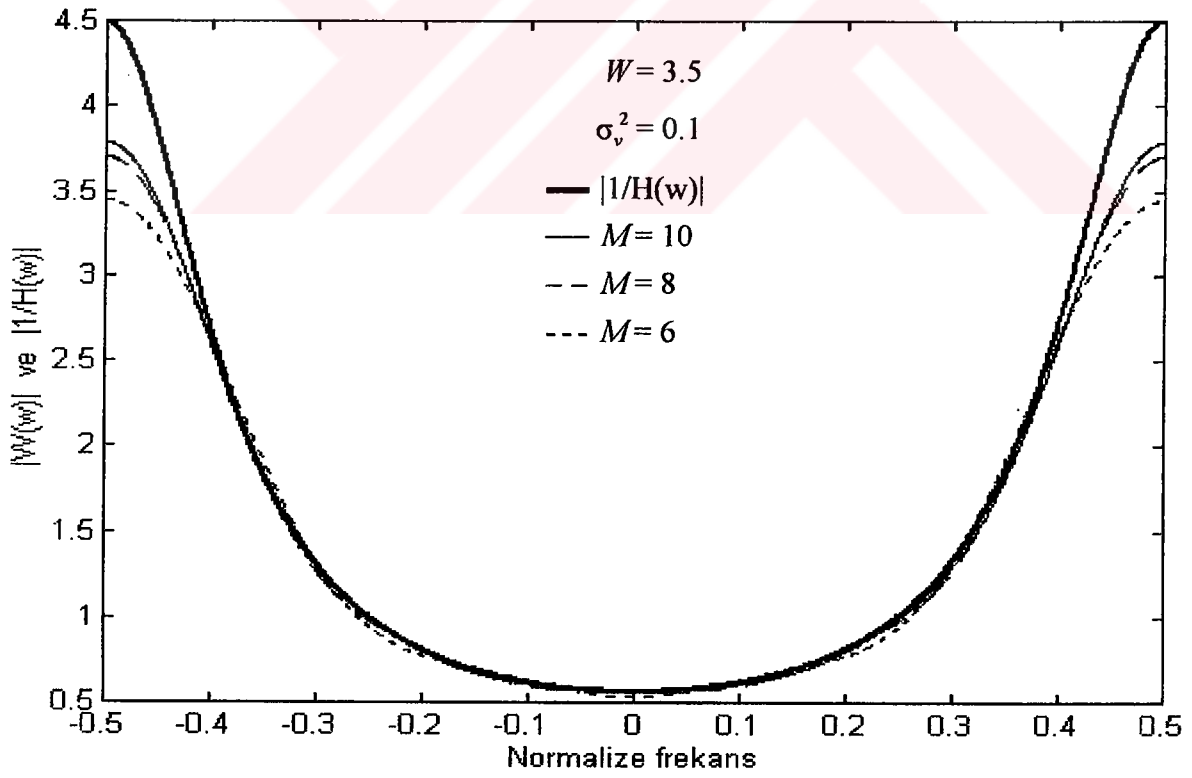
Şekil 6.41 (a)



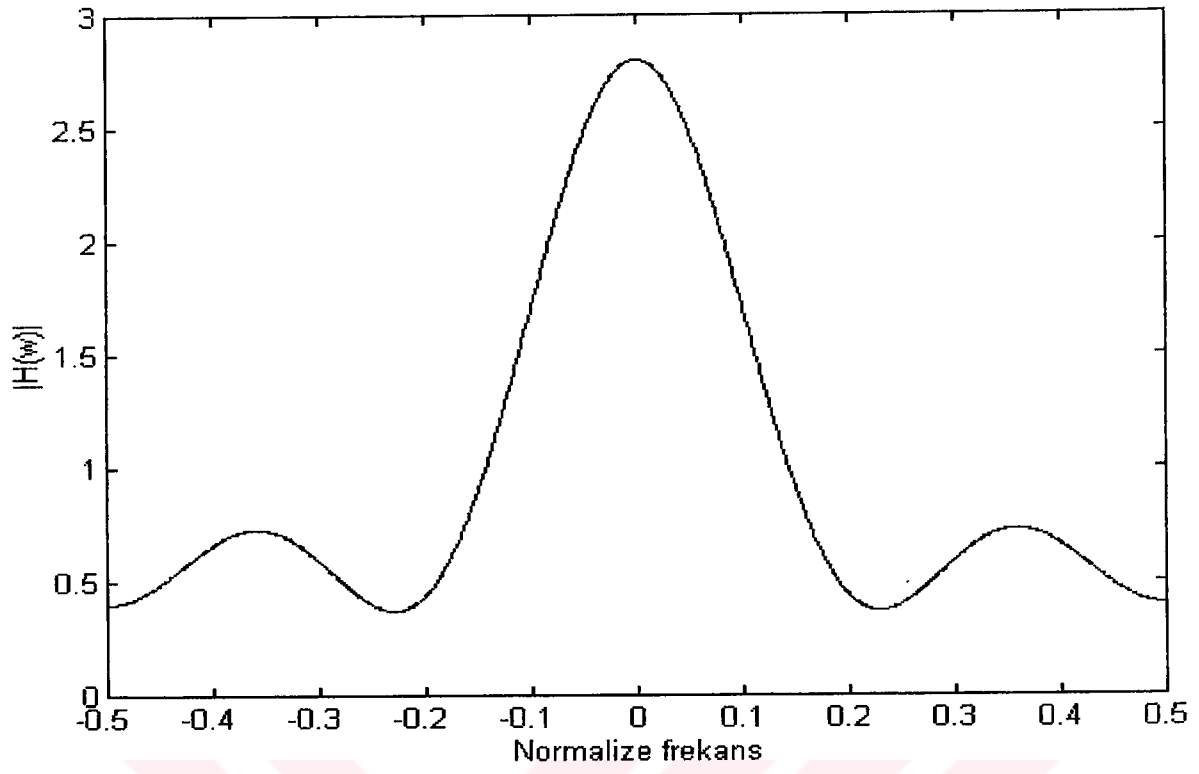
Şekil 6.41 (b)



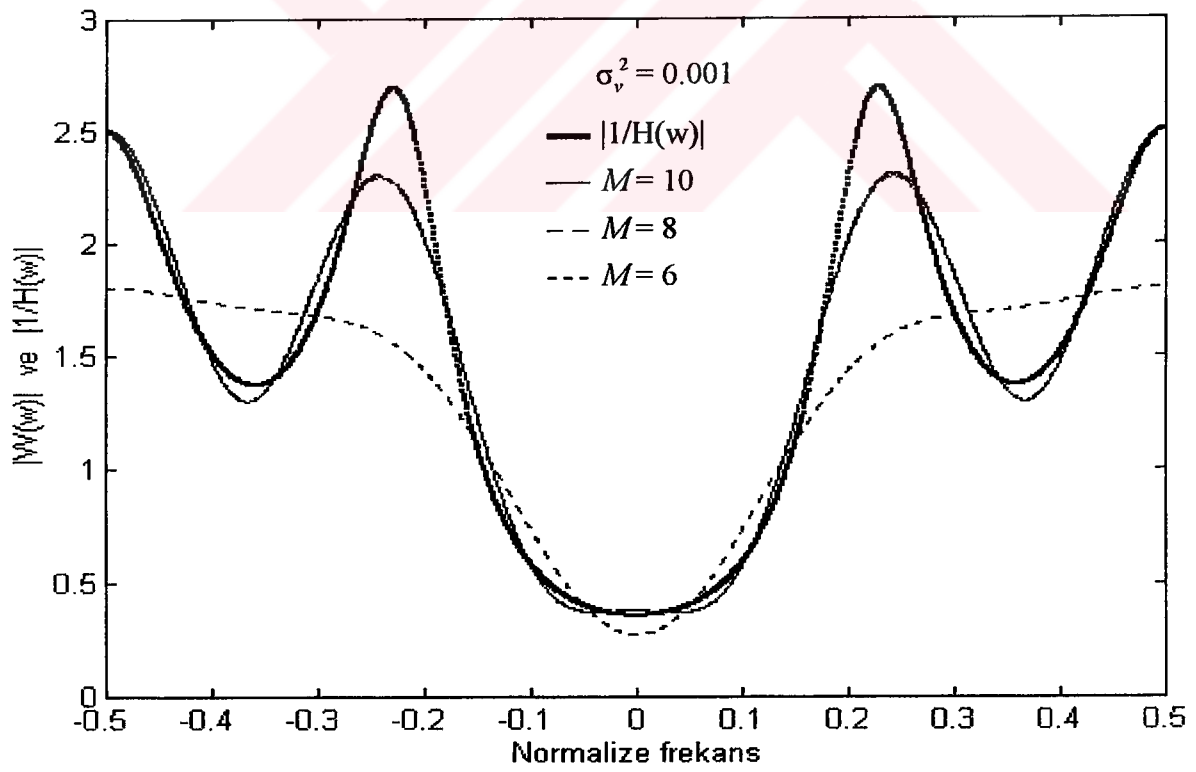
Şekil 6.41 (c)



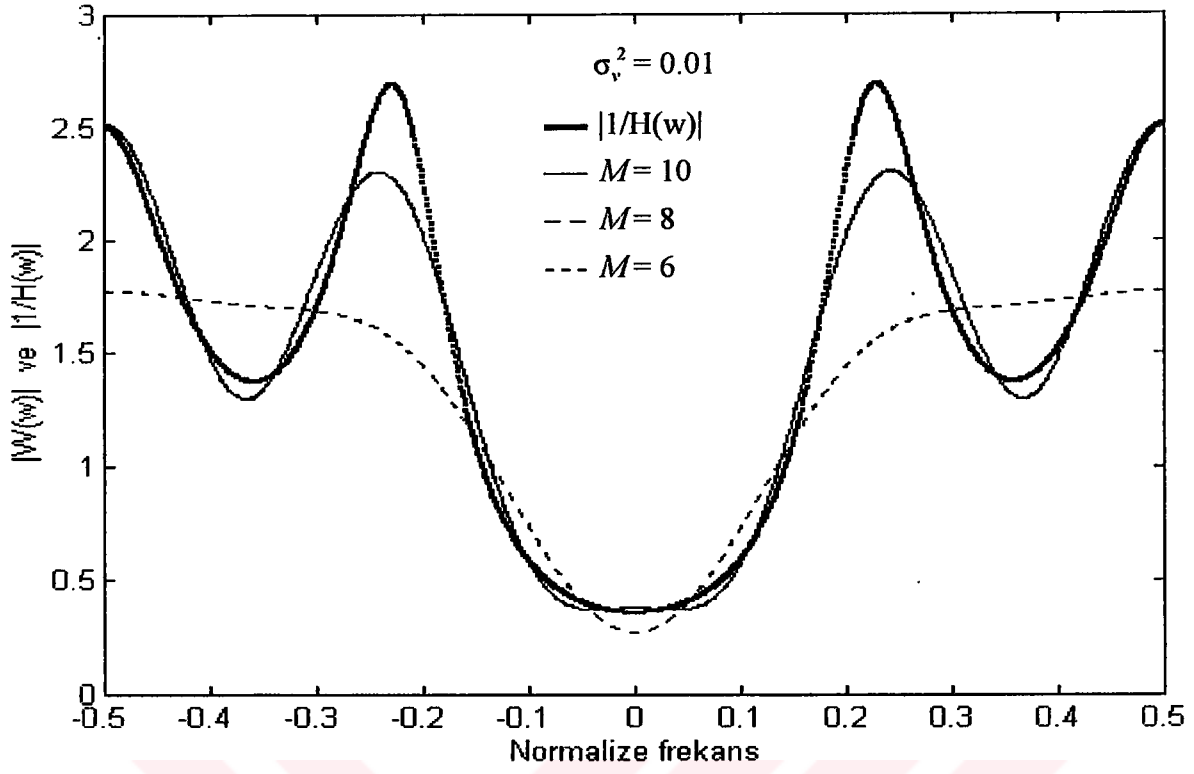
Şekil 6.41 (d)



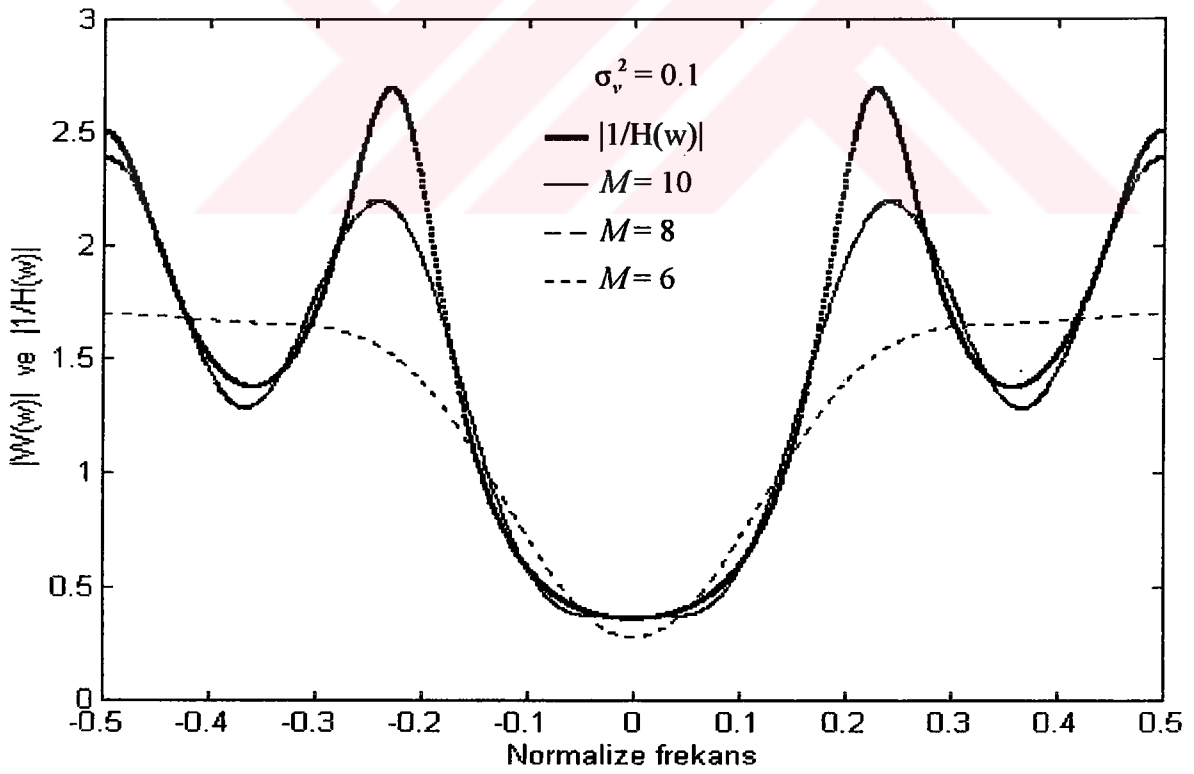
Şekil 6.42 (a)



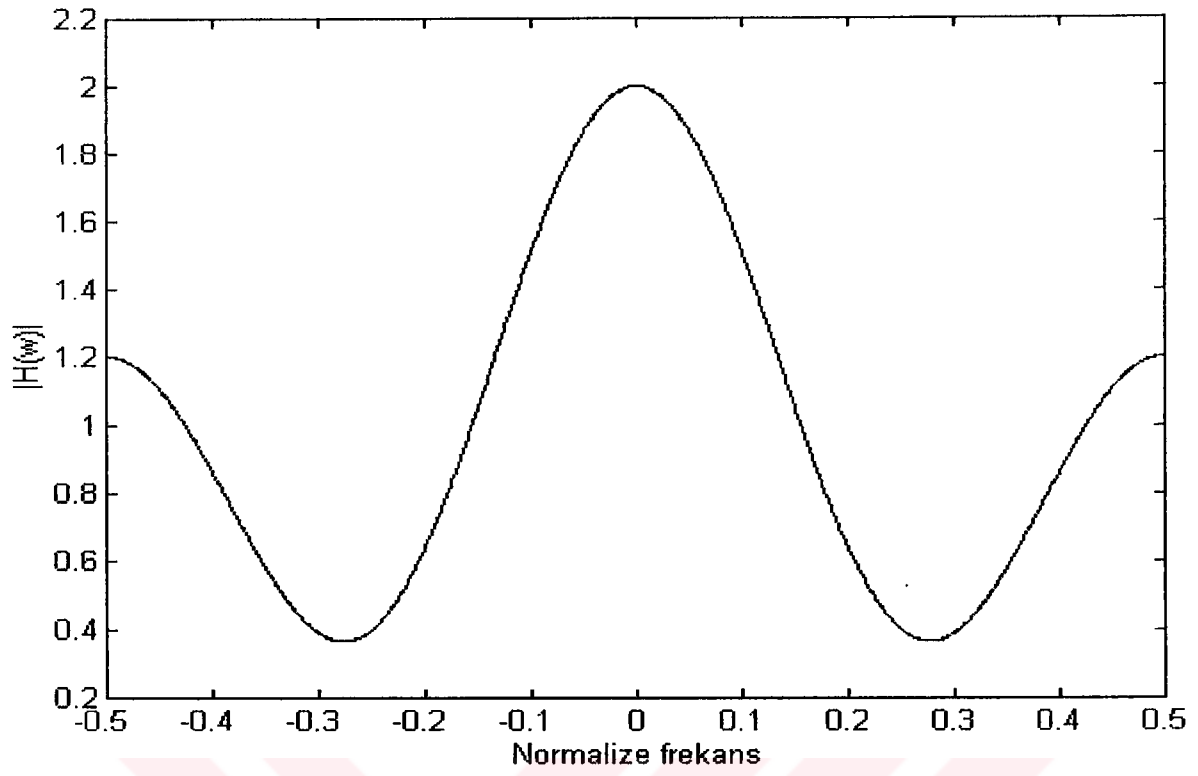
Şekil 6.42 (b)



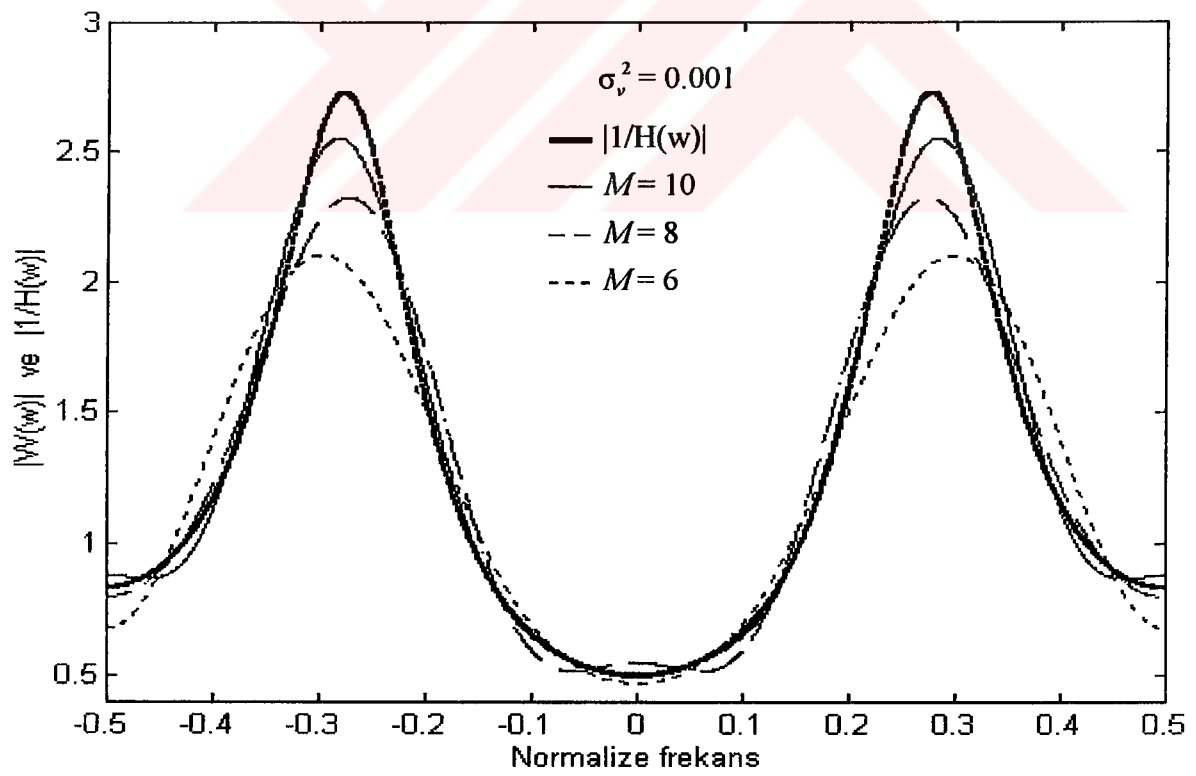
Şekil 6.42 (c)



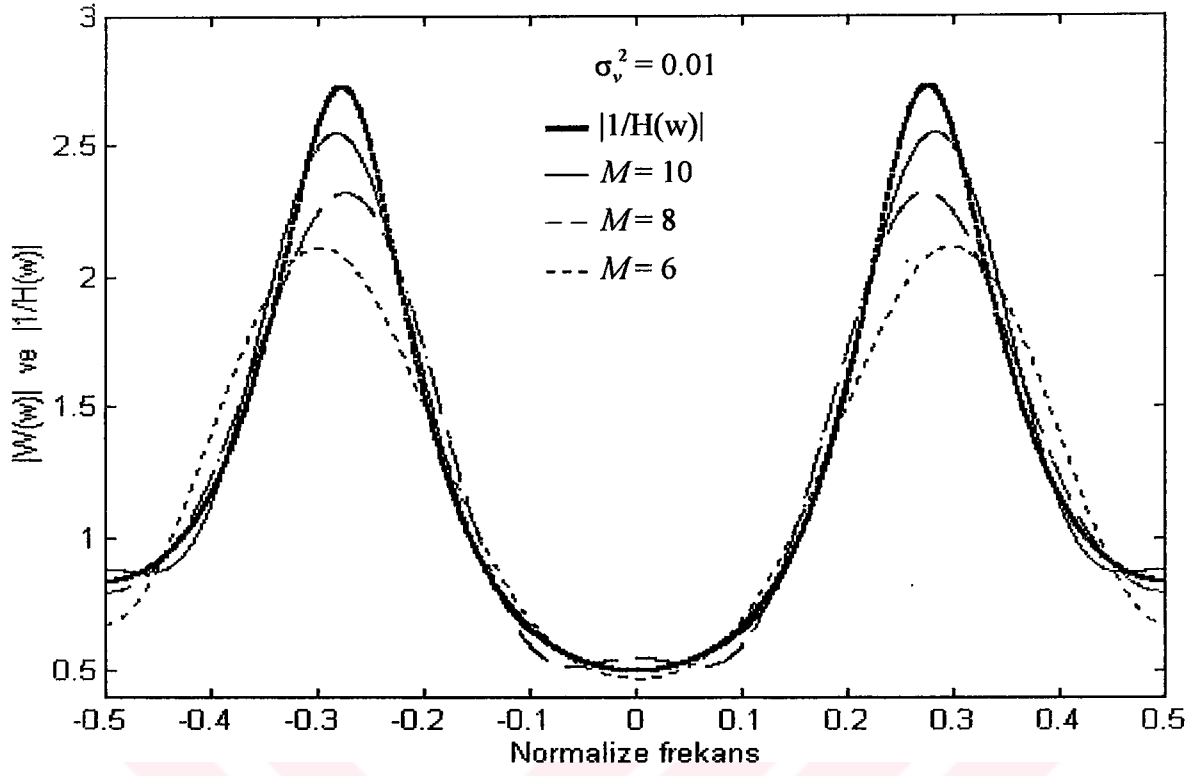
Şekil 6.42 (d)



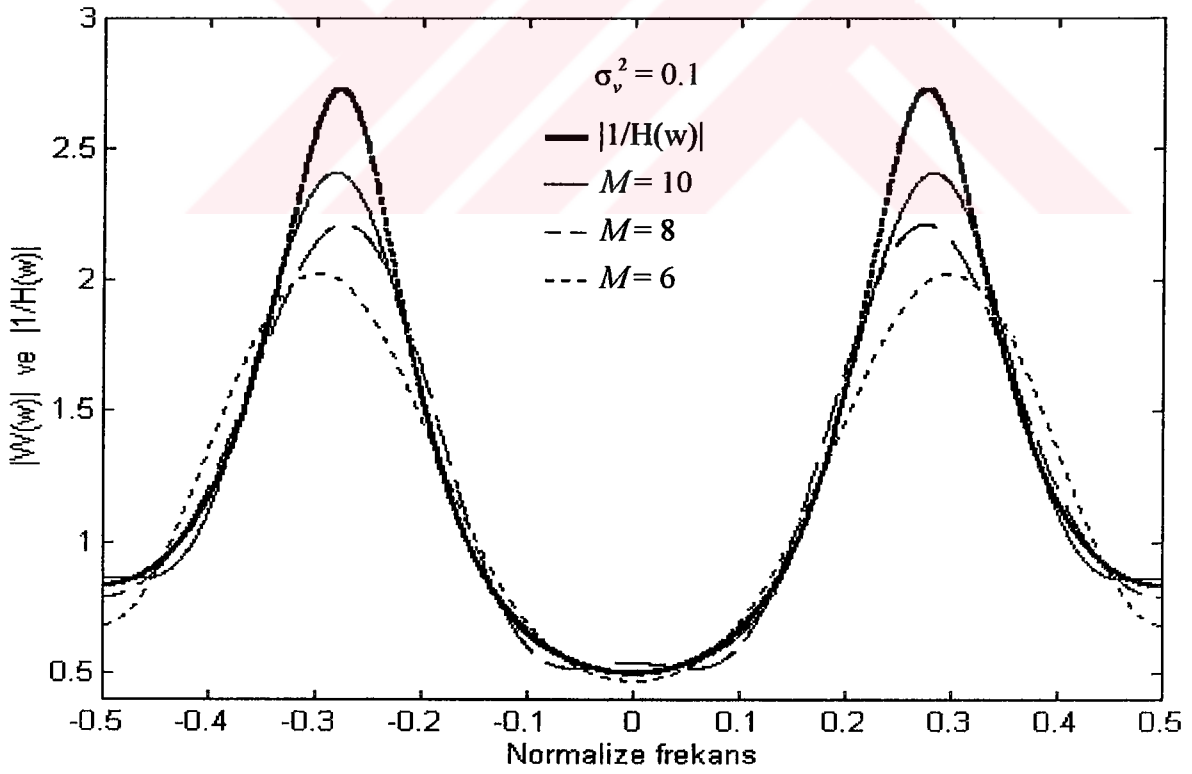
Şekil 6.43 (a)



Şekil 6.43 (b)



Şekil 6.43 (c)



Şekil 6.43 (d)

7. SONUÇLAR

Bu tezde, yinelemeli en küçük kareler kafes algoritması kullanarak, doğrusal kanalların uyarlamalı dengelenmesi incelenmiştir.

Kanal dengelemede amaç: verici kısımdan iletilen işaretle kanalın neden olduğu bozulmayı, gürültünün varlığına rağmen ortadan kaldırmaktır. Uyarlamalı dengeleyici bunu, kendisini kanalın tersine benzeterek başarmaya çalışır. Dengeleyicinin seçilen kanal modelleri karşısında sağladığı başarı bu çalışmada, öğrenme eğrisi ve transfer fonksiyonu genlik eğrisi bazında incelenmiştir.

Karesel ortalama hatanın iterasyon sayısına göre değişimini gösterdikleri için, öğrenme eğrilerinde başarı ölçüsü: yakınsama hızının yüksek olması ve karesel ortalama hatanın en aza inmesi olarak kabul edilir. Karesel ortalama hatanın en aza inmiş olmasının, uyarlamalı dengeleyicinin kendisini kanalın tersine benzetmiş olması anlamına gelip gelmediğini öğrenmek için, uyarlamalı dengeleyiciye ait transfer fonksiyonunun genlik eğrisi $|W(w)|$ ile kanalın transfer fonksiyonunun tersine ait genlik eğrisi $|1/H(w)|$ karşılaştırılmıştır. Bu karşılaştırma sonucunda görülmüştür ki: bir uyarlamalı dengeleyicinin karesel ortalama hata kalıcı hal değeri sıfır sayılabilecek kadar küçük ise, o dengeleyici kendisini, $|W(w)|=|1/H(w)|$ ideal durumunu sağlayabilecek kadar mükemmel bir şekilde kanalın tersine benzetebilmiş demektir.

Uyarlamalı kanal dengeleme deneylerinde, başta kanal modelinin kendisi olmak üzere: yükseltilmiş kosinüs kanal modeline ait W parametresi, beyaz gürültünün değışintisi σ_v^2 ve kafes filtre modül sayısı M değıştirilerek, RLSE algoritmasının başarımı incelenmiştir. Bu deneylerden elde edilen öğrenme eğrilerine dayanarak, RLSE algoritması için şu sonuçlar çıkarılabilir:

1. RLSE algoritmasının yakınsama hızı yüksektir. Özellikle, işaret-gürültü oranının büyük olması durumunda, M değeri karşı duyarlıdır ve M değeri ile ters orantılı bir şekilde değışmektedir. İşaret-gürültü oranının küçük olması durumunda ise, M değeri karşı olan duyarlılık ortadan kalkmakta ve RLSE algoritması, her M değeri için aynı sürede yakınsamaktadır.

2. RLSL algoritmasının yakınsama hızı, giriş işaretine ait ilişki matrisinin özdeğer yayılımında meydana gelen değişimlere karşı duyarsızdır. Diğer bir ifadeyle, W parametresindeki herhangi bir artış veya düşüş, yakınsama hızını etkilememektedir.
3. Bu ilk iki sonuca ve Bölüm 5.2.5.3'te RLS algoritması için anlatılanlara göre, RLS algoritması ile RLSL algoritmasının yakınsama özellikleri birbirinin aynıdır.
4. RLSL algoritmasının karesel ortalama hata kalıcı hal değeri, özdeğer yayılımına, gürültünün değişimine ve kafes filtre modül sayısına bağlı olarak değişmektedir. Özdeğer yayılımı (veya W parametresi) arttıkça, karesel ortalama hatanın kalıcı hal değeri artmakta ve dolayısıyla da, RLSL algoritmasının hata başarımı düşmektedir. Benzer durum, gürültünün değişimi için de geçerlidir. Buna göre, özdeğer yayılımında veya gürültünün değişiminde meydana gelen herhangi bir artış, RLSL algoritmasının hata başarımını kötüleştirmektedir. Kafes filtre modül sayısı M için ise, tam tersi söz konusudur. Özdeğer yayılımı ve gürültünün değişimi sabit tutulup M artırıldıkça, karesel ortalama hatanın kalıcı hal değeri azalmakta ve dolayısıyla da, RLSL algoritmasının hata başarımı yükselmektedir. Özdeğer yayılımı ve gürültünün değişimi ne olursa olsun, RLSL algoritmasının büyük M değeri ile elde ettiği hata başarımı, küçük M değeri ile elde ettiği hata başarımından daha iyi olmaktadır.

RLSL algoritması için çıkarılan bu sonuçlara göre, başta yakınsama hızları olmak üzere, RLS algoritması ile RLSL algoritması birbirlerine çok benzemektedir. Ancak, RLSL algoritmasının, kendisini RLS algoritmasından ayıran ve ona karşı üstün kılan iki önemli özelliği vardır:

1. Kafes filtreden kaynaklanan modüler yapısı

Enine filtre kullanımına dayalı olan algoritmalara (LMS ve RLS gibi) karşı, oldukça üstün bir özelliktir. Bu sayede, öngörü derecesi (kafes filtrenin modül sayısı) arttırılmak istendiğinde, önceki hesaplamaları etkilemeksizin sadece yeni modülün hesaplamaları yapılabilmektedir. Oysa enine filtrelerde, bu bir sorundur ve öngörü derecesinin her arttırılışında, önceki hesaplamaların yeniden yapılması gerekir; bu da, o filtreyi baştan tasarlamak demektir.

2. Hesaplama karmaşıklığının azlığı

Bölüm 5.3'ten hatırlanacağı gibi, hesaplama karmaşıklığının bir ölçüsü olarak, algoritmanın tek bir iterasyonu gerçekleştirmek için gerek duyduğu işlem sayısı kullanılmaktaydı.

Ayrıca, giriş verisinin gerçel değerli ve filtre öngörü derecesinin ise M 'e eşit olduğu varsayılarak, RLS algoritmasının hesaplama karmaşıklığının M^2 ile arttığı söylenmişti. RLS algoritmasının yüksek yakınsama hızına rağmen, hesaplama karmaşıklığının fazla olması, hesaplama karmaşıklığı daha az olan algoritmaların (hızlı algoritmalar) geliştirilmesine neden olmuştur. RLSE algoritması, bu amaçla geliştirilen hızlı algoritmalarından sadece bir tanesidir. Bu nedenle, diğer tüm hızlı algoritmalar gibi, RLSE algoritmasının hesaplama karmaşıklığı da M ile doğrusal olarak artar.



KAYNAKLAR

Box, G. E. P. ve Jenkins, G. M., (1976), Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco.

Carayannis, G., Manolakis, D. G. ve Kalouptsidis, N., (1983), "A Fast Sequential Algorithm for Least-Squares Filtering and Prediction", IEEE Trans. Acoust. Speech Signal Process., ASSP-31: 1394-1402.

Doob, L. J., (1953), Stochastic Processes, Wiley, New York.

Durbin, J., (1960), "The Fitting of Time Series Models", Rev. Int. Stat. Inst., 28: 233-244.

Friedlander, B., (1982), "Lattice Filters for Adaptive Processing", Proc. IEEE, 70: 829-867.

Gersho, A., (1969), "Adaptive Equalization of Highly Dispersive Channels for Data Transmission", Bell Syst. Tech. J., 48: 55-70.

Griffiths, L. J., (1977), "A Continuously Adaptive Filter Implemented as a Lattice Structure", Proc. IEEE Int. Conf. Acoust. Speech Signal Process., Hartford, Conn., 683-686.

Hayes, M. H., (1996), Statistical Digital Signal Processing and Modeling, John Wiley & Sons, New York.

Haykin, S., (1988), Digital Communications, Wiley, New York.

Haykin, S., (1991), Adaptive Filter Theory, Prentice-Hall, Englewood Cliffs, New Jersey.

Lee, D. T. L., Morf, M. ve Friedlander, B., (1981), "Recursive Least-Squares Ladder Estimation Algorithms", IEEE Trans. Circuits and Systems, CAS-28: 467-481.

Lev-Ari, H., Kailath, T. ve Cioffi, J., (1984), "Least-Squares Adaptive Lattice and Transversal Filters: A Unified Geometric Theory", IEEE Trans. Information Theory, IT-30: 222-236.

Levinson, N., (1947), "The Wiener RMS (Root-Mean-Square) Error Criterion in Filter Design and Prediction", J. Math Phys., 25: 261-278.

Ling, F., Manolakis, D. ve Proakis, J. G., (1985), "New Forms of LS Lattice Algorithms and an Analysis of Their Round-Off Error Characteristics", Proc. ICASSP 85, Tampa, Fla., 1739-1742.

Lucky, R. W., (1965), "Automatic Equalization for Digital Communication", Bell Syst. Tech. J., 44: 547-588.

Lucky, R. W., (1966), "Techniques for Adaptive Equalization of Digital Communication Systems", Bell Syst. Tech. J., 45: 255-286.

Lucky, R. W., Salz, J. ve Weldon, E. J., (1968), Principles of Data Communication, McGraw-Hill, New York.

Makhoul, J., (1978), "A Class of All-Zero Lattice Digital Filters: Properties and Applications", IEEE Trans. Acoust. Speech Signal Process., ASSP-26: 304-314.

Morf, M., (1974), "Fast Algorithms for Multivariable Systems", Ph. D. dissertation, Stanford University, Stanford, Calif.

Morf, M., Vieira, A. ve Lee, D. T., (1977), "Ladder Forms for Identification and Speech Processing", Proc. 1977 IEEE Conf. Decision and Control, New Orleans, Calif., 1074-1078.

Morf, M. ve Lee, D. T., (1978), "Recursive Least-Squares Ladder Forms for Fast Parameter Tracking", Proc. 1978 Conf. Decision and Control, San Diego, Calif., 1362-1367.

Mueller, M. S., (1981a), "Least-Squares Algorithms for Adaptive Equalizers", Bell Syst. Tech. J., 60: 1905-1925.

Mueller, M. S., (1981b), "On The Rapid Initial Convergence of Least-Squares Equalizers Adjustment Algorithms", Bell Syst. Tech. J., 60: 2345-2358.

Oppenheim, A. V. ve Schaffer, R. W., (1975), Digital Signal Processing, Prentice-Hall, Englewood Cliffs, New Jersey.

Porat, B. ve Kailath, T., (1983), "Normalized Lattice Algorithms for Least-Squares FIR System Identification", IEEE Trans. Acoust. Speech Signal Process., ASSP-31: 122-128.

Priestley, M. B., (1981), Spectral Analysis and Time Series, Academic Press, New York.

Proakis, J. G., (1975), "Advances in Equalization for Intersymbol Interference", in Advances in Communication Systems, Academic Press, New York, 4: 123-198.

Qureshi, S., (1982), "Adaptive Equalization", IEEE Commun. Mag., 20: 9-16.

Qureshi, S., (1985), "Adaptive Equalization", Proc. IEEE, 73: 1349-1387.

Satorius, E. H. ve Alexander, S. T., (1979), "Channel Equalization Using Adaptive Lattice Algorithms", IEEE Trans. Commun., COM-27: 899-905.

Satorius, E. H. ve Pack, J. D., (1981), "Application of Least-Squares Lattice Algorithms to Adaptive Equalization", IEEE Trans. Commun., COM-29: 136-142.

Strang, G., (1980), Linear Algebra and Its Applications, Academic Press, New York.

EKLER

- Ek 1 Sonraki Kestirim Hatalarına Dayalı RLSL Algoritmasını Kullanarak Uyarlamalı Kanal Dengeleme Deneyini Gerçekleyen Matlab Programı
- Ek 2 Ayırık Zamanlı Fourier Dönüşümünü Hesaplayan Matlab Programı
- Ek 3 Uyarlamalı Dengeleyicinin İmpuls Yanıtını $M = 6$ İçin Hesaplayan Matlab Programı
- Ek 4 Uyarlamalı Dengeleyicinin İmpuls Yanıtını $M = 8$ İçin Hesaplayan Matlab Programı
- Ek 5 Uyarlamalı Dengeleyicinin İmpuls Yanıtını $M = 10$ İçin Hesaplayan Matlab Programı
- Ek 6 $|W(w)|$, $|H(w)|$ ve $|1/H(w)|$ Eğrilerini $M = 6$ İçin Çizen Matlab Programı
- Ek 7 $|W(w)|$, $|H(w)|$ ve $|1/H(w)|$ Eğrilerini $M = 8$ İçin Çizen Matlab Programı
- Ek 8 $|W(w)|$, $|H(w)|$ ve $|1/H(w)|$ Eğrilerini $M = 10$ İçin Çizen Matlab Programı

Ek 1 Sonraki Kestirim Hatalarına Dayalı RLSL Algoritmasını Kullanarak Uyarlamalı Kanal Dengeleme Deneyini Gerçekleyen Matlab Programı

```

clear
format long

S = input('İterasyon sayısını giriniz:');
M = input('Modül sayısını giriniz:');
W = input('Kanal parametresini giriniz:');
Var = input('Gürültünün değışintisini giriniz:');
Den = input('Bağımsız deneme sayısını giriniz:');

clc

H(1:3) = (1/2) * (1 + cos(2 * pi * ((1:3)-2)/W));

su = max(size(H));
SS = S + (su-1);
if rem(M, 2) == 1
    D = (su+1)/2 + (M+1)/2;
else
    D = (su+1)/2 + M/2;
end

J = zeros(1, SS);
SK = zeros(M+1, SS-1);
SKF = zeros(M, SS);
SKB = zeros(M, SS);

for b = 1:Den

    U = zeros(1, S) - 1;
    L = rand(1, S);
    p = find(L<0.5);
    U(p) = ones(size(p));

    X = conv(U, H);

    R1 = rand(1, SS/2);
    R2 = rand(1, SS/2);
    S1 = sqrt(-2 * log(R1)).* sin(2 * pi * R2);
    S2 = sqrt(-2 * log(R2)).* cos(2 * pi * R1);
    N(1, 1:2:SS-1) = Var * S1;
    N(1, 2:2:SS) = Var * S2;

    Y = X + N;

```

```

DEL(1:M, 1) = zeros(M, 1);
BUF(1:M, 1) = 0.004 * ones(M, 1);
for n = 2:SS+1
    BUF(1, n) = BUF(1, n-1) + Y(n-1)^2 ;
end
BUB = BUF;
NU(1, 1:SS+1) = ones(1, SS+1);
KF(1:M, 1) = zeros(M, 1);
KB(1:M, 1) = zeros(M, 1);
F(1, 1) = 0;
F(1, 2:SS+1) = Y;
B(1, 1) = 0;
B(1, 2:SS+1) = Y;

for n = 2:SS+1
    for m = 1:M
        DEL(m, n) = DEL(m, n-1) + (B(m, n-1) * F(m, n))/NU(m, n-1);
        KF(m, n) = - DEL(m, n)/BUB(m, n-1);
        KB(m, n) = - DEL(m, n)/BUF(m, n);
        F(m+1, n) = F(m, n) + KF(m, n) * B(m, n-1);
        B(m+1, n) = B(m, n-1) + KB(m, n) * F(m, n);
        BUF(m+1, n) = BUF(m, n) - DEL(m, n)^2/BUB(m, n-1);
        BUB(m+1, n) = BUB(m, n-1) - DEL(m, n)^2/BUF(m, n);
        NU(m+1, n-1) = NU(m, n-1) - B(m, n-1)^2/BUB(m, n-1);
    end
end

SKF = SKF + KF(1:M, 2:SS+1);
SKB = SKB + KB(1:M, 2:SS+1);

RO(1:M+1, 1) = zeros(M+1, 1);

E(1, 1:D) = zeros(1, D);
E(1, D+1:SS) = U(1, 1:SS-D);
K(1:M+1, 1) = zeros(M+1, 1);

for n = 2:SS
    for m = 1:M+1
        RO(m, n) = RO(m, n-1) + (B(m, n)/NU(m, n)) * E(m, n);
        K(m, n) = RO(m, n)/BUB(m, n);
        E(m+1, n) = E(m, n) - K(m, n) * B(m, n);
    end
end

SK = SK + K(1:M+1, 2:SS);

J = J + (E(M+2, 1:SS)./NU(M+1, 1:SS)).^2;

```

```

UK(1, 1:D) = zeros(1, D);
UK(1, D+1:SS) = U(1, 1:SS-D) - (E(M+2, D+1:SS)./NU(M+1, D+1:SS));
for n = 1:SS-D
    fprintf('U(%g)=.....    UK(%g)=.....    \n', n, n+D);
    fprintf('          %f          %f \n', U(n), UK(n+D));
end

end

J = (1/Den) * J;
SK = (1/Den) * SK;
SKF = (1/Den) * SKF;
SKB = (1/Den) * SKB;

whitebg, figure, axes('position', [0.1, 0.1, 0.877, 0.872])
semilogy(1:S, J(1:S), 'k')
xlabel('İterasyon Sayısı'), ylabel('Karesel Ortalama Hata')

```

Uyarı (1): Bu programın çalışması, öğrenme eğrisinin çizdirilmesi ile sona erer. Ancak, programda oluşturulan bazı değişkenler (H, SK, SKF ve SKB) aynı zamanda, $H(w)$ ve $W(w)$ transfer fonksiyonlarının hesaplanmasında da kullanılmaktadırlar. Öyleyse bu değişkenler, 'save dosyaadi H SK SKF SKB' komutu ile oluşturulacak 'dosyaadi . mat' isimli bir dosyaya kaydedilmelidir. Bu sayede, başka programlarda kullanılmaları gerektiği zaman, 'load dosyaadi' komutu ile hafızaya tekrar yüklenebilirler.

Uyarı (2): Yukarıda, $U(n)$ istenen yanıtının kestirimi olan $UK(n)$, önceki kestirim hatası kullanarak hesaplanmıştır. $UK(n)$ 'i sonraki kestirim hatası kullanarak hesaplamak için,

$$UK(1, D+1:SS) = U(1, 1:SS-D) - (E(M+2, D+1:SS)./NU(M+1, D+1:SS));$$

satırını

$$UK(1, D+1:SS) = U(1, 1:SS-D) - E(M+2, D+1:SS);$$

haline getirmek gerekir. Aslında,

$$UK(1, 1:D) = \text{zeros}(1, D);$$

$$UK(1, D+1:SS) = U(1, 1:SS-D) - (E(M+2, D+1:SS)./NU(M+1, D+1:SS));$$

for n = 1:SS-D

$$\text{fprintf}('U(\%g)=..... UK(\%g)=..... \n', n, n+D);$$

$$\text{fprintf}(' \%f \%f \n', U(n), UK(n+D));$$

end

satırları, uyarlamalı dengeleyicinin istenen yanıtı ne kadar başarılı kestirebildiğini görmek için kullanılmıştır; istenilirse programdan çıkarılabilirler.

Ek 2 Ayrık Zamanlı Fourier Dönüşümünü Hesaplayan Matlab Programı

```
function [H, w] = dtft(h, N)

% DTFT: birim daire üzerinde eşit olarak dağılmış N adet
%       frekansa ilişkin Ayrık Fourier dönüşümünü
%       hesaplar. Kullanımı [H, w] = dtft(h, N) şeklindedir.
% h      : L şeklinde sonlu uzunluğu olan giriş vektörüdür.
% N      :  $N \geq L$  olmak şartıyla  $[-\pi, \pi)$  aralığındaki
%       frekans sayısıdır.
% H      : DTFT değerleridir (kompleks).
% w      : DTFT hesabında kullanılan frekanslardır.

w = 0:(2 * pi)/N:(2 * pi * (N-1))/N;
[m, sat] = size(h);
[p, sut] = size(w);

for k = 1:sut
    for n = 1:sat
        EXP(n, k) = h(n) * exp(- w(k) * n * i);
    end
end
H = sum (EXP);

mid = ceil(N/2) + 1;
w(mid:N) = w(mid:N) - 2 * pi;    %  $[\pi, 2\pi)$  aralığı  $[-\pi, 0)$  aralığına taşınıyor.
w = fftshift(w);                %  $(0, \pi]$  ve  $[-\pi, 0)$  aralığı, birbiriyle yer değiştiriyor.
H = fftshift(H);
```

Uyarı: Ek 6, Ek 7 ve Ek 8’de verilen çizim programlarında kullanılacağı için, bu program ‘dtft . m’ şeklinde kaydedilmelidir. Bu sayede, başka programlarda kullanılması gerektiği zaman ‘[H, w] = dtft(h, N)’ komutunu yazmak yeterli olacaktır. Burada h ve N, dtft fonksiyonuna dışardan girilen değerler; H ve w ise, dtft fonksiyonun sonuç olarak ürettiği değerler olmaktadır.

Ek 3 Uyarlamalı Dengeleyicinin İmpuls Yanıtını $M = 6$ İçin Hesaplayan Matlab Programı

$S = \max(\text{size}(\text{SK}));$

$G0 = \text{SK}(1, S); \quad G1 = \text{SK}(2, S); \quad G2 = \text{SK}(3, S); \quad G3 = \text{SK}(4, S);$
 $G4 = \text{SK}(5, S); \quad G5 = \text{SK}(6, S); \quad G6 = \text{SK}(7, S);$

$\text{KF1} = \text{SKF}(1, S); \quad \text{KF2} = \text{SKF}(2, S); \quad \text{KF3} = \text{SKF}(3, S);$
 $\text{KF4} = \text{SKF}(4, S); \quad \text{KF5} = \text{SKF}(5, S); \quad \text{KF6} = \text{SKF}(6, S);$

$\text{KB1} = \text{SKB}(1, S); \quad \text{KB2} = \text{SKB}(2, S); \quad \text{KB3} = \text{SKB}(3, S);$
 $\text{KB4} = \text{SKB}(4, S); \quad \text{KB5} = \text{SKB}(5, S); \quad \text{KB6} = \text{SKB}(6, S);$

$A1 = \text{KF1} + (\text{KF2} * \text{KB1});$
 $B1 = \text{KB1} + (\text{KB2} * \text{KF1});$
 $C1 = A1 + (\text{KF3} * \text{KB2});$
 $D1 = \text{KF2} + (\text{KF3} * B1);$
 $E1 = \text{KB2} + (\text{KB3} * A1);$
 $F1 = B1 + (\text{KB3} * \text{KF2});$
 $H1 = C1 + (\text{KF4} * \text{KB3});$
 $I1 = D1 + (\text{KF4} * E1);$
 $J1 = \text{KF3} + (\text{KF4} * F1);$
 $K1 = \text{KB3} + (\text{KB4} * C1);$
 $L1 = E1 + (\text{KB4} * D1);$
 $M1 = F1 + (\text{KB4} * \text{KF3});$
 $N1 = H1 + (\text{KF5} * \text{KB4});$
 $O1 = I1 + (\text{KF5} * K1);$
 $P1 = J1 + (\text{KF5} * L1);$
 $Q1 = \text{KF4} + (\text{KF5} * M1);$
 $R1 = \text{KB4} + (\text{KB5} * H1);$
 $S1 = K1 + (\text{KB5} * I1);$
 $T1 = L1 + (\text{KB5} * J1);$
 $U1 = M1 + (\text{KB5} * \text{KF4});$

$W(1) = G0 + (G1 * \text{KB1}) + (G2 * \text{KB2}) + (G3 * \text{KB3}) + (G4 * \text{KB4}) + (G5 * \text{KB5}) + \dots$
 $(G6 * \text{KB6});$

$W(2) = G1 + (G2 * B1) + (G3 * E1) + (G4 * K1) + (G5 * R1) + (G6 * (\text{KB5} + \text{KB6} * N1));$

$W(3) = G2 + (G3 * F1) + (G4 * L1) + (G5 * S1) + (G6 * (R1 + \text{KB6} * O1));$

$W(4) = G3 + (G4 * M1) + (G5 * T1) + (G6 * (S1 + \text{KB6} * P1));$

$W(5) = G4 + (G5 * U1) + (G6 * (T1 + \text{KB6} * Q1));$

$W(6) = G5 + (G6 * (U1 + \text{KB6} * \text{KF5}));$

$W(7) = G6;$

Uyarı: Bu program, 'modelw6 . m' şeklinde kaydedilmelidir. Bu sayede, başka programlarda kullanılması gerektiği zaman 'modelw6' komutunu yazmak yeterli olacaktır.

Ek 4 Uyarlamalı Dengeleyicinin İmpuls Yanıtını $M = 8$ İçin Hesaplayan Matlab Programı

$S = \max(\text{size}(\text{SK}));$

$G0 = \text{SK}(1, S); \quad G1 = \text{SK}(2, S); \quad G2 = \text{SK}(3, S); \quad G3 = \text{SK}(4, S); \quad G4 = \text{SK}(5, S);$
 $G5 = \text{SK}(6, S); \quad G6 = \text{SK}(7, S); \quad G7 = \text{SK}(8, S); \quad G8 = \text{SK}(9, S);$

$\text{KF1} = \text{SKF}(1, S); \quad \text{KF2} = \text{SKF}(2, S); \quad \text{KF3} = \text{SKF}(3, S); \quad \text{KF4} = \text{SKF}(4, S);$
 $\text{KF5} = \text{SKF}(5, S); \quad \text{KF6} = \text{SKF}(6, S); \quad \text{KF7} = \text{SKF}(7, S); \quad \text{KF8} = \text{SKF}(8, S);$

$\text{KB1} = \text{SKB}(1, S); \quad \text{KB2} = \text{SKB}(2, S); \quad \text{KB3} = \text{SKB}(3, S); \quad \text{KB4} = \text{SKB}(4, S);$
 $\text{KB5} = \text{SKB}(5, S); \quad \text{KB6} = \text{SKB}(6, S); \quad \text{KB7} = \text{SKB}(7, S); \quad \text{KB8} = \text{SKB}(8, S);$

$A1 = \text{KF1} + (\text{KF2} * \text{KB1});$
 $B1 = \text{KB1} + (\text{KB2} * \text{KF1});$
 $C1 = A1 + (\text{KF3} * \text{KB2});$
 $D1 = \text{KF2} + (\text{KF3} * B1);$
 $E1 = \text{KB2} + (\text{KB3} * A1);$
 $F1 = B1 + (\text{KB3} * \text{KF2});$
 $H1 = C1 + (\text{KF4} * \text{KB3});$
 $I1 = D1 + (\text{KF4} * E1);$
 $J1 = \text{KF3} + (\text{KF4} * F1);$
 $K1 = \text{KB3} + (\text{KB4} * C1);$
 $L1 = E1 + (\text{KB4} * D1);$
 $M1 = F1 + (\text{KB4} * \text{KF3});$
 $N1 = H1 + (\text{KF5} * \text{KB4});$
 $O1 = I1 + (\text{KF5} * K1);$
 $P1 = J1 + (\text{KF5} * L1);$
 $Q1 = \text{KF4} + (\text{KF5} * M1);$
 $R1 = \text{KB4} + (\text{KB5} * H1);$
 $S1 = K1 + (\text{KB5} * I1);$
 $T1 = L1 + (\text{KB5} * J1);$
 $U1 = M1 + (\text{KB5} * \text{KF4});$
 $A2 = N1 + (\text{KF6} * \text{KB5});$
 $B2 = O1 + (\text{KF6} * R1);$
 $C2 = P1 + (\text{KF6} * S1);$
 $D2 = Q1 + (\text{KF6} * T1);$
 $E2 = \text{KF5} + (\text{KF6} * U1);$
 $F2 = \text{KB5} + (\text{KB6} * N1);$
 $H2 = R1 + (\text{KB6} * O1);$
 $I2 = S1 + (\text{KB6} * P1);$
 $J2 = T1 + (\text{KB6} * Q1);$
 $K2 = U1 + (\text{KB6} * \text{KF5});$
 $L2 = A2 + (\text{KF7} * \text{KB6});$
 $M2 = B2 + (\text{KF7} * F2);$
 $N2 = C2 + (\text{KF7} * H2);$

$$\begin{aligned}
O2 &= D2 + (KF7 * I2); \\
P2 &= E2 + (KF7 * J2); \\
Q2 &= KF6 + (KF7 * K2); \\
R2 &= KB6 + (KB7 * A2); \\
S2 &= F2 + (KB7 * B2); \\
T2 &= H2 + (KB7 * C2); \\
U2 &= I2 + (KB7 * D2); \\
A3 &= J2 + (KB7 * E2); \\
B3 &= K2 + (KB7 * KF6);
\end{aligned}$$

$$\begin{aligned}
W(1) &= G0 + (G1 * KB1) + (G2 * KB2) + (G3 * KB3) + (G4 * KB4) + (G5 * KB5) + \dots \\
&\quad (G6 * KB6) + (G7 * KB7) + (G8 * KB8); \\
W(2) &= G1 + (G2 * B1) + (G3 * E1) + (G4 * K1) + (G5 * R1) + (G6 * F2) + \dots \\
&\quad (G7 * R2) + (G8 * (KB7 + KB8 * L2)); \\
W(3) &= G2 + (G3 * F1) + (G4 * L1) + (G5 * S1) + (G6 * H2) + (G7 * S2) + \dots \\
&\quad (G8 * (R2 + KB8 * M2)); \\
W(4) &= G3 + (G4 * M1) + (G5 * T1) + (G6 * I2) + (G7 * T2) + (G8 * (S2 + KB8 * N2)); \\
W(5) &= G4 + (G5 * U1) + (G6 * J2) + (G7 * U2) + (G8 * (T2 + KB8 * O2)); \\
W(6) &= G5 + (G6 * K2) + (G7 * A3) + (G8 * (U2 + KB8 * P2)); \\
W(7) &= G6 + (G7 * B3) + (G8 * (A3 + KB8 * Q2)); \\
W(8) &= G7 + (G8 * (B3 + KB8 * KF7)); \\
W(9) &= G8;
\end{aligned}$$

Uyarı: Bu program, 'modelw8 . m' şeklinde kaydedilmelidir. Bu sayede, başka programlarda kullanılması gerektiği zaman 'modelw8' komutunu yazmak yeterli olacaktır.

Ek 5 Uyarlamalı Dengeleyicinin İmpuls Yanıtını $M = 10$ İçin Hesaplayan Matlab Programı

$S = \max(\text{size}(\text{SK}));$

$G0 = \text{SK}(1, S); G1 = \text{SK}(2, S); G2 = \text{SK}(3, S); G3 = \text{SK}(4, S); G4 = \text{SK}(5, S);$
 $G5 = \text{SK}(6, S); G6 = \text{SK}(7, S); G7 = \text{SK}(8, S); G8 = \text{SK}(9, S); G9 = \text{SK}(10, S);$
 $G10 = \text{SK}(11, S);$

$\text{KF1} = \text{SKF}(1, S); \text{KF2} = \text{SKF}(2, S); \text{KF3} = \text{SKF}(3, S); \text{KF4} = \text{SKF}(4, S);$
 $\text{KF5} = \text{SKF}(5, S); \text{KF6} = \text{SKF}(6, S); \text{KF7} = \text{SKF}(7, S); \text{KF8} = \text{SKF}(8, S);$
 $\text{KF9} = \text{SKF}(9, S); \text{KF10} = \text{SKF}(10, S);$

$\text{KB1} = \text{SKB}(1, S); \text{KB2} = \text{SKB}(2, S); \text{KB3} = \text{SKB}(3, S); \text{KB4} = \text{SKB}(4, S);$
 $\text{KB5} = \text{SKB}(5, S); \text{KB6} = \text{SKB}(6, S); \text{KB7} = \text{SKB}(7, S); \text{KB8} = \text{SKB}(8, S);$
 $\text{KB9} = \text{SKB}(9, S); \text{KB10} = \text{SKB}(10, S);$

$A1 = \text{KF1} + (\text{KF2} * \text{KB1});$
 $B1 = \text{KB1} + (\text{KB2} * \text{KF1});$
 $C1 = A1 + (\text{KF3} * \text{KB2});$
 $D1 = \text{KF2} + (\text{KF3} * B1);$
 $E1 = \text{KB2} + (\text{KB3} * A1);$
 $F1 = B1 + (\text{KB3} * \text{KF2});$
 $H1 = C1 + (\text{KF4} * \text{KB3});$
 $I1 = D1 + (\text{KF4} * E1);$
 $J1 = \text{KF3} + (\text{KF4} * F1);$
 $K1 = \text{KB3} + (\text{KB4} * C1);$
 $L1 = E1 + (\text{KB4} * D1);$
 $M1 = F1 + (\text{KB4} * \text{KF3});$
 $N1 = H1 + (\text{KF5} * \text{KB4});$
 $O1 = I1 + (\text{KF5} * K1);$
 $P1 = J1 + (\text{KF5} * L1);$
 $Q1 = \text{KF4} + (\text{KF5} * M1);$
 $R1 = \text{KB4} + (\text{KB5} * H1);$
 $S1 = K1 + (\text{KB5} * I1);$
 $T1 = L1 + (\text{KB5} * J1);$
 $U1 = M1 + (\text{KB5} * \text{KF4});$
 $A2 = N1 + (\text{KF6} * \text{KB5});$
 $B2 = O1 + (\text{KF6} * R1);$
 $C2 = P1 + (\text{KF6} * S1);$
 $D2 = Q1 + (\text{KF6} * T1);$
 $E2 = \text{KF5} + (\text{KF6} * U1);$
 $F2 = \text{KB5} + (\text{KB6} * N1);$
 $H2 = R1 + (\text{KB6} * O1);$
 $I2 = S1 + (\text{KB6} * P1);$
 $J2 = T1 + (\text{KB6} * Q1);$
 $K2 = U1 + (\text{KB6} * \text{KF5});$

$$\begin{aligned}
L2 &= A2 + (KF7 * KB6); \\
M2 &= B2 + (KF7 * F2); \\
N2 &= C2 + (KF7 * H2); \\
O2 &= D2 + (KF7 * I2); \\
P2 &= E2 + (KF7 * J2); \\
Q2 &= KF6 + (KF7 * K2); \\
R2 &= KB6 + (KB7 * A2); \\
S2 &= F2 + (KB7 * B2); \\
T2 &= H2 + (KB7 * C2); \\
U2 &= I2 + (KB7 * D2); \\
A3 &= J2 + (KB7 * E2); \\
B3 &= K2 + (KB7 * KF6); \\
C3 &= L2 + (KF8 * KB7); \\
D3 &= M2 + (KF8 * R2); \\
E3 &= N2 + (KF8 * S2); \\
F3 &= O2 + (KF8 * T2); \\
H3 &= P2 + (KF8 * U2); \\
I3 &= Q2 + (KF8 * A3); \\
J3 &= KF7 + (KF8 * B3); \\
K3 &= KB7 + (KB8 * L2); \\
L3 &= R2 + (KB8 * M2); \\
M3 &= S2 + (KB8 * N2); \\
N3 &= T2 + (KB8 * O2); \\
O3 &= U2 + (KB8 * P2); \\
P3 &= A3 + (KB8 * Q2); \\
Q3 &= B3 + (KB8 * KF7); \\
R3 &= C3 + (KF9 * KB8); \\
S3 &= D3 + (KF9 * K3); \\
T3 &= E3 + (KF9 * L3); \\
U3 &= F3 + (KF9 * M3); \\
A4 &= H3 + (KF9 * N3); \\
B4 &= I3 + (KF9 * O3); \\
C4 &= J3 + (KF9 * P3); \\
D4 &= KF8 + (KF9 * Q3); \\
E4 &= KB8 + (KB9 * C3); \\
F4 &= K3 + (KB9 * D3); \\
H4 &= L3 + (KB9 * E3); \\
I4 &= M3 + (KB9 * F3); \\
J4 &= N3 + (KB9 * H3); \\
K4 &= O3 + (KB9 * I3); \\
L4 &= P3 + (KB9 * J3); \\
M4 &= Q3 + (KB9 * KF8);
\end{aligned}$$

$$\begin{aligned}
W(1) &= G0 + (G1 * KB1) + (G2 * KB2) + (G3 * KB3) + (G4 * KB4) + (G5 * KB5) + ... \\
&\quad (G6 * KB6) + (G7 * KB7) + (G8 * KB8) + (G9 * KB9) + (G10 * KB10); \\
W(2) &= G1 + (G2 * B1) + (G3 * E1) + (G4 * K1) + (G5 * R1) + (G6 * F2) + (G7 * R2) + ... \\
&\quad (G8 * K3) + (G9 * E4) + (G10 * (KB9 + KB10 * R3)); \\
W(3) &= G2 + (G3 * F1) + (G4 * L1) + (G5 * S1) + (G6 * H2) + (G7 * S2) + (G8 * L3) + ... \\
&\quad (G9 * F4) + (G10 * (E4 + KB10 * S3));
\end{aligned}$$

$$W(4) = G3 + (G4 * M1) + (G5 * T1) + (G6 * I2) + (G7 * T2) + (G8 * M3) + (G9 * H4) + \dots \\ (G10 * (F4 + KB10 * T3));$$

$$W(5) = G4 + (G5 * U1) + (G6 * J2) + (G7 * U2) + (G8 * N3) + (G9 * I4) + \dots \\ (G10 * (H4 + KB10 * U3));$$

$$W(6) = G5 + (G6 * K2) + (G7 * A3) + (G8 * O3) + (G9 * J4) + (G10 * (I4 + KB10 * A4));$$

$$W(7) = G6 + (G7 * B3) + (G8 * P3) + (G9 * K4) + (G10 * (J4 + KB10 * B4));$$

$$W(8) = G7 + (G8 * Q3) + (G9 * L4) + (G10 * (K4 + KB10 * C4));$$

$$W(9) = G8 + (G9 * M4) + (G10 * (L4 + KB10 * D4));$$

$$W(10) = G9 + (G10 * (M4 + KB10 * KF9));$$

$$W(11) = G10;$$

Uyarı: Bu program, 'modelw10 . m' şeklinde kaydedilmelidir. Bu sayede, başka programlarda kullanılması gerektiği zaman 'modelw10' komutunu yazmak yeterli olacaktır.



Ek 6 $|W(w)|$, $|H(w)|$ ve $|1/H(w)|$ Eğrilerini $M = 6$ İçin Çizen Matlab Programı

```

clear
format long
clc
load dosyaadi           % Ek 1'deki programın sonunda oluşturulan 'dosyaadi . mat'
                        % dosyasındaki H, SK, SKF ve SKB değişkenleri yeniden
                        % hafızaya yükleniyor.

modelw6
save impuls6 H W        %  $h(n)$  ve  $w(n)$  impuls yanıtları, 'impuls6 . mat' dosyasına
                        % kaydediliyor.

clear
load impuls6

N = 1000;
[WF, w] = dtft(W, N);
[HF, w] = dtft(H, N);
THF = 1./HF;

whitebg, figure, axes('position', [0.1, 0.1, 0.88, 0.88])
plot((w/2)/pi, abs(WF), 'k'), xlabel('Normalize frekans'), ylabel('|W(w)|')
figure, axes('position', [0.1, 0.1, 0.88, 0.88])
plot((w/2)/pi, abs(HF), 'k'), xlabel('Normalize frekans'), ylabel('|H(w)|')
figure, axes('position', [0.1, 0.1, 0.88, 0.88])
plot((w/2)/pi, abs(THF), 'k'), xlabel('Normalize frekans'), ylabel('|1/H(w)|')

```

Ek 7 $|W(w)|$, $|H(w)|$ ve $|1/H(w)|$ Eğrilerini $M = 8$ İçin Çizen Matlab Programı

```

clear
format long
clc
load dosyaadi           % Ek 1'deki programın sonunda oluşturulan 'dosyaadi . mat'
                        % dosyasındaki H, SK, SKF ve SKB değişkenleri yeniden
                        % hafızaya yükleniyor.

modelw8
save impuls8 H W        %  $h(n)$  ve  $w(n)$  impuls yanıtları, 'impuls8 . mat' dosyasına
                        % kaydediliyor.

clear
load impuls8

N = 1000;
[WF, w] = dtft(W, N);
[HF, w] = dtft(H, N);
THF = 1./HF;

whitebg, figure, axes('position', [0.1, 0.1, 0.88, 0.88])
plot((w/2)/pi, abs(WF), 'k'), xlabel('Normalize frekans'), ylabel('|W(w)|')
figure, axes('position', [0.1, 0.1, 0.88, 0.88])
plot((w/2)/pi, abs(HF), 'k'), xlabel('Normalize frekans'), ylabel('|H(w)|')
figure, axes('position', [0.1, 0.1, 0.88, 0.88])
plot((w/2)/pi, abs(THF), 'k'), xlabel('Normalize frekans'), ylabel('|1/H(w)|')

```

Ek 8 $|W(w)|$, $|H(w)|$ ve $|1/H(w)|$ Eğrilerini $M = 10$ İçin Çizen Matlab Programı

```

clear
format long
clc
load dosyaadi           % Ek 1'deki programın sonunda oluşturulan 'dosyaadi . mat'
                        % dosyasındaki H, SK, SKF ve SKB değişkenleri yeniden
                        % hafızaya yükleniyor.

modelw6
save impuls10 H W       %  $h(n)$  ve  $w(n)$  impuls yanıtları, 'impuls10 . mat' dosyasına
                        % kaydediliyor.

clear
load impuls10

N = 1000;
[WF, w] = dtft(W, N);
[HF, w] = dtft(H, N);
THF = 1./HF;

whitebg, figure, axes('position', [0.1, 0.1, 0.88, 0.88])
plot((w/2)/pi, abs(WF), 'k'), xlabel('Normalize frekans'), ylabel('|W(w)|')
figure, axes('position', [0.1, 0.1, 0.88, 0.88])
plot((w/2)/pi, abs(HF), 'k'), xlabel('Normalize frekans'), ylabel('|H(w)|')
figure, axes('position', [0.1, 0.1, 0.88, 0.88])
plot((w/2)/pi, abs(THF), 'k'), xlabel('Normalize frekans'), ylabel('|1/H(w)|')

```

ÖZGEÇMİŞ

Doğum tarihi	30.08.1972	
Doğum yeri	Konya / Ereğli	
Lise	1986-1989	50. Yıl Tahran Lisesi
Lisans	1990-1994	Yıldız Teknik Üniversitesi Mühendislik Fak. Elektronik ve Haberleşme Mühendisliği Bölümü
	1995-1996	Yıldız Teknik Üniversitesi Yabancı Diller Bölümü Yabancı Dil Geliştirme Sınıfı (İngilizce)

