

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MUHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK ELEKTRONİK BİLİM DALI



KOÇAN TABANLI UYARLANABİLİR KANAL DENKLEŞTİRME

YÜKSEK LİSANS TEZİ

Hazırlayan

Noor J. Ahmed

Tez Danışman

Prof. Dr. Osman Nuri Uçan

İstanbul -2014



T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ
MÜDÜRLÜĞÜ'NE

Tez Onay Belgesi

Enstitümüz Elektrik-Elektronik Mühendisliği Anabilim Dalı, Elektrik-Elektronik Mühendisliği (Tezli) Yüksek Lisans Programı Y1213.100006 numaralı öğrencisi **Noor Jamal AHMED**' in "KOÇAN TABANLI UYARLANABİLİR KANAL DENKLEŞTİRME" adlı tez çalışması Enstitümüz Yönetim Kurulunun 18.07.2014 tarih ve 2014/12 sayılı kararıyla oluşturulan jüri tarafından **oybirliği / oyçokluğu** ile Yüksek Lisans Tezi olarak **kabul** edilmiştir.

Öğretim Üyesi Adı Soyadı

İmzası

Tez Savunma Tarihi : 14.10.2014

- 1) Tez Danışmanı : Prof. Dr. Osman Nuri UÇAN
2) Jüri Üyesi : Prof. Dr. Hasan Hüseyin BALIK
3) Jüri Üyesi : Prof. Dr. Serhat ŞEKER

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

ÖNSÖZ

Bu çalışma İstanbul Aydın Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Bölümünde hazırladığım koçan tabanlı uyarlanabilir kanal denkleştirme üzerine olan yüksek lisans tezimin sonucu olarak tamamlanmıştır.

Yüksek lisans öğrenimim sırasında ve tez çalışmalarım boyunca gösterdiği her türlü destek ve paylaştığı görüşlerinden dolayı çok değerli hocam Prof. Dr. Osman Nuri Uçan'a en içten dileklerle teşekkür ederim. Ayrıca hayatımın her aşamasında maddi ve manevi desteklerinden dolayı aileme teşekkür ederim.

Eğitimim boyunca çalışmamı destekleyen İstanbul Aydın Üniversitesi ve Elektrik-Elektronik Mühendisliği Bölümüne teşekkürü borç bilirim. Çalışmamın tüm ilgililere yararlı olmasını dilerim.

İÇİNDEKİLER

	Sayfa No.
ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİL LİSTESİ	iv
TABLO LİSTESİ	v
1.GİRİŞ	1
2.UYARLANABİLİR KANAL DENKLEŞTİRME	8
2.1.HABERLEŞME KANAL PARAMETRELERİ	9
2.1.1.Çok Yollu Yayılım	9
2.1.2.Evre-uyumlu Bant Genişliği	9
2.1.3.Evre-uyumlu Zaman.....	10
2.1.4.Doppler Yayılımı	10
2.1.5.Geniş Ölçekli Sönümlenme.....	10
2.1.6.Dar Ölçekli Sönümlenme.....	11
2.2.ÇOK YOLLU YAYILIMIN SÖNÜMLEMEYE ETKİSİ	11
2.3.DOPPLER YAYILIMI SONUCU OLUŞAN SÖNÜMLEMELER	13
2.4.RAYLEIGH SAĞILIMI	14
2.5.RİCEAN DAĞILIMI	15
2.6.KANAL TAHMİN YÖNTEMLERİ.....	16
3.DOĞURAN KOÇANLI HİLBERT UZAYINDA UYARLANABİLİR SÜZGEÇLEME	22
3.1.İÇ ÇARPIM UZAYI VE HİLBERT UZAYI	23
3.1.1.Schwarz Eşitsizliği	25

3.1.2.İç Çarpımın Süreksizliği	25
3.1.3.Tamlaştırma.....	26
3.2.DOĞURAN KOÇANLI HİLBERT UZAYI.....	27
3.2.1.Mercer Koçanı.....	28
3.3.KOÇAN UYARLANABİLİR SÜZGEÇLER.....	34
4.KOÇAN - EN KÜÇÜK ORTALAMA KARELER.....	35
4.1.EN KÜÇÜK ORTALAMA KARELER	35
4.2.KOÇAN EN KÜÇÜK EORTALAMA KARELER.....	37
5.KOÇAN YİNELEMELİ EN KÜÇÜK KARELER	41
5.1.ÜSTEL AĞIRLIKLI YİNELEMELİ EN KÜÇÜK KARELER	46
5.2.KOÇAN YİNELEMELİ EN KÜÇÜK KARELER.....	48
5.3.ÜSTEL AĞIRLIKLI YİNELEMELİ EN KÜÇÜK KARELER	54
6.UYGULAMA VE BENZETİM ÇALIŞMALARI.....	56
7. SONUÇ	70
8.KAYNAKÇA.....	72
EK	76
ÖZET.....	82
SUMMARY	83

ŞEKİL LİSTESİ

 Sayfa No.
Şekil 1.1.....	3
Şekil 1.2.....	4
Şekil 1.3.....	5
Şekil 2.1.....	17
Şekil 2.2.....	18
Şekil 3.1.....	31
Şekil 4.1.....	40
Şekil 6.1.....	58
Şekil 6.2.....	59
Şekil 6.3.....	60
Şekil 6.4.....	61
Şekil 6.5.....	62
Şekil 6.6.....	63
Şekil 6.7.....	64
Şekil 6.8.....	65
Şekil 6.9.....	67

TABLO LİSTESİ

	Sayfa No.
Tablo 1.1	7
Tablo 1.2	7
Tablo 4.1	39
Tablo 5.1	45
Tablo 5.2	48
Tablo 5.3	54
Tablo 5.4	55
Tablo 6.1	66
Tablo 6.2	68

1. GİRİŞ

Günümüzde işaret işleme ve veri haberleşmesi alanlarında doğrusal olmayan haberleşme tekniklerinin gelişimi için çok çeşitli ve kapsamlı araştırmalar yapılmaktadır. Haberleşme sistemlerinde kullanılan işaretlerin içermesi istenilen özellikler; düzensiz, periyodik olmayan, geniş bantlı ve uzun zamanlar boyunca tahmin edilebilmeleri mümkün olmayan işaretler olan kaotik işaretlerde mevcut olduğundan dolayı bu işaretler kullanılarak yeni teknikler geliştirilmektedir [Stavroulakis], [Abel ve Schwarz]. Bu özelliklerden yararlanan uygulama alanlarından bazıları yayılı-spektrum haberleşmesi, çok kullanıcılı haberleşme ve şifrelemedir [Çetinel].

Kaotik sistemler başlangıç değerlerine karşı aşırı duyarlıdırlar. Başlangıç değeri ve denklemleri bilinen deterministik bir sistemin sonraki durumları öngörülebilirken, kaotik sistemler için başlangıç değerleri sonsuz duyarlılıkta bilinmiyorsa sistemin ilerleyişini tam olarak kestirmek imkansızdır. Kaotik sistemlerde bu kestirimin tam olarak yapılamamasının en büyük sebebi başlangıç değerindeki hatanın üstel olarak artarak sistem çıktısına yansımadır [Başkurt].

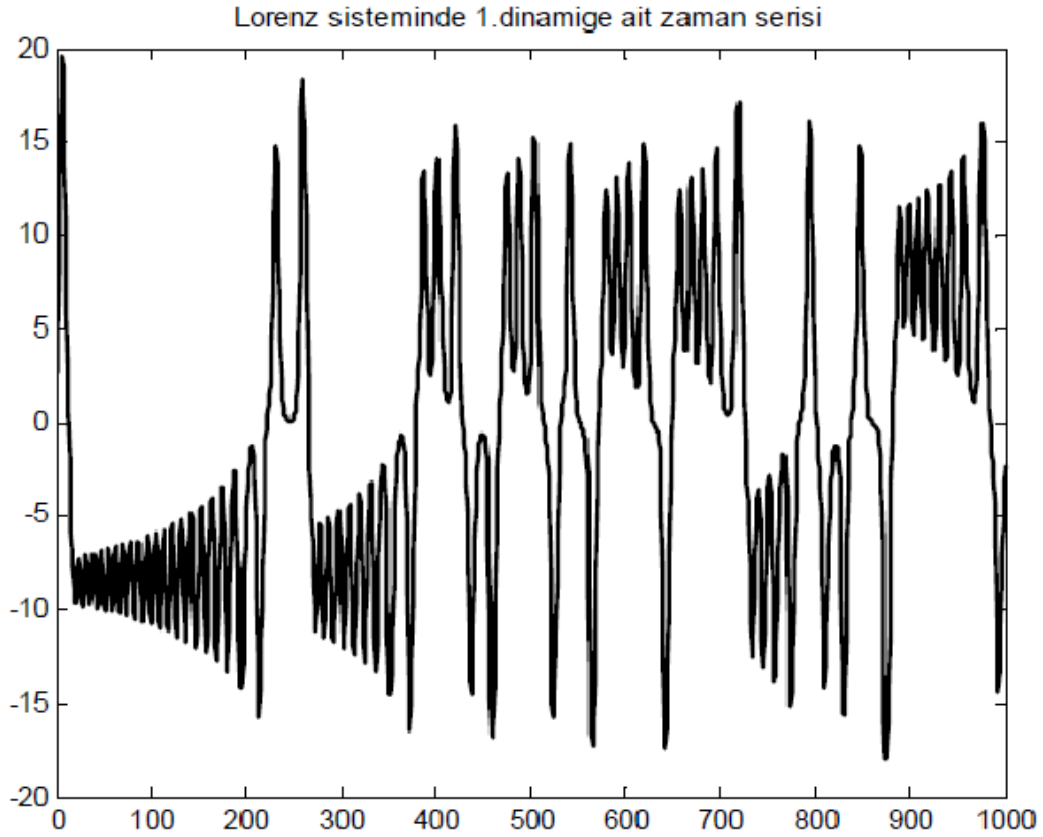
Kaotik işaret örneği olarak uygulamada da kullanacağımız ve değineceğimiz gibi Lorenz ve Mackey-Glass sistemleri dinamik cevap olarak ve/veya güç izgesi çözümlemesi uygulandığında yüksek bir çeşitlilik ve/veya geniş güç izgesine sahiptir. Buna karşın belirtilen kaotik haberleşme işareti üzerine çeşitli sebeplerle etki eden gürültüler de geniş güç izgesine sahiptirler ve bundan dolayı üzerine gürültü etki eden kaotik işareti doğrusal süzgeçler kullanarak gürültüden ayırmak zordur. Bu nedenle bu çalışmada doğrusal olmayan süzgeçleme yöntemlerinden koçan tabanlı uyarlanabilir süzgeçleri inceleyeceğiz.

Model olarak faydalanacağımız Lorenz ve Mackey-Glass sistemlerinin matematiksel ifadelerine burada kısaca değinmek yararlı olacaktır. Yine uyarlamalı süzgeçler ve kanal denkleştirme sorunlarına da bu bölümde kısaca değindikten sonra 2. Bölümde uyarlanabilir kanal denkleştirme, 3. Bölümde doğuran koçanlı Hilbert uzayında uyarlanabilir süzgeçleme, 4 ve 5. Bölümlerde sırasıyla en küçük ortalama kareler ve yinelemeli en küçük kareler yöntemlerinin doğuran koçanlı Hilbert uzayı kuramından faydalanılarak elde edilen doğrusal olmayan modelleri ve son bölüm olan 6. Bölümde de uygulamalar ve karşılaştırmalı sonuçlar sunulacak, performans değerlendirmesi yapılacaktır.

Lorenz sistemi Edward Lorenz'in hava durumu tahmini için topladığı verileri kullanarak, meteorolojik sistemleri modellemeye çalışması sonucunda ortaya çıkmıştır. Bu sistem şöyledir:

$$\begin{aligned}
 \frac{dx_1(t)}{dt} &= \sigma [x_2(t) - x_1(t)] \\
 \frac{dx_2(t)}{dt} &= x_1(t) [r - x_3(t)] - x_2(t) \\
 \frac{dx_3(t)}{dt} &= x_1(t)x_2(t) - bx_3(t)
 \end{aligned}
 \tag{1.1}$$

burada σ , b ve r değişken parametrelerdir. Sistem $\sigma = 10$, $b = 8/3$ ve $r = 28$ değerlerinde kaotik davranış göstermektedir. Kaotik davranış altında sistem, izgesi (spektrumu) geniş bir frekans bölgesine yayılmış periyodik olmayan salınımlar üretmektedir.



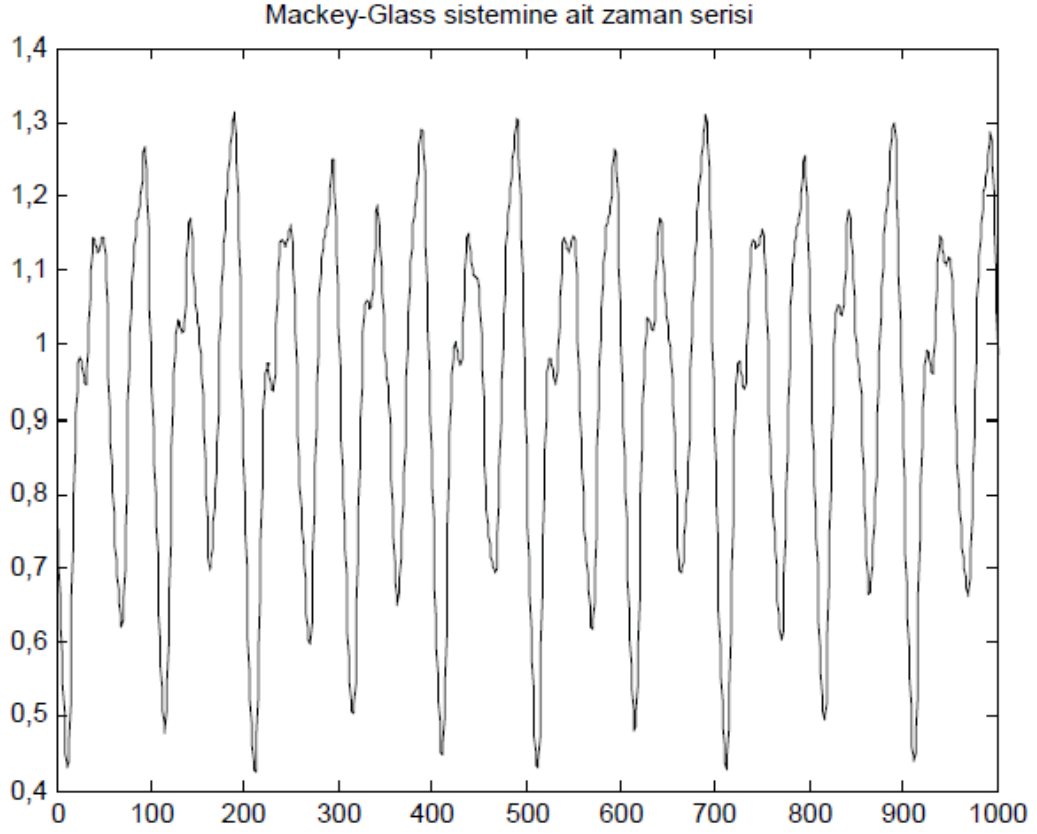
Şekil 1.1 Lorenz sisteminde $x_1(t)$ 'ye ait zaman serisi [Özer ve Zorlu]

Mackey-Glass sistemi literatürde tahmin amaçlı yapılan uygulamalarda sıklıkla yer bulan bir kaotik sistemdir. Matematiksel ifadesi şöyledir:

$$\frac{dy(t)}{dt} = \frac{0.2y(t-\tau)}{1+y^{10}(t-\tau)} - 0.2y(t) \quad (1.2)$$

burada $\tau > 17$ alındığında sistem çıkışında kaotik zaman serisi görülmektedir.

Her iki kaotik davranışlı dinamik sistemin benzetimleri gerçekleştirildiğinde elde edilen dinamik cevaplar, fikir oluşturması açısından Şekil 1.1 ve Şekil 1.2'de gösterilmiştir.



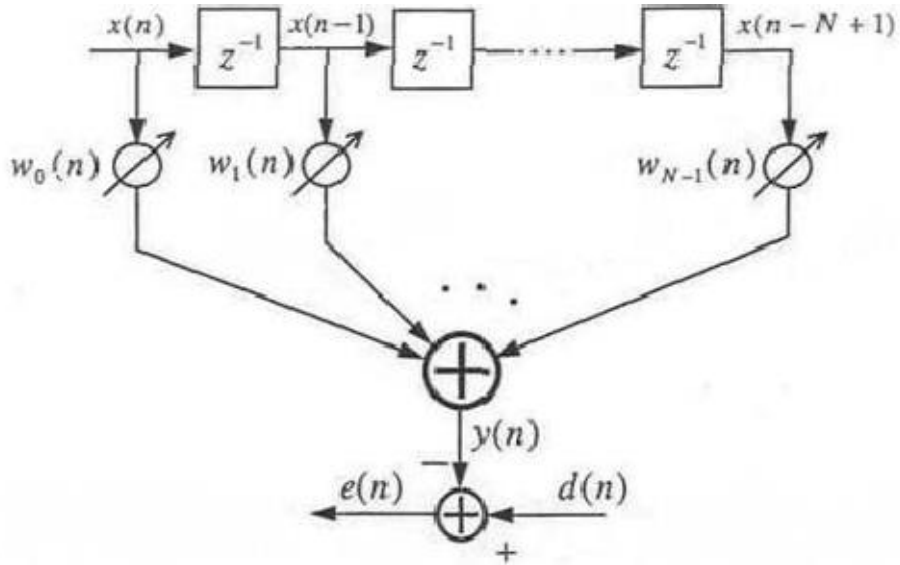
Şekil 1.2 Mackey-Glass Sistemine ait zaman serisi [Özer ve Zorlu]

Gürültülü işaretten süzgeçleme işlemi sonunda elde edilecek asıl sinyal hakkında istatistiksel bilginin bulunmadığı durumlarda uyarlanabilir süzgeç yöntemleri kullanılmaktadır. Uyarlanabilir süzgeçler, ilgili işaretin karakteristik bilgisinin ortamda bulunmadığı durumlarda yinelemeli algoritmalarla, kendi kendini bulunduğu ortama göre ayarlayarak istenilen çıktının oluşturulmasını sağlamaktadır [Sevim].

Kanal denkleştirme problemi, bu bağlamda alınan işaretin kanalın tersinden tekrar geçirilmesi olarak yorumlanabilir. Bunun gerçekleştirilebilmesi için, alıcıda kanal bilgisinin ya da, doğrudan tersinin elde edilmesi gerekmektedir.

Bu amaçla, kanal veya kanalın tersini (kanal denkleştirici) kestirmek için çeşitli algoritmalar kullanılmaktadır. İletişim kanalının zamanla değişmesi durumunda, kanal denkleştiricinin kanaldaki değişimleri takip edebilmesi gerekmektedir. Bu tür kanal denkleştiriciler, uyarlanabilir kanal denkleştiriciler olarak adlandırılırlar.

En küçük ortalama kareler (LMS - LeastMeanSquares) algoritması, maliyet fonksiyonunun, eğim vektörünün (gradyan vektörü) anlık kestirimini kullanan en dik azalış (steepestdescend) algoritmasının bir versiyonudur. Eğimin kestirimi, tapa giriş ve hata işaretinin örnek değerlerini temel alır. Algoritma süzgeçteki her katsayı üzerinden tekrarlanır ve yaklaşık eğim doğrultusunda ilerler [Feldbauer, Pernkopf, Rank].



Şekil 1.3 N-tapalı uyarlanabilir süzgeç yapısı [Farhang-Boroujeny]

N-tapalı uyarlanabilir süzgecin yapısı Şekil 1.3'te görülmektedir. Süzgeç girişi $x(n)$ olarak tanımlanır ve referans işaret de $d(n)$ ile gösterilirse süzgeç çıkışı da,

$$y(n) = W_0(n)*x(n) + W_1(n)*x(n-1) + \dots + W_{N-1}(n)*x(n - N + 1) \quad (1.3)$$

şeklinde ifade edilebilir. W ağırlık değerleri hatayı en aza indirecek şekilde seçilmelidir. Hata ise referans işaret ile süzgeç çıkışının farkına karşılık gelmektedir.

$$e(n) = d(n) - y(n) \quad (1.4)$$

Süzgeçteki ağırlık değerleri, "n" zaman indeksinin fonksiyonlarıdır. Bu, uyarlanabilir süzgeçte ağırlık değerlerinin zamanla değiştiği anlamına gelir. İşaret karakteristiğinde herhangi bir değişimin olduğunda süzgecin ağırlık değerleri güncellenerek süzgeç çıkışının referans işareti izlemesi sağlanabilir. En küçük ortalama kareler (LMS) algoritması ağırlık değerlerini uyarlayarak ortalama kare hatayı minimize eder.

En küçük ortalama kareler (LMS) algoritmasında asıl amaç kestirim hatasının ortalama karesini minimize etmektir. Yinelemeli en küçük kareler (RecursiveLeastSquares - RLS) algoritmasında ise $n > 0$ için herhangi bir zamanda, ağırlık değerleri hesaplanır ve maliyet fonksiyonunun elde edilen bu ağırlık değerleri yardımıyla minimum olması sağlanır. Yinelemeli en küçük kareler (RLS) algoritması giriş verisi içerisindeki bilgiyi kullanır. Bundan dolayı yakınsama hızı en küçük ortalama kareler (LMS) algoritmasına göre daha hızlıdır. Karmaşık hesaplamalarda bu performans artırımını daha belirgin

hale gelmektedir [Haykin]. Her iki algoritma için genel kod yazımları Tablo 1.1 ve Tablo 1.2'de gösterilmektedir.

Başlangıç

$\mathbf{w}(0) = 0$, seç η

Hesap

while $\{\mathbf{u}(i), d(i)\}$ ise **do**
 $e(i) = d(i) - \mathbf{w}^T(i-1)\mathbf{u}(i)$
 $\mathbf{w}(i) = \mathbf{w}(i-1) + \eta e(i)\mathbf{u}(i)$
end while

Tablo 1.1 En küçük ortalama kareler algoritması [Liu, Principe ve Haykin]

Başlangıç

$\mathbf{w}(0) = 0$, $\mathbf{P}(0)$

Hesap

İlerlet $i \geq 1$

$$r(i) = 1 + \mathbf{u}(i)^T \mathbf{P}(i-1) \mathbf{u}(i)$$

$$\mathbf{k}(i) = \mathbf{P}(i-1) \mathbf{u}(i) / r(i)$$

$$e(i) = d(i) - \mathbf{u}(i)^T \mathbf{w}(i-1)$$

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \mathbf{k}(i) e(i)$$

$$\mathbf{P}(i) = [\mathbf{P}(i-1) - \mathbf{k}(i) \mathbf{k}(i)^T r(i)]$$

Tablo 1.2 Yinelenen en küçük kareler algoritması [Liu, Principe ve Haykin]

2. UYARLANABİLİR KANAL DENKLEŞTİRME

Kanal tanımı genel olarak herhangi bir haberleşme işaretinin kaynaktan hedefe ulaşıncaya kadar iletim esnasında maruz kaldığı değişimler bütünü olarak verilebilir, kısaca fiziksel iletim ortamı olarak da kabul edilebilir. Kanal kestirimi ise, fiziksel ortamın giriş verisi üzerindeki etkilerini belirleme ve veriyi çıkışta yeniden oluşturma işlemidir. Bu sayede gürültü, semboller arası girişim ve kanallar arası girişim-karışma gibi etkiler ortadan kaldırılabilmekte ve iletimi hedeflenen işarete ulaşılabilmektedir.

Örneğin geniş bantlı mobil haberleşme sistemlerinde, radyo kanalı frekans seçicili olduğu ve zamanla değiştiği için OFDM işareti demodüle edilmeden önce dinamik kanal kestirimi gereklidir [Özşahin].

Kanal denkleştirme problemi, kanal parametrelerinin bilinip bilinmemesine göre gözü kapalı olmayan ve gözü kapalı kanal denkleştirme olmak üzere ikiye ayrılabilir ve haberleşme sisteminin giriş ve çıkışındaki işaret sayısına göre ise tek girişli tek çıkışlı (SISO – SingleInputSingleOutput) ve çok girişli çok çıkışlı (MIMO – Multi Input Multi Output) kanal denkleştirme şeklinde de sınıflandırılabilir [Çetinel].

Bu bölümde ilk olarak kablosuz haberleşme kanallarına ait parametreler üzerinde durulacak, ardından uyarlanabilir süzgeçler ile kanal kestirimi için kullanılan model tabanlı (gözü kapalı olmayan) yöntemlere değinilecektir.

İletişim sistemlerini en fazla etkileyen bozucular şöyle sıralanabilir; toplanır beyaz Gauss gürültüsü (AWGN – additivewhitegaussiannoise), simgeler arası girişim (ISI – intersymbolinference), kanallar arası girişim (ICI – interchannelinference), yol kaybı (pathloss), gölgeleme (shadowing) ve çok yollu sönümlenme (multipathfading).

2.1. HABERLEŐME KANAL PARAMETRELERİ

Bu baŐlık altında, haberleŐme kanallarının özellik ve performanslarını belirleyen parametreler incelenecektir.

2.1.1. Çok Yollu Yayılım

Sönümlenmeli bir kanal girişine gönderilmiş olan dürtü (impuls) işaretinin alıcı tarafında, ilk ve son görünümleri arasındaki farka çok yollu yayılım(maksimum zaman gecikmesi) denilmektedir ve T_m ile gösterilmektedir [Özşahin], [Sklar], [Stüber].

2.1.2. Evre-uyumlu Bant GeniŐliđi

Evre-uyumlu bant geniŐliđi, kanalın belirli bir işaretin izgesel(spektral) bileŐenlerinin yaklaşık olarak eŐit kazanç ve dođrusal evrede geçirdiđi frekans aralıđıdır ve f_c ile gösterilmektedir [Özşahin], [Sklar], [Stüber]. Evre-uyumlu bant geniŐliđi ile çok yollu yayılım arasındaki iliŐki,

$$f_c \approx 1 / T_m \quad (2.1)$$

Őeklinde yazılabilir. BaŐka bir ifadeyle, f_c yardımıyla işaretlerin kanalda farklı derecelerde sönümlenmeye uğramaları için frekanslarının ne kadar ayık ya da belirgin olacakları hakkında bir ön bilgi edinilmektedir [Özşahin], [Stüber].

2.1.3. Evre-uyumlu Zaman

Evre-uyumlu zaman, T_c , kanalın dürtü (impuls) işaretine verdiği yanıtının sabit kaldığı ya da yüksek ilişkiye sahip olduğu zaman aralığını ifade etmek için kullanılır. Bu durumda, eğer simge süresi T_c 'den küçük olursa simgenin iletim boyunca zamanda değişmediği varsayılır [Özşahin, [Stüber].

2.1.4. Doppler Yayılımı

Kanalın zamanla değişim özelliğinden dolayı meydana gelen izgesel(spektral) genişleme Doppler yayılımı B_d ile ifade edilir. Doppler yayılımı, iletilen bir işaretin Dopplerizgesinin (spektrumunun) sıfırdan farklı olduğu frekans aralığını ya da aralıklarını vermektedir. İzgesel genişlemenin miktarı gezgin (mobil) birimin göreceli hızına ve/veya işaretin geliş açısına bağlıdır [Özşahin], [Stüber]. Evre-uyumlu zaman ile Doppler yayılımı arasında,

$$T_c \approx 1 / B_d \quad (2.2)$$

şeklinde bir ilişki vardır.

2.1.5. Geniş Ölçekli Sönümlenme

Bu tip sönümlenme, geniş alanlarda hareket etme sonucu ortalama işaret gücünün zayıflamasını ya da yoldan kaynaklanan kaybını ifade eder. Geniş ölçekli sönümlenmede verici ile alıcı arasında bulunan engebeli yeryüzü şekilleri ve çok katlı binalar önemli etkenlerdir. Bu tür etkenlerden dolayı

meydana gelen sönümlenmelere gölgeleme(shadowing) denilmektedir [Özşahin], [Stüber].

2.1.6. Dar Ölçekli Sönümlenme

Dar ölçekli sönümlenme, alıcı ve verici arasındaki uzaklığın küçük değişimleri sonucuna bağlı olarak işaretin genlik ve fazındaki dinamik değişimleri ifade eder. Dar ölçekli sönümlenme, işaretin zaman yayılımı (time-spreading) ve kanalın zamanla değişim özelliğinden kaynaklanmaktadır [Özşahin], [Sklar]. İşaret parametreleri (bant genişliği, sembol periyodu, vs) ve kanal parametreleri arasındaki ilişkiye göre, kanaldan iletilen her farklı işaret farklı sönümlenmelere uğrayacaktır [Özşahin], [Stüber]. Çok yollu yayılım, zaman yayılımı ve frekans seçici sönümlenmeye yol açarken Doppler yayılımı ve frekans yayılımı, zaman seçici sönümlenmeye (time selective fading) yol açmaktadır. Bu iki yayılım mekanizmaları ise birbirinden bağımsızdır [Özşahin],[Stüber].

2.2. ÇOK YOLLU YAYILIMIN SÖNÜMLENMEYE ETKİSİ

Çok yollu yayılım iletilen işaretin düz (flat) ya da frekans seçici (frequencyselective)sönümlenmeye uğramasına neden olur [Özşahin],[Stüber].

Eğer mobil radyo kanalı iletilen işaretin bant genişliğinden geniş, bant genişliği boyunca ise sabit kazanç ve doğrusal evre yanıtına sahipse alıcıdaki işaret de düzsönümlenmeye uğrar. Düz sönümlenmede kanalın çok yollu yapısı iletilen işaretin izgesel(spektral) karakteristiğinin alıcıda aynen korunacağı anlamına gelir. Fakat alıcıdaki işaretin gücü zamanla

değişmektedir; bunun nedeni ise, çok sayıda yolun etkisiyle kanalın kazancında değişmelerin olmasıdır [Özşahin],[Sklar].

Düz sönümlenmeli kanallar genlik değiştiren kanallar olarak da bilinir. Kanal girişine gönderilen işaretin bant genişliği kanalın bant genişliğinden çok daha dar olduğundan dolayı ise dar bantlı kanallar olarak da bilinmektedirler. Düz sönümlenmeli kanallar derin sönümlenmelere neden olurlar, bunun için sönümlenme olmayan kanallara oranla daha düşük bitte hata olasılığı elde etmek için gönderilecek işaretin gücü 20 dB ya da 30 dB fazla olmalıdır [Özşahin],[Stüber]. Kısaca ifade etmek gerekirse, bir işaretin düz sönümlenmeye uğraması için,

$$B_s \ll B_c, T_s \gg \sigma \quad (2.3)$$

olmalıdır. Burada T_s , işaretin simge periyodu, B_s , işaretin bant genişliği, σ , kanalın gecikme yayılımı, B_c , kanalın evre-uyumlu bant genişliğidir.

Eğer, kanal iletilen işaretin bant genişliğinden daha dar bir bant genişliği boyunca sabit kazanç ve doğrusal evre yanıtına sahipse bu kanal bir frekans seçici sönümlenmeli kanaldır.

Bu durumda kanalın darbe yanıtı iletilen işaretin simge periyodundan daha büyük bir çok yönlü yayılıma sahiptir. Bu şekilde, alıcı tarafındaki işaret, gönderilen işaretin zayıflamış ve zamanda gecikmiş bir çok bileşeninden oluşacaktır. Doğal olarak bu durumda da alıcıda işaret bozulmaya uğramış olacaktır. Frekans seçici sönümlenme iletilen simgelerin kanal içinde zaman yayılımına uğramalarından kaynaklanmaktadır. Bu yüzden bu kanallar semboller arası girişime (ISI – intersymbol interference) neden olurlar [Özşahin],[Stüber].

Frekans seçici sönümlenmede iletilen işaretin bant genişliği kanalın bant genişliğinden büyüktür. Yine, frekans seçici sönümlenme çok yollu gecikmelerin iletilen simgenin periyodunu geçmesi sonucu meydana gelir. Özetlersek, frekans seçici sönümlenme durumu için,

$$B_s > B_c, T_s < \sigma \quad (2.4)$$

eşitsizliği sağlanmış olmalıdır [Özşahin].

2.3. DOPPLER YAYILIMI SONUCU OLUŞAN SÖNÜMLENMELER

Doppler yayılımı sonucu oluşan sönümlenmeler hızlı sönümlenme ve yavaş sönümlenme olmak üzere ikiye ayrılır.

İletilen temel bant işaretin değişim hızının kanalın değişim hızına göre ne kadar hızlı olduğu, kanalın hızlı sönümlenmeli ya da yavaş sönümlenmeli bir kanal olduğunu belirler. Hızlı sönümlenmeli bir kanalda, kanalın dürtü yanıtı bir sembol periyodu boyunca çok hızlı değişir. Bu durum kanalın evre-uyum zamanının iletilen işaretin sembol periyodundan daha küçük olduğu anlamına gelmektedir. Bu kanal türünde Doppler yayılımından dolayı frekans yayılımı meydana gelir ve bu da işaret bozulmalarına neden olur [Özşahin],[Sklar]. Hızlı sönümlenme aşağıdaki eşitsizlik durumlarında meydana gelir,

$$T_s > T_c, B_s < B_d \quad (2.5)$$

Bir kanalın hızlı ya da yavaş sönümlenmeli bir kanal olması ise bu kanalın düz ya da frekans seçici sönümlenmeli bir kanal olduğunu belirtmez. Hızlı sönümlenme sadece kanalın hareket sonucu değişme oranı ile ilgilidir. Düz sönümlenmeli kanallarda ise kanalın dürtü yanıtının sadece bir delta işlevi

(gecikmesiz) olduğunu düşünülebilir. Bundan dolayı, düz ve hızlı sönümlenmeli bir kanal,delta işlevi genliğinin değişimi iletilen temel bant işaretin değişim oranından daha hızlı olduğu bir kanaldır. Frekans seçici, hızlı sönümlenmelikanalarda ise her birçok yollu elemanın zaman gecikmesinin, genliğinin ve evresinin değişimi, iletilen işaretin değişim oranına göre daha hızlıdır [Özşahin], [Stüber].

Yavaş sönümlenmeli kanallarda, kanalın darbe yanıtı iletilen temel bant işaretinin değişimine göre daha yavaş değişmektedir. Bu durumda kanaldaki iletimin birkaç simge periyodu boyunca sabit kaldığı düşünülebilir. Frekans bölgesinde, kanalın Doppler yayılımı temel bant işaretinin bant genişliğinden çok küçüktür [Özşahin],[Stüber]. Yavaş sönümlenme şu koşullarda meydana gelir,

$$T_s \ll T_c$$

$$B_s \gg B_d$$

(2.6)

Mobil birimlerin hızları ve temel bant işaretin karakteristiği iletilen işaretin hızlı ya da yavaş sönümlenmeye uğrayıp uğramayacağını belirler [Özşahin],[13].

2.4. RAYLEIGH DAĞILIMI

Mobil radyo kanallarında Rayleigh dağılımı düz sönümlenmeye uğramış bir işaretin alıcı tarafındaki zarfının istatistiksel olarak zamanla değişimini tanımlamakta kullanılmaktadır. Gürültü işaretinin dik bileşenleri Gauss dağılımlı olup zarfı Rayleigh dağılımını vermektedir [Özşahin], [Stüber]. Rayleigh dağılımı aşağıdaki olasılık yoğunluk işlevine (pdf – probabilitydensityfunction) sahiptir,

$$P_{Rayleigh}(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right) & 0 \leq r \leq \infty \\ 0 & r < 0, \end{cases}$$

(2.7)

burada σ^3 dik bileşenlerin ortalama gücüdür. Rayleigh dağılımının ortalama değeri ve varyansı şöyledir,

$$r_{mean} = E[r] = \int_0^{\infty} r p(r) dr = \sigma \sqrt{\frac{\pi}{2}} = 1.2533\sigma$$

$$\sigma_r^2 = E[r^2] - E^2[r] = \int_0^{\infty} r^2 p(r) dr - \frac{\sigma^2 \pi}{2}$$

$$= \sigma^2 \left(2 - \frac{\pi}{2}\right) = 0.4292\sigma^2$$

(2.8)

2.5. RICEAN DAĞILIMI

Küçük ölçekli sönümlenmeli kanalların genlik zarfı, baskın zayıflamamış bir işaret bileşeni varsa (Line-of-side gibi) Ricean dağılımlıdır [Özşahin], [Stüber]. Zarf algılayıcı çıkışında baskın işaret, doğru akım bileşeninin oluşmasına neden olur. Ricean dağılımında baskın işaret bileşeni sönümlenirse Rayleigh dağılımı elde edilir [Özşahin], [Stüber]. Ricean dağılımı şöyledir,

$$p_r(r) = \frac{r}{\sigma^2} e^{-\frac{(r^2+A^2)}{2\sigma^2}} I_0\left(\frac{rA}{\sigma^2}\right) \quad , A \geq 0 \text{ ve } r \geq 0$$

(2.9)

A , baskın işaretin maksimum değeri (genliği); $I_0(\cdot)$, 0. derece 1. türden Bessel fonksiyonudur.

Ricean dağılımı genellikle K parametresi ile ifade edilir. K parametresi ise şu şekilde hesaplanmaktadır,

$$K(dB) = 10 \log \frac{A^2}{2\sigma^2} dB$$

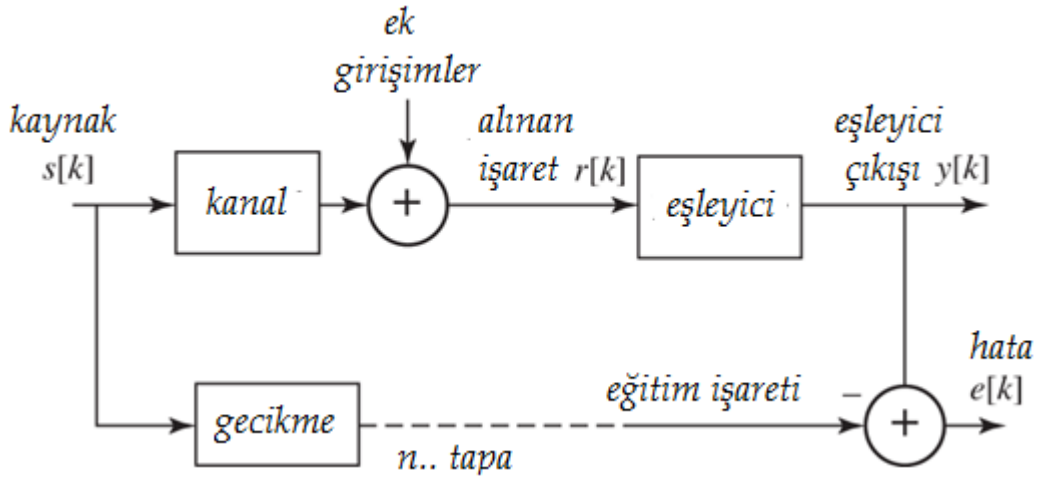
(2.10)

$A \rightarrow 0$, $K \rightarrow -\infty dB$ limit durumunda Ricean dağılımı Rayleigh dağılımına dönüşür. $K \gg 1$ olduğu durumda ise Ricean dağılımı Gauss dağılımına yakınsar [Özşahin], [Stüber].

2.6. KANAL TAHMİN YÖNTEMLERİ

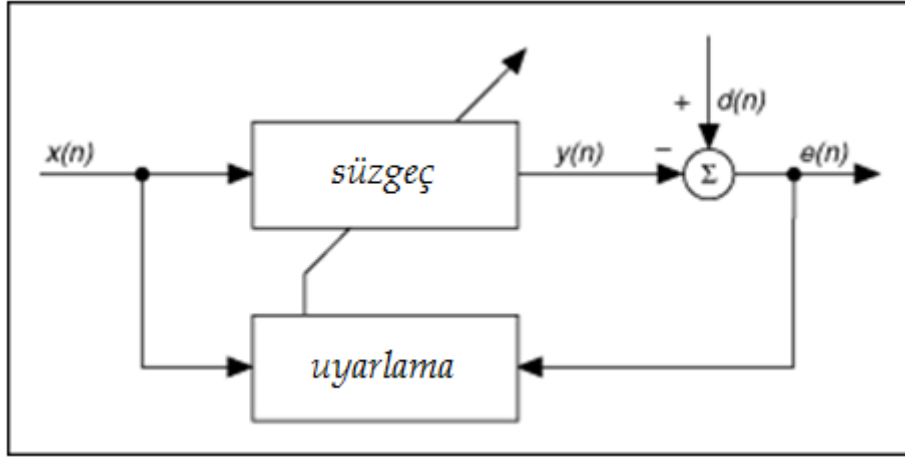
Uyarlanabilir kanal denkleştirme için kullanılan model tabanlı yöntemlerin genel yapısı Şekil 2.1'te gösterilmektedir. Burada daha önce de belirtildiği

gibi kaynak işareti $s[k]$ kanaldan iletilmekte ve çeşitli sönümlenme, gürültü ve ek girişimlere maruz kalmaktadır. Tüm bozulma ve değişimler sonucunda alıcı tarafında elde edilen işaret ise $r[k]$ ile temsil edilen alınan işarettir ve bu işaret denkleştirici bloğuna veya devresine girmektedir. Denkleştirici çıkışı sabit olabileceği gibi şekilde gösterilen yapıda olduğu gibi uyarlanabilir de olabilir. Nitekim denkleştirici çıkışını düzenleyebilmek amacı ile kaynak işareti n tapalı bir geciktiriciden geçirilerek denkleştirici çıkışı ile karşılaştırıldığında denkleştirici bloğunda bulunan uyarlanabilir süzgeç parametreleri hatayı en aza indirecek şekilde belirlenebilir. Bu durumda uyarlanabilir kanal denkleştirme işlemi gerçekleştirilmiş olur.



Şekil 2.1 Kanal ve denkleştirici yapısı

Şekil 2.1'de gösterilen yapıda denkleştirici bloğu esasında bir uyarlanabilir süzgeç içermektedir ve bu süzgeç de en genel hali ile Şekil 2.2'de verilmiştir. Bu genel diyagram daha önce Şekil 1.3'te gösterilen ve açıklanan n tapalı uyarlanabilir süzgeç yapısının kompakt gösterimi olarak görülebilir. Ancak bu yapıda uyarlama bloğu çok çeşitli doğrusal ve/veya doğrusal olmayan algoritmalar ile işlem yapılabilir, bu yüzden en genel yapıdır.



Şekil 2.2 Uyarlanabilir süzgeç - Denkleştirici

Bu en genel yapıdan yola çıkacak olursak, uyarlanabilir süzgeci şöyle tanımlarız: Giriş işaretindeki değişimlere uygun olarak frekans tepkesini değiştiren süzgeçlere uyarlanabilir süzgeç adı verilir [Sunan]. Nitekim bu yapıda,

$x(n)$: Giriş işareti

$y(n)$: Çıkış işareti

$d(n)$: Referans işareti

$e(n)$: Hata işareti

Uyarlanabilir süzgeç, ayarlanabilir bir süzgeç ve uyarlama algoritması olmak üzere iki ana kısımdan oluşur. Ayarlanabilir süzgeçler, katsayılarının değiştirilerek frekans yanıtının değiştirilebildiği süzgeçlerdir. Süzgeç katsayılarının değiştirilmesi uyarlama algoritması tarafından yapılır. Uyarlanabilir süzgeç çıkışı $y(n)$, referans işaret $d(n)$ ' den çıkartılarak hata

işareti elde edilir. Uyarlama algoritmasının görevi hata işareti $e(n)$ ' i minimum yapacak şekilde süzgeç katsayılarını değiştirmektir. İdealde hata sıfır olur [Mahmut].

Uyarlanabilir süzgeçler sayısal, analog veya her ikisinin karışımı olan hibrit yapıda olabilirler. Bu durumda işaretlerin işlenebilmesi ve/veya kullanılabilmesi için gerekli Analog/Sayısal ve Sayısal/Analog dönüşümlerinin yapılması gereklidir.

Sayısal süzgeçler darbe tepkilerine göre,

- Sonlu darbe tepkili (FIR) süzgeçler
- Sonsuz darbe tepkili (IIR) süzgeçler

olarak iki ana gruba ayrılırlar. FIR süzgeçlerde çıkış, sadece girişin o andaki ve eski değerlerine bağlı olup çıkışın önceki değerlerine bağlı değildir. IIR süzgeçlerde ise çıkış, giriş ve girişin gecikmelerine ek olarak çıkışın önceki değerlerine de bağlıdır. Bu işlemler geri besleme sayesinde gerçekleştirilir [Birlikbaş].

Uyarlama algoritmalarının görevi hata işaretini en aza indirecek şekilde süzgeç katsayılarını değiştirmektir. Genel olarak uyarlama algoritmaları süzgeç katsayılarının eski değerlerini kullanarak yeni değerlerinin hesaplandığı yinelemeli algoritmalarlardır. Uyarlama algoritmaları iki ana gruba ayrılabilir.

- Rasgele Gradyan Yaklaşımı (StochasticGradientApproach)
- En Küçük Kareler Kestirimi (LeastSquaresEstimation)

Rasgele gradyan (eğim) yaklaşımında algoritma hatanın ortalama karesini en aza indirmeye çalışır. Ortalama kare hata (MeanSquareError-MSE),

hatanın karesinin beklenen değeri(Expectedvalue) alınarak hesaplanır. Beklenen değer, istatistikte olası olasılıkların dağılımının aritmetik ortalaması veya kısaca ortalaması olarak tanımlanır. Beklenen değer ise şöyledir,

$$E[X] = \sum_{x=0}^n x f(x) \quad (2.11)$$

$f(x)$ frekans veya yoğunluk fonksiyonu olarak adlandırılır ve,

$$f(x) = \frac{n!}{x!(n-x)!} p^x q^{n-x} \quad ; p + q = 1 \quad (2.12)$$

şeklindedir. Burada:

- p: Arzulanan sonucun gelme olasılığı
- q: Arzulanan sonucun gelmeme olasılığı
- x: Belli bir sonucun gelmesini kaç defa istediğimiz
- n: Deneme sayısıdır.

Gradyan (eğim) yaklaşımlı algoritmaların en sık kullanılanları, en dik iniş (SteepestDescent) ve en küçük ortalama kareler (LeastMeanSquares-LMS) algoritmalarıdır.

En küçük kareler kestirim yaklaşımında ortalama kare hata (MeanSquareError-MSE), yerine hataların karelerinin toplamı minimize edilmeye çalışılır. Bu yapıda en sık kullanılan algoritma ise yinelemeli en küçük kareler (RecursiveLeastSquares - RLS) kestirim algoritmasıdır.

En küçük ortalama kareler (LeastMeanSquares – LMS) algoritması, basitliğinden ve kararlılığından dolayı en çok kullanılan denkleştirici algoritmasıdır. En büyük dezavantajı ise, diğer algoritmaya göre daha yavaş yakınsamasıdır. Sözelimi, LMS algoritması ortalama 100-1000 arası bir sembolde yakınsar [Mahmut].

Daha hızlı olan kanal denkleştirme algoritması ise yinelemeli en küçük kareler (RecursiveLeastSquares – RLS) algoritmasıdır. RLS algoritmasının farklı karmaşıklık ve yakınsama kriterlerine sahip olan çeşitli versiyonları da mevcuttur. Ancak RLS algoritmasının uygulanması LMS algoritmasına göre daha zordur. Bunun yanında RLS algoritması da LMS algoritmasına göre daha az sembolde yakınsar.

Gezgin iletişim sistemlerinde, RLS ve LMS algoritmaları üzerinde birçok araştırma yapıyor olmasına rağmen, bu algoritmalar yüksek bit hızlı kablosuz erişim uygulamaları için karmaşıklık, kararlılık ve hızlı yakınsama gibi kriterlerden dolayı canlı bir araştırma konusu olma özelliğini korumaktadır. Nitekim bu çalışmada da hem hızlı yakınsama hem de kararlılık üzerine iyileştirme sağlamak amacı ile bu algoritmaların doğrusal olmayan versiyonları üzerinde yoğunlaşmıştır.

3. DOĞURAN KOÇANLI HİLBERT UZAYINDA UYARLANABİLİR SÜZGEÇLEME

Normu olan bir uzayda, elemanter vektör analizinde olduğu gibi, vektörleri toplayabilir veskalerler ile çarpabiliriz. Ayrıca, norm kavramı, böyle bir uzay üzerinde, bir vektörünelemanter uzunluk kavramını genelleştirir. Bununla birlikte, genel bir normlu uzayda, yapılması istenilen,

$$a.b = a_1.b_1 + a_2.b_2 + a_3.b_3 \quad (3.1)$$

gibi bir skaler çarpım tanımlamak ve bunun sonucu olarak da,

$$"a" = \sqrt{a.a} \quad (3.2)$$

formülünü ve diklik için de,

$$a.b = 0 \quad (3.3)$$

koşulunu elde edebilmektir. Bu durumda, skaler çarpım ve diklik kavramlarının herhangi vektör uzaylarına genişletilmesi sonucunda iç çarpım uzayı ve daha sonra da, Hilbert uzayı adı verilen tam iç çarpım uzaylarına ulaşılmaktadır.

Burada konu ile ilgili önemli kavramlar ve temel tanımları [Çakar]'da yapıldığı gibi kısaca özetlenmesi gerekmektedir.

Bir X iç çarpım uzayı, üzerinde bir $\langle x,y \rangle$ iç çarpımı tanımlı olan bir X vektör uzayıdır. İç çarpım, üç boyutlu uzaylarda vektörlerin skaler çarpımı kavramını genelleştiren ve

- I. " x " = $\epsilon_{xx} X^2$ ile bir "." normu
- II. $\epsilon_{xx} X = 0$ ile bir dikliđi tanımlamaktadır.

Bir H Hilbert uzayı, tam olan bir iç çarpım uzayıdır. İç çarpım ve Hilbert uzayları teorisi, genel normlu uzaylar ve Banach uzayları teorisinden daha zengindir. Bu zenginliđi ortaya çıkaran hususlar,

1. H 'nin, kapalı bir alt uzayı ile bu alt uzayın dik tümleyeninin direkt toplamı olarak belirtilmesi
2. Ortonormal kümeler ve diziler ve H 'nin elemanlarının bunlara karşılık gelen gösterimleri
3. Sınırlı lineer fonksiyonellerin iç çarpım yardımıyla Riesz gösterimi
4. Sınırlı lineer bir T operatörünün, T^* Hilbert-adjoint operatörünün bulunmasıdır.

3.1. İÇ ÇARPIM UZAYI VE HİLBERT UZAYI

Bir iç çarpım uzayı (diđer bir deyişle, ön-Hilbert uzayı), üzerinde bir iç çarpım tanımlanmış bir X vektör uzayıdır.

Bir Hilbert uzayı ise, üzerindeki iç çarpım ile tanımlanmış metriđe göre, tam olan bir iç çarpım uzayıdır.

Burada sözü edilen iç çarpım, $X \times X$ 'den X 'in bir k skaler cisimi içine yapılan bir dönüşümdür; yani, X 'in her x ve y vektör çifti, x ve y 'nin vektörel çarpımı olarak adlandırılan ve $\langle x, y \rangle$ ile gösterilen ve her x, y ve z vektörleri ve a skaleri için aşağıdaki özellikleri gerçekleyen bir skalerle eşlenmektedir:

$$a) \langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$$

$$b) \langle ax, y \rangle = a \langle x, y \rangle$$

$$c) \langle x, y \rangle_X = \overline{\langle y, x \rangle_X} \quad (\text{karmaşık eşlenik / reel ise } \langle x, y \rangle_X = \langle y, x \rangle_X)$$

$$d) \langle x, x \rangle_X \geq 0 \quad \langle x, x \rangle_X = 0 \iff x = 0$$

X üzerinde tanımlanan bir iç çarpım, X üzerinde,

$$\|x\| = \sqrt{\langle x, x \rangle_X} \quad (3.4)$$

ile verilen bir norm ve

$$d(x, y) = \|x - y\| = \sqrt{\langle x - y, x - y \rangle_X} \quad (3.5)$$

ile verilen bir metrik tanımlar [Çakar].

Bu özellikler sonucunda aşağıdaki bağıntılar elde edilir:

$$(i) \quad \langle ax + by, z \rangle_X = a \langle x, z \rangle_X + b \langle y, z \rangle_X$$

$$(ii) \quad \langle x, ay \rangle_X = \overline{a} \langle x, y \rangle_X$$

$$(iii) \quad \langle x, ay + bz \rangle_X = \overline{a} \langle x, y \rangle_X + \overline{b} \langle x, z \rangle_X$$

i bağıntısından iç çarpımın birinci çarpana göre lineer olduğu, iii bağıntısından da iç çarpımın ikinci çarpana göre eşlenik lineer olduğu sonucu ortaya çıkmaktadır.

3.1.1. Schwarz Eşitsizliği

Bir iç çarpım ve buna karşılık gelen norm Schwarz eşitsizliğini ve üçgen eşitsizliğini, aşağıda belirtildiği şekilde gerçekler [Çakar]:

- (a) Eşitlik hali, ancak ve ancak, $\{x,y\}$ 'nin lineer bağımlı olması halinde geçerli olmak üzere şöyledir:

$$|\langle x, y \rangle| \leq \|x\| \|y\| \quad (\text{Schwarz Eşitsizliği})$$

- (b) Söz konusu norm, eşitlik hali, ancak ve ancak, $y = 0$ ya da, $x = cy$ (c reel ve ≥ 0) halinde geçerli olmak üzere,

$$\|x + y\| \leq \|x\| + \|y\| \quad (\text{Üçgen Eşitsizliği})$$

eşitsizliğini gerçekler.

3.1.2. İç Çarpımın Sürekliliği

Bir iç çarpım uzayında, $x_n \rightarrow x$ ve $y_n \rightarrow y$ ise, $\langle x_n, y_n \rangle \rightarrow \langle x, y \rangle$ 'dir.

Bir X iç çarpım uzayının, aynı cisim üzerinde tanımlı diğer bir \tilde{X} iç çarpım uzayı üzerine olan T eş biçimliliği, iç çarpımı koruyan, yani, her $x, y \in X$ için,

$$\langle Tx, Ty \rangle = \langle x, y \rangle \quad (3.6)$$

olacak şekilde tanımlanan bire-bir ve üzerine bir lineer $T:X \rightarrow X$ operatörüdür. Bu koşullara uygun X ve Y iç çarpım uzaylarına izomorfik iç çarpım uzayları adı verilir [Çakar].

3.1.3. Tamlaştırma

Herhangi bir X iç çarpım uzayı için, bir H Hilbert uzayı ve X 'ten $W \subset H$ yoğun alt uzayı üzerine bir A izomorfizmi vardır. H uzayı, kendisine izomorf olanların dışında değildir.

Bir X iç çarpım uzayının bir Y alt uzayı, X üzerindeki iç çarpım, $Y \times Y$ üzerine kısıtlanmış olarak alınmak üzere, X 'in vektör alt uzayı olarak tanımlanır.

Benzer şekilde, bir H Hilbert uzayının bir Y alt uzayı, H 'nin, bir iç çarpım uzayı olarak düşünülen bir alt uzayı olarak tanımlanır. Bununla birlikte Y 'nin bir Hilbert uzayı olması gerekmez.

O halde, Y bir H Hilbert uzayının bir alt uzayı olsun. Bu durumda,

- (a) Y 'nin tam olması için gerek ve yeter koşul, Y 'nin H uzayında kapalı olmasıdır.
- (b) Y sonlu boyutlu ise, tamdır.
- (c) H ayrılabilir ise, Y de ayrılabilirdir. Daha genel olarak, ayrılabilir bir iç çarpım uzayının her alt kümesi de ayrılabilirdir.

3.2. DOĞURAN KOÇANLI HİLBERT UZAYI

Bir ön-Hilbert uzayı $\{x_k\}_{k=1}^{\infty}$ 'a bir ortonormal tabanı olan bir iç çarpım uzayıdır. $\{x_k\}_{k=1}^{\infty}$ sonsuz kümesinin taban olması için H , en büyük ve en kapsayıcı vektör uzayı olsun. Bu halde aşağıdaki şekilde ifade edilen ve orijinal iç çarpım uzayında bulunması gerekmeyen vektörler $\{x_k\}_{k=1}^{\infty}$ tabanı ile gerilmektedir denir.

$$\mathbf{x} = \sum_{k=1}^{\infty} a_k \mathbf{x}_k \quad (3.7)$$

burada a_k ifadedeki katsayılarıdır. Bir vektör daha tanımlayalım,

$$\mathbf{y}_n = \sum_{k=1}^n a_k \mathbf{x}_k \quad (3.8)$$

y_m gibi bir vektör daha da benzer şekilde tanımlanabilir. $n > m$ için y_n ve y_m vektörleri arasındaki kareli Öklid mesafesini şöyle ifade edebiliriz:

$$\begin{aligned} \|\mathbf{y}_n - \mathbf{y}_m\|^2 &= \left\| \sum_{k=1}^n a_k \mathbf{x}_k - \sum_{k=1}^m a_k \mathbf{x}_k \right\|^2 \\ &= \left\| \sum_{k=m+1}^n a_k \mathbf{x}_k \right\|^2 \\ &= \sum_{k=m+1}^n a_k^2 \end{aligned} \quad (3.9)$$

burada aşağıdaki şartların sağlanıyor olması gerekmektedir.

$$1. \sum_{k=m+1}^n a_k^2 \rightarrow 0 \quad n, m \rightarrow \infty$$

$$2. \sum_{k=1}^m a_k^2 < \infty$$

$\{y_k\}_{k=1}^{\infty}$ vektörlerinin bir dizisi tanımlandığı gibi Cauchy dizisidir. Bu nedenle $\{x_k\}_{k=1}^{\infty}$ tabanında bir x vektörünün tanımlanabilmesi ancak ve ancak x taban vektörlerinin lineer bir kombinasyonu ise ve ilgili katsayılar kare toplanabilir ise mümkündür.

Buradan yola çıkarak tamlık kavramını yeniden ifade edelim [Liu, Principe ve Haykin]:

Bir iç çarpım uzayı H , ele alalım. Eğer bu H uzayından alınan her bir Cauchy dizisi H 'deki bir limite yakınsıyor ise bu uzay tamdır.

3.2.1. Mercer Koçanı

Bir Mercer koçanı sürekli, simetrik ve pozitif tanımlı bir fonksiyondur [Aronszajn]. Bu türde en sık kullanılan koçanlar Gaussçu koçan ve polinomsal koçandır. Bunlar sırasıyla şöyle ifade edilir:

$$\begin{aligned} \kappa(\mathbf{u}, \mathbf{u}') &= \exp(-a\|\mathbf{u} - \mathbf{u}'\|^2) \\ \kappa(\mathbf{u}, \mathbf{u}') &= (\mathbf{u}^T \mathbf{u}' + 1)^p \end{aligned} \quad (3.10)$$

$h(\cdot)$ ve $g(\cdot)$ fonksiyonları, $\kappa(\mathbf{u}, \cdot)$ koçanı ile üretilmiş gerçek değerli fonksiyonlardan oluşan herhangi bir H vektör uzayından alınmış olsun. Bu fonksiyonlar şöyle ifade edilir:

$$h = \sum_{i=1}^l a_i \kappa(\mathbf{c}_i, \cdot) \quad (3.11)$$

$$g = \sum_{j=1}^m b_j \kappa(\tilde{\mathbf{c}}_j, \cdot)$$

burada a_i ve b_j genişleme katsayılarıdır. Bilineer form da şu şekilde tanımlanır:

$$\langle h, g \rangle = \sum_{i=1}^l \sum_{j=1}^m a_i \kappa(\mathbf{c}_i, \tilde{\mathbf{c}}_j) b_j \quad (3.12)$$

ve iç çarpım uzayları konusunda belirtilen özellikleri sağlar.

Sonuç olarak $\langle h, g \rangle$ bilineer formu da bir iç çarpımdır. Ancak iç çarpım özelliklerini takip eden bir özellik daha vardır. Eğer,

$$g(\cdot) = \kappa(\mathbf{u}, \cdot) \quad (3.13)$$

olarak ifade edersek, aşağıda belirtilen doğurma özelliğine ulaşırız:

$$\begin{aligned} \langle h, \kappa(\mathbf{u}, \cdot) \rangle &= \sum_{i=1}^l a_i \kappa(\mathbf{c}_i, \mathbf{u}) \\ &= h(\mathbf{u}) \end{aligned} \quad (3.14)$$

buradaki $\kappa(\mathbf{u}, \mathbf{u}')$ koçanı da aşağıdaki iki koşulu sağlamak koşuluyla H vektör uzayının bir doğuran koçanıdır.

- 1) Her $u \in U$ için, $\kappa(\mathbf{u}, \mathbf{u}')$ koçanı u' vektörünün bir fonksiyonudur ve H 'ye dahildir.
- 2) Doğurma özelliğini sağlamaktadır.

Bu halde, Mercer koçanı yukarıdaki koşulları yerine getirmektedir ve doğuran koçan olarak adlandırılabilir. Eğer doğuran koçan uzayı olarak tanımlanan H iç çarpım uzayı tam ise bu uzaya da doğuran koçanlı Hilbert uzayı adı verilir.

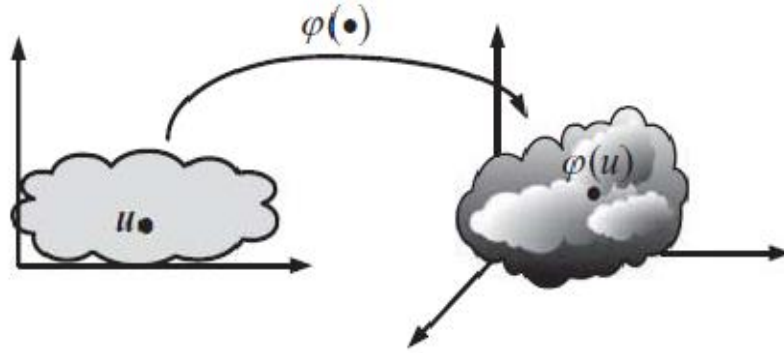
Mercer kuramına göre [Burgess], herhangi bir doğuran koçan aşağıdaki gibi genişletilebilir:

$$K(\mathbf{u}, \mathbf{u}') = \sum_{i=1}^{\infty} \zeta_i \phi_i(\mathbf{u}) \phi_i(\mathbf{u}') \quad (3.15)$$

burada ζ_i ve ϕ_i sırasıyla özdeğerler ve özfonksiyonlardır. Bunun yanında, özdeğerlerin negatif olmayan tanımlı oldukları belirtilmelidir. Bu durumda aşağıdaki haritalama gerçekleştirilebilir:

$$\begin{aligned} \varphi: \mathbb{U} &\rightarrow \mathbb{F} \\ \varphi(\mathbf{u}) &= [\sqrt{\zeta_1} \phi_1(\mathbf{u}), \sqrt{\zeta_2} \phi_2(\mathbf{u}), \dots] \end{aligned} \quad (3.16)$$

Öznilelik uzayı olan F 'nin boyutu kesinlikle pozitif olan özdeğerlerin sayısına belirlenir. Gaussçu koçanda bu sayı sonsuzdur ve dolayısı ile Gaussçu koçan kullanımı girdi uzayını sonsuz boyutta bir öznilelik uzayına haritalar. Bu olgu görsel olarak Şekil 3.1'de temsil edilmektedir.



Şekil 3.1 Girdi uzayından öznitelik uzayı $\varphi(\cdot)$ 'ye doğrusal olmayan haritalamanın görsel bir ifadesi [Liu, Principe ve Haykin]

Burada, hem anlatılanları özetlemek ve hem de netleştirmek amacı ile sayısal bir örneğe yer vermek uygun olacaktır [Gretton].

ϕ , iki boyutlu uzaydaki değişkenleri üç boyutlu öznitelik uzayına haritalayan bir fonksiyon olsun.

$$\phi: \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \phi(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}$$

Bu durumda koçan,

$$k(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} x_1 & x_2 \\ x_1 x_2 \end{bmatrix}^T \begin{bmatrix} y_1 \\ y_2 \\ y_1 y_2 \end{bmatrix}$$

Bu öznitelik uzayını \mathcal{H} ile gösterelim.

Şimdi bu özniteliklerin $(x_1, x_2$ re $x_1x_2)$ bir fonksiyonunu tanımlayalım:

$$f(x) = ax_1 + bx_2 + cx_1x_2$$

Bu fonksiyon $\mathcal{X} = \mathbb{R}^2$ 'ten \mathbb{R} 'ye haritalama yapan fonksiyonlar uzayına dahildir ve eşdeğer bir gösterim ile aşağıdaki gibi ifade edilebilir:

$$f(\cdot) = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Bu gösterim, fonksiyonun kendisini soyut bir şekilde belirtmektedir. Bu gösterimi kullanarak aşağıdaki ifadeyi yazabiliriz:

$$\begin{aligned} f(x) &= f(\cdot)^\top \phi(x) \\ &:= \langle f(\cdot), \phi(x) \rangle_{\mathcal{H}} \end{aligned}$$

Burada, f 'in x 'e göre hesabını öznitelik uzayında bir iç çarpım olarak ifade etmiş oluyoruz.

Daha açık ifade etmek gerekirse,

$$k(\cdot, y) = \begin{bmatrix} y_1 \\ y_2 \\ y_1y_2 \end{bmatrix} = \phi(y)$$

İfadesini ele alalım. Burada yapılan, bir y verildiğinde, \mathcal{H} 'de bir $k(\cdot, y)$ vektörünün elde edilmesidir. Şöyle ki:

$$\langle k(\cdot, y), \phi(x) \rangle_{\mathcal{H}} = ax_1 + bx_2 + cx_1x_2$$

Burada $a = y_1$, $b = y_2$ ve $c = y_1y_2$ 'dir.

Simetri özelliğinden yararlanarak aşağıdaki ifadeyi de yazabiliriz.

$$\begin{aligned} \langle k(\cdot, x), \phi(y) \rangle &= uy_1 + vy_2 + wy_1y_2 \\ &= k(x, y). \end{aligned}$$

Simetri özelliği:

$$\begin{aligned} \phi(x) &= k(\cdot, x) \\ \phi(y) &= k(\cdot, y) \end{aligned}$$

Son olarak, doğurma özelliğini ele alalım.

$$\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \langle f, k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$$

Dolayısı ile $x, y \in \mathcal{X}$ için:

$$k(x, y) = \langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}}$$

3.3. KOÇAN UYARLANABİLİR SÜZGEÇLER

Koçan yöntemi güçlü bir parametrik olmayan modelleme aracıdır. Yöntem kısaca şöyle izah edilebilir: Giriş verisi bir doğuran koçan aracılığı ile yüksek boyutlu bir öznitelik uzayına taşınır. Bu işlem gerçekleştirildikten sonra, taşınmış veri üzerine uygun doğrusal yöntemler uygulanır. Uygulanacak algoritma iç çarpımlar şeklinde formülize edilebiliyor ise yüksek boyutlu öznitelik uzayında, doğrusal olmayan, işlemler yapmaya gerek kalmaz, işlemler doğrusal olarak alçak boyutlu uzayda gerçekleştirilir. Bu olguya koçan hilesi (kernel trick) adı verilmektedir.

Yüksek boyutlu öznitelik uzayında işlem yapılmaması bilgi sayımsal maliyeti düşürmekle kalmaz, bu uzayda işlem yapmanın aşağıda belirtilen sakıncalarından da kurtulmayı sağlar.

- a) Boyut yükseldikçe bilgi sayımsal karmaşıklık çok fazla artmaktadır.
- b) Boyut yükseldikçe genelleştirme performansı düşmektedir ve bu da aşırı uyum riskine yol açmaktadır.

Sayılan bu avantajlar göz önünde bulundurularak bundan sonraki aşamada en temel uyarlanabilir süzgeçler olan en küçük ortalama kareler (LMS – Least Mean Squares) ve yinelemeli en küçük kareler (RLS – Recursive Least Squares), doğuran koçanlı Hilbert uzayında ifade edilecektir.

4. KOÇAN - EN KÜÇÜK ORTALAMA KARELER ALGORİTMASI

Bir önceki bölümde bahsedilen koçan hilesi ile geleneksel en küçük ortalama kareler algoritmasının bir araya getirilmesi, doğuran koçanlı Hilbert uzayında uyarlanabilir bir süzgeç için örneklemeden örnekleme güncelleme gibi ilginç bir özellik ortaya koymaktadır ve bu da – koçan en küçük ortalama kareler (KLMS – Kernel Least Mean Squares) adını almaktadır [Liu]. Koçan en küçük ortalama kareler doğal olarak bir büyüyen radyal taban fonksiyonlu ağ oluşturmaktadır [Pokharel, Liuve Principe]. Bu radyal taban fonksiyonlu ağda, ağ topolojisini öğrenme ve serbest parametreleri doğrudan eğitim verisinden uyarlama gibi özellikler bulunmaktadır ve bu özellikler de hata düzeltme ve bellek tabanlı öğrenme yöntemlerini bir araya getirebilme imkanı sunmaktadır.

4.1. EN KÜÇÜK ORTALAMA KARELER

En küçük ortalama kareler algoritması şu prosedüre dayanmaktadır [Widrow ve Hoff]:

$$\mathbf{w}(0) = 0$$

$$e(i) = d(i) - \mathbf{w}(i-1)^T \mathbf{u}(i)$$

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \eta e(i) \mathbf{u}(i)$$

ve empirik riski minimize eden optimal ağırlığı \mathbf{w}^o bulmak için,

$$\min_{\mathbf{w}} J_{emp}(\mathbf{w}) = \sum_{i=1}^N (d(i) - \mathbf{w}^T \mathbf{u}(i))^2 \quad (4.1)$$

bağıntısı kullanılmaktadır.

En küçük ortalama kereler algoritmasında, $e(i)$ tahmin hatası, η adım büyüklüğü ve $w(i)$ de i anı için hesaplanan optimal ağırlıktır.

Tekrarlanan ağırlık güncelleme uygulaması da şu sonucu doğurur:

$$\mathbf{w}(n) = \eta \sum_{i=1}^n e(i) \mathbf{u}(i) \quad (4.2)$$

burada, n adımlı eğitimden sonra ağırlık hesabı tahmin hataları ile ağırlıklandırılmış önceki ve şu anki giriş verisinin bir kombinasyonu olarak ifade edilmektedir. Ancak koçan hilesini uygulayacağımız ya da doğuran koçanlı Hilbert uzayına geçiş yapacağımız nokta da buradadır. Bu öğrenme algoritmasının giriş-çıkış işlemi iç çarpımlar kullanılarak da aşağıdaki gibi ifade edilebilir:

$$\begin{aligned}
f_n(\mathbf{u}') &= \mathbf{w}(n)^T \mathbf{u}' = \eta \sum_{i=1}^n e(i) \mathbf{u}(i)^T \mathbf{u}' \\
e(n+1) &= d(n+1) - \eta \sum_{i=1}^n e(i) \mathbf{u}(i)^T \mathbf{u}(n+1)
\end{aligned} \tag{4.3}$$

burada koçan hilesi uygulayarak doğuran koçanlı Hilbert uzayına geçiş yaparak yeni algoritmayı oluşturacağız [Pokharel, Liu ve Principe].

4.2. KOÇAN EN KÜÇÜK ORTALAMA KARELER

Veri kümesini, $u(i)$, öznitelik uzayına $\varphi(u(i))$ olarak haritalamak için iç çarpım özelliğinden yararlanacağız. Burada $\varphi(u(i))$ ifadesini kısaca $\varphi(i)$ olarak alırsak, koçan en küçük ortalama kareler (KLMS – Kernel Least Mean Squares) algoritması aşağıda verildiği gibi ifade edilebilir:

$$\begin{aligned}
\omega(0) &= 0 \\
e(i) &= d(i) - \omega(i-1)^T \varphi(i) \\
\omega(i) &= \omega(i-1) + \eta e(i) \varphi(i)
\end{aligned}$$

burada $\omega(i)$, i anında ağırlık vektörünün hesaplanmasıdır ve bu tahmin hesabı doğuran koçanlı Hilbert uzayında gerçekleşmektedir. Algoritmanın iç çarpımlar ile ifade edilebilme özelliğinden faydalanarak,

$$\begin{aligned}
f_n(\mathbf{u}') &= \boldsymbol{\omega}(n)^T \boldsymbol{\varphi}(\mathbf{u}') = \eta \sum_{i=1}^n e(i) \boldsymbol{\varphi}(\mathbf{u}(i))^T \boldsymbol{\varphi}(\mathbf{u}') \\
e(n+1) &= d(n+1) - \eta \sum_{i=1}^n e(i) \boldsymbol{\varphi}(\mathbf{u}(i))^T \boldsymbol{\varphi}(\mathbf{u}(n+1))
\end{aligned} \tag{4.4}$$

burada koçan hilesi kullanarak süzgeç çıkışını ve hataları şöyle hesaplayabiliriz:

$$\begin{aligned}
f_n(\mathbf{u}') &= \boldsymbol{\omega}(n)^T \boldsymbol{\varphi}(\mathbf{u}') = \eta \sum_{i=1}^n e(i) \kappa(\mathbf{u}(i), \mathbf{u}') \\
e(n+1) &= d(n+1) - \eta \sum_{i=1}^n e(i) \kappa(\mathbf{u}(i), \mathbf{u}(n+1))
\end{aligned} \tag{4.5}$$

f_i ile i anındaki giriş-çıkış haritalama hesabını gösterirsek, koçan en küçük ortalama kareler algoritması için sıralı öğrenme kuralımızı şu şekilde tanımlayabiliriz:

$$f_i = f_{i-1} + \eta e(i) \kappa(\mathbf{u}(i), \cdot) \tag{4.6}$$

Görüüleceği üzere, koçan en küçük ortalama kareler algoritmasında, $\mathbf{u}(i)$ merkezli ve $\eta e(i)$ katsayılı her yeni eğitim verisi girişine karşılık yeni bir koçan birimi oluşturmaktadır [Liu, Pokharel ve Principe] ve bu özellik Şekil 7'de gösterilmiştir. Eğitim süreci boyunca, katsayı ve merkezlerin kaydedilmesi gerekmektedir. Bu durumda i anındaki katsayı vektörünü $\mathbf{a}(i)$ ile ve bunlara karşılık gelen merkezleri de $\mathbf{C}(i)$ olarak gösterirsek, i anı için koçan en küçük ortalama kareler algoritması için gerekli güncellemeler şu şekilde özetlenebilir.

$$\mathbf{a}_i(i) = \eta e(i)$$

$$\mathbf{a}_j(i) = \mathbf{a}_j(i-1), j = 1, \dots, i-1$$

$$\mathcal{C}(i) = \{\mathcal{C}(i-1), \mathbf{u}(i)\}$$

Bu aşamada algoritmayı bir bütün olarak Tablo 4.1'den görüldüğü gibi yazabiliriz ve Şekil 4.1'den de olduğu gibi, özellikle her yeni eğitim verisi girişine karşılık yeni bir koçan birimi oluşturma durumunu görsel hale getirebiliriz.

başlangıç

adım boyu η ve koçan κ seç

$\mathbf{a}_1(1) = \eta d(1), \mathcal{C}(1) = \{\mathbf{u}(1)\}, f_1 = \mathbf{a}_1(1)\kappa(\mathbf{u}(1), \cdot)$

hesap

while $\{\mathbf{u}(i), d(i)\}$ *ise* **do**

 % *çıkışı hesapla*

$f_{i-1}(\mathbf{u}(i)) = \sum_{j=1}^{i-1} \mathbf{a}_j(i-1)\kappa(\mathbf{u}(i), \mathbf{u}(j))$

 % *hatayı hesapla*

$e(i) = d(i) - f_{i-1}(\mathbf{u}(i))$

 % *yeni merkezi kaydet*

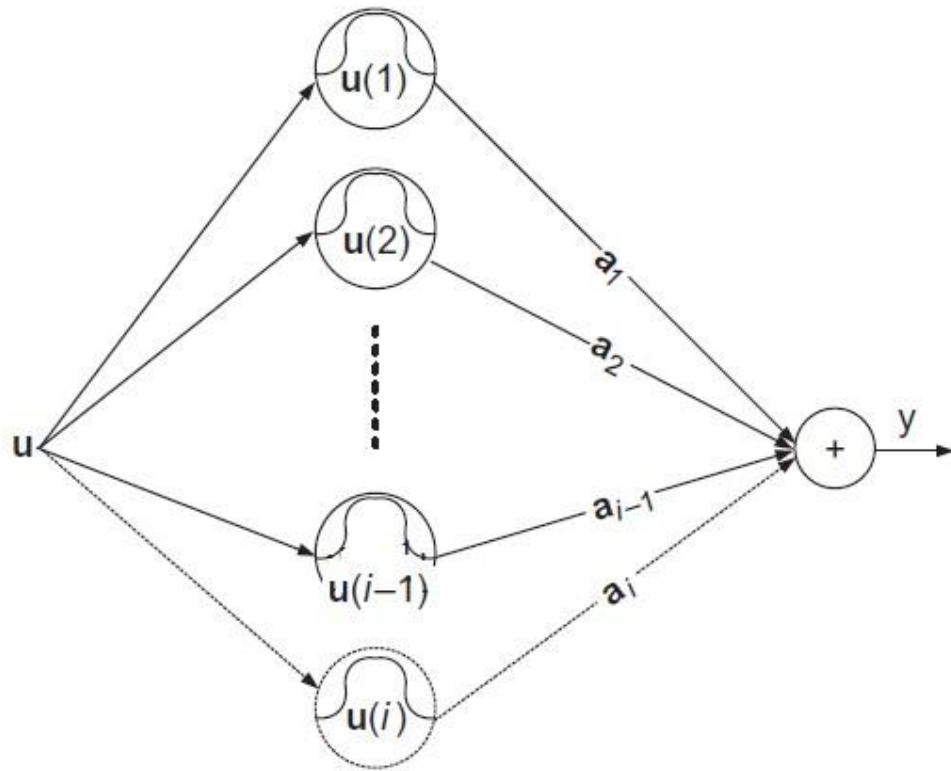
$\mathcal{C}(i) = \{\mathcal{C}(i-1), \mathbf{u}(i)\}$

 % *katsayıyı hesapla ve kaydet*

$\mathbf{a}_i(i) = \eta e(i)$

end while

Tablo 4.1 Koçan en küçük ortalama kareler algoritması [Liu, Principe ve Haykin]



Şekil 4.1 Koçan En Küçük Kareler [Pokharel, Liu ve Principe]

5. KOÇAN - YİNELEMELİ EN KÜÇÜK KARELER ALGORİTMASI

Eğitim veri kümesinden $\{u(j), d(j)\}_{j=1}^{i-1}$ gibi (i-1) anını da içeren bir diziyi göz önünde bulunduralım, burada yinelenen en küçük kareler algoritması $w(i-1)$ ağırlık vektörünü aşağıdaki maliyet fonksiyonunu minimize ederek hesaplar:

$$\min_w \sum_{j=1}^{i-1} |d(j) - \mathbf{u}(j)^T \mathbf{w}|^2 \quad (5.1)$$

burada $u(j)$ $L \times 1$ boyutunda regresör girişi ve $d(j)$ de istenen çıkıştır. Dolayısı ile şu şekilde gösterilebilirler:

$$\mathbf{U}(i-1) = [\mathbf{u}(1), \dots, \mathbf{u}(i-1)]_{L \times (i-1)} \quad (5.2)$$

$$\mathbf{d}(i-1) = [d(1), \dots, d(i-1)]^T \quad (5.3)$$

Belirtilen maliyet fonksiyonunun çözümü ise şöyledir:

$$\mathbf{w}(i-1) = (\mathbf{U}(i-1)\mathbf{U}(i-1)^T)^{-1}\mathbf{U}(i-1)\mathbf{d}(i-1) \quad (5.4)$$

$\{u(i), d(i)\}$ biçiminde bir giriş çıkış ikilisi mevcut ise,

$$\min_{\mathbf{w}} \sum_{j=1}^J |d(j) - \mathbf{u}(j)^T \mathbf{w}|^2 \quad (5.5)$$

maliyet fonksiyonunu minimize eden $w(i)$ ağırlık hesabı şöyle gerçekleştirilir:

$$\mathbf{w}(i) = (\mathbf{U}(i)\mathbf{U}(i)^T)^{-1} \mathbf{U}(i)\mathbf{d}(i) \quad (5.6)$$

burada da,

$$\mathbf{U}(i) = [\mathbf{U}(i-1), \mathbf{u}(i)] \quad (5.7)$$

$$\mathbf{d}(i) = [\mathbf{d}(i-1)^T, d(i)]^T$$

dir.

$$\mathbf{P}(i) = (\mathbf{U}(i)\mathbf{U}(i)^T)^{-1} \quad (5.8)$$

$$\mathbf{P}(i-1) = (\mathbf{U}(i-1)\mathbf{U}(i-1)^T)^{-1}$$

$\mathbf{P}(i)$ ve $\mathbf{P}(i-1)$ matrislerini yukarıdaki gibi tanımladığımızda, aralarındaki ilişkiyi şu şekilde kurabiliriz:

$$\mathbf{P}(i)^{-1} = \mathbf{P}(i-1)^{-1} + \mathbf{u}(i)\mathbf{u}(i)^T \quad (5.9)$$

Matrisin tersi alma lemması,

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1} + \mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}$$

kullanılarak ve aşağıdaki tanımları yaparak,

$$\mathbf{P}(i-1) \rightarrow \mathbf{A}, \mathbf{u}(i) \rightarrow \mathbf{B}, 1 \rightarrow \mathbf{C}, \mathbf{u}(i)^T \rightarrow \mathbf{D} \quad (5.10)$$

doğrudan $\mathbf{P}(i-1)$ 'den $\mathbf{P}(i)$ güncel değerini hesaplamak için aşağıdaki yineleme ifadesini elde ederiz:

$$\mathbf{P}(i) = \left[\mathbf{P}(i-1) - \frac{\mathbf{P}(i-1)\mathbf{u}(i)\mathbf{u}(i)^T\mathbf{P}(i-1)}{1 + \mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)} \right] \quad (5.11)$$

Ayrıca, $w(i)$ 'yi doğrudan önceki değer $w(i-1)$ 'den güncellemek için aşağıdaki yineleme sürecini elde ederiz:

$$\begin{aligned}
\mathbf{w}(i) &= \mathbf{P}(i)\mathbf{U}(i)\mathbf{d}(i) \\
&= \left[\mathbf{P}(i-1) - \frac{\mathbf{P}(i-1)\mathbf{u}(i)\mathbf{u}(i)^T\mathbf{P}(i-1)}{1+\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)} \right] [\mathbf{U}(i-1)\mathbf{d}(i-1) + \mathbf{u}(i)d(i)] \\
&= \underbrace{\mathbf{P}(i-1)\mathbf{U}(i-1)\mathbf{d}(i-1)}_{\mathbf{w}(i-1)} - \frac{\mathbf{P}(i-1)\mathbf{u}(i)\mathbf{u}(i)^T}{1+\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)} \underbrace{\mathbf{P}(i-1)\mathbf{U}(i-1)\mathbf{d}(i-1)}_{\mathbf{w}(i-1)} \\
&\quad + \mathbf{P}(i-1)\mathbf{u}(i)d(i) - \frac{\mathbf{P}(i-1)\mathbf{u}(i)\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)d(i)}{1+\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)} \\
&= \mathbf{w}(i-1) - \frac{\mathbf{P}(i-1)\mathbf{u}(i)\mathbf{u}(i)^T\mathbf{w}(i-1)}{1+\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)} \\
&\quad + \left[\mathbf{P}(i-1)\mathbf{u}(i)d(i) - \frac{\mathbf{P}(i-1)\mathbf{u}(i)\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)d(i)}{1+\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)} \right] \\
&= \mathbf{w}(i-1) - \frac{\mathbf{P}(i-1)\mathbf{u}(i)\mathbf{u}(i)^T\mathbf{w}(i-1)}{1+\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)} + \frac{\mathbf{P}(i-1)\mathbf{u}(i)d(i)}{1+\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)}
\end{aligned}$$

Dolayısı ile,

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \frac{\mathbf{P}(i-1)\mathbf{u}(i)}{1+\mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i)} [d(i) - \mathbf{u}(i)^T\mathbf{w}(i-1)] \quad (5.12)$$

elde etmiş oluruz. Aşağıdaki tanımlamaları yapalım,

$$\begin{aligned}
r(i) &= 1 + \mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i) \\
\mathbf{k}(i) &= \mathbf{P}(i-1)\mathbf{u}(i)/r(i)
\end{aligned}$$

bu durumda standart yinelemeli en küçük kareler (RLS – Recursive Least Squares) algoritmasını elde etmiş oluruz ve bu algoritma kısa şekliyle Tablo 5.1’de gösterilmektedir.

Başlangıç

$$\mathbf{w}(0) = 0, \mathbf{P}(0)$$

Hesap

İlerlet $i \geq 1$

$$\begin{aligned} r(i) &= 1 + \mathbf{u}(i)^T \mathbf{P}(i-1) \mathbf{u}(i) \\ \mathbf{k}(i) &= \mathbf{P}(i-1) \mathbf{u}(i) / r(i) \\ e(i) &= d(i) - \mathbf{u}(i)^T \mathbf{w}(i-1) \\ \mathbf{w}(i) &= \mathbf{w}(i-1) + \mathbf{k}(i) e(i) \\ \mathbf{P}(i) &= [\mathbf{P}(i-1) - \mathbf{k}(i) \mathbf{k}(i)^T r(i)] \end{aligned}$$

Tablo 5.1 Standart Yinelenen En Küçük Ortalama Kareler Algoritması [Liu, Principe ve Haykin]

burada $\mathbf{k}(i)$ kazanç vektörü, ve $e(i)$ tahmin hatasıdır.

Algoritmada $\mathbf{P}(i)$ matrisi veri öz-ilinti matrisinin tersi olarak tanımlanmıştı. Ancak matrisin tersi her koşulda mevcut olmayabilir. Bu durumda regülarizasyon yapmak gereklidir ve dolayısı ile maliyet fonksiyonunu aşağıdaki gibi yeniden düzenlemek gereklidir.

$$\min_{\mathbf{w}} \sum_{j=1}^L |d(j) - \mathbf{u}(j)^T \mathbf{w}|^2 + \lambda \|\mathbf{w}\|^2 \quad (5.13)$$

burada, $\lambda \|\mathbf{w}\|^2$ terimi regülarizasyon terimi olarak adlandırılır ve λ da pozitif bir sayı olmak üzere regülarizasyon parametresi adını alır. Önceki adımlara benzer şekilde çözüm şöyle olur:

$$\mathbf{w}(i) = (\mathbf{U}(i)\mathbf{U}(i)^T + \lambda\mathbf{I})^{-1}\mathbf{U}(i)\mathbf{d}(i)$$

$$\mathbf{P}(i) = (\mathbf{U}(i)\mathbf{U}(i)^T + \lambda\mathbf{I})^{-1}$$

Bu durumda da $\mathbf{P}(i)$ matrisinin tersinin bulunması garanti edilmiş olur.

5.1. ÜSTEL AĞIRLIKLIL YİNELEMELİ EN KÜÇÜK KARELER

Yinelemeli en küçük kareler algoritmasında, yakın geçmişe ait verileri vurgulamak ve uzak geçmişe ait verileri dikkate almamak için üstel ağırlıklandırma yöntemi ortaya atılmıştır [Liu, Principe ve Haykin]. Aşağıdaki aralık ve yakınlıkta bir β sayısı ele alalım.

$$0 \ll \beta \leq 1 \tag{5.14}$$

Bu durumda üstel ağırlıklı regülarize edilmiş yinelemeli en küçük ortalama kareler algoritması için maliyet fonksiyonu:

$$\min_{\mathbf{w}} \sum_{j=1}^L \beta^{L-j} |d(j) - \mathbf{u}(j)^T \mathbf{w}|^2 + \beta^L \lambda \|\mathbf{w}\|^2 \tag{5.15}$$

burada, β genellikle unutmama faktörü olarak adlandırılır. Ayrıca regülarizasyon terimi β , burada β^t olarak ifade edilmiştir ki bu da zaman ilerledikçe regülarizasyonun etkisini azaltma yönünde bir etki yapmaktadır.

Yine benzer şekilde çözüm de aşağıdaki gibi elde edilir:

$$\mathbf{w}(i) = (\mathbf{U}(i)\mathbf{B}(i)\mathbf{U}(i)^T + \beta^t \lambda \mathbf{I})^{-1} \mathbf{U}(i)\mathbf{B}(i)\mathbf{d}(i) \quad (5.16)$$

burada $\mathbf{B}(i)$,

$$\mathbf{B}(i) = \text{diag}\{\beta^{t-1}, \beta^{t-2}, \dots, 1\} \quad (5.17)$$

dir. Aşağıdaki tanımlamaları yaparsak,

$$\begin{aligned} \mathbf{P}(i) &= (\mathbf{U}(i)\mathbf{B}(i)\mathbf{U}(i)^T + \beta^t \lambda \mathbf{I})^{-1} \\ r(i) &= 1 + \beta^{-1} \mathbf{u}(i)^T \mathbf{P}(i-1) \mathbf{u}(i) \\ \mathbf{k}(i) &= \beta^{-1} \mathbf{P}(i-1) \mathbf{u}(i) / r(i) \\ e(i) &= d(i) - \mathbf{u}(i)^T \mathbf{w}(i-1) \end{aligned}$$

ve önceki adımları tekrarlarsak üstel ağırlıklı yinelenen en küçük kareler algoritmasını elde etmiş oluruz ve bu algoritmayı kısaca Tablo 5.2'den görebiliriz.

Başlangıç:

$$\mathbf{w}(0) = 0, \mathbf{P}(0) = \lambda^{-1}\mathbf{I}$$

Hesap

İlerlet $i \geq 1$

$$r(i) = 1 + \beta^{-1} \mathbf{u}(i)^T \mathbf{P}(i-1) \mathbf{u}(i)$$

$$\mathbf{k}(i) = \beta^{-1} \mathbf{P}(i-1) \mathbf{u}(i) / r(i)$$

$$e(i) = d(i) - \mathbf{u}(i)^T \mathbf{w}(i-1)$$

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \mathbf{k}(i) e(i)$$

$$\mathbf{P}(i) = [\beta^{-1} \mathbf{P}(i-1) - \mathbf{k}(i) \mathbf{k}(i)^T r(i)]$$

Tablo 5.2 Üstel Ağırlıklı Yinelemeli En Küçük Kareler Algoritması [Liu, Principe ve Haykin]

5.2. KOÇAN YİNELEMELİ EN KÜÇÜK KARELER

Yinelemeli en küçük kareler algoritmasını doğuran koçanlı Hilbert uzayında gerçekleştirebilmek için koçan en küçük ortalama kareler algoritmasında olduğu gibi yine Mercer kuramından yararlanacağız. Dolayısı ile giriş verisi $\mathbf{u}(i)$ 'yi öznitelik uzayına $\varphi(\mathbf{u}(i))$ olarak taşımak gerekmektedir. $\varphi(\mathbf{u}(i))$ ifadesini yine kısaca $\varphi(i)$ olarak alalım ve yinelemeli en küçük kareler algoritmasını $\{d(1), d(2), \dots\}$, ve $\{\varphi(1), \varphi(2), \dots\}$ örnek dizisi üzerinden kurmaya başlayalım. Her bir iterasyonda, $\omega(i)$ ağırlık vektörü yinelemeli olarak çözülmelidir ve bu ağırlık vektörü aşağıda verilen maliyet fonksiyonunu minimize etmek zorundadır.

$$\min_{\omega} \sum_{j=1}^l |d(j) - \omega^T \varphi(j)|^2 + \lambda \|\omega\|^2 \quad (5.18)$$

Doğuran koçanlı Hilbert uzayı yüksek boyutlu bir uzay olduğundan dolayı, yinelemeli en küçük kareler algoritmasını buraya taşıırken regülarize edilmiş versiyonunu düşünmemiz gerekmektedir. Bununla birlikte $\varphi(j)$ 'nin boyutu da çok yüksek olduğundan (Gauss koçanı kullanıldığında boyut sonsuz olmaktadır) burada farklı bir yol izlemek gerekmektedir.

$$\mathbf{d}(i) = [d(1), \dots, d(i)]^T$$

$$\Phi(i) = [\varphi(1), \dots, \varphi(i)]$$

Bu amaçla, yukarıdaki tanımlamaları yaptığımızda,

$$\omega(i) = [\lambda \mathbf{I} + \Phi(i) \Phi(i)^T]^{-1} \Phi(i) \mathbf{d}(i) \quad (5.19)$$

elde ederiz. Buna ek olarak, yine matris tersi alma lemmasından faydalanarak, ve aşağıdaki belirlemeler ile,

$$\lambda \mathbf{I} \rightarrow \mathbf{A}, \quad \Phi(i) \rightarrow \mathbf{B}, \quad \mathbf{I} \rightarrow \mathbf{C}, \quad \Phi(i)^T \rightarrow \mathbf{D}$$

aşağıdaki ifadeyi doğrulayabiliriz:

$$[\lambda \mathbf{I} + \Phi(i)\Phi(i)^T]^{-1}\Phi(i) = \Phi(i)[\lambda \mathbf{I} + \Phi(i)^T\Phi(i)]^{-1} \quad (5.20)$$

Bu ifadeyi, yukarıdaki ağırlık ifadesinde yerine koyduğumuzda,

$$\omega(i) = \Phi(i)[\lambda \mathbf{I} + \Phi(i)^T\Phi(i)]^{-1}\mathbf{d}(i) \quad (5.21)$$

elde ederiz. Burada kazancımız $\Phi(i)^T\Phi(i)$ çarpımının koçan hilesi kullanılarak hesaplanabilmesi ve ağırlığın giriş verisinin lineer bir kombinasyonu olarak kesin bir şekilde ifade edilebilmesidir.

$$\omega(i) = \Phi(i)\mathbf{a}(i) \quad (5.22)$$

burada,

$$\mathbf{a}(i) = [\lambda \mathbf{I} + \Phi(i)^T\Phi(i)]^{-1}\mathbf{d}(i) \quad (5.23)$$

dir. Ayrıca,

$$\mathbf{Q}(i) = [\lambda \mathbf{I} + \Phi(i)^T\Phi(i)]^{-1} \quad (5.24)$$

ifadesinden de,

$$\mathbf{Q}(i)^{-1} = \begin{bmatrix} \mathbf{Q}(i-1)^{-1} & \mathbf{h}(i) \\ \mathbf{h}(i)^T & \lambda + \boldsymbol{\varphi}(i)^T \boldsymbol{\varphi}(i) \end{bmatrix} \quad (5.25)$$

kolayca elde edilebilmektedir. Burada,

$$\mathbf{h}(i) = \boldsymbol{\Phi}(i-1)^T \boldsymbol{\varphi}(i) \quad (5.26)$$

dir. Bu kayan pencere yapısını kullanarak, bu büyüyen matrisin tersinin güncellenmesi aşağıdaki gibi mümkün hale gelmektedir.

$$\mathbf{Q}(i) = r(i)^{-1} \begin{bmatrix} \mathbf{Q}(i-1)r(i) + \mathbf{z}(i)\mathbf{z}(i)^T & -\mathbf{z}(i) \\ -\mathbf{z}(i)^T & 1 \end{bmatrix} \quad (5.27)$$

burada da,

$$\begin{aligned} \mathbf{z}(i) &= \mathbf{Q}(i-1)\mathbf{h}(i) \\ r(i) &= \lambda + \boldsymbol{\varphi}(i)^T \boldsymbol{\varphi}(i) - \mathbf{z}(i)^T \mathbf{h}(i) \end{aligned} \quad (5.28)$$

dir. $\mathbf{Q}(i)$ ifadesi blok matrisin tersi tanımıyla elde edilir.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{bmatrix}$$

burada, A ve D karesel matris bloklarıdır.

Bunun sonucu olarak, ağırlık ifadesindeki genişleme katsayılarının hesaplanması aşağıdaki gibi gerçekleştirilebilir.

$$\begin{aligned} \mathbf{a}(i) &= \mathbf{Q}(i)\mathbf{d}(i) \\ &= \begin{bmatrix} \mathbf{Q}(i-1) + \mathbf{z}(i)\mathbf{z}(i)^T r(i)^{-1} & -\mathbf{z}(i)r(i)^{-1} \\ -\mathbf{z}(i)^T r(i)^{-1} & r(i)^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{d}(i-1) \\ d(i) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{a}(i-1) - \mathbf{z}(i)r(i)^{-1}e(i) \\ r(i)^{-1}e(i) \end{bmatrix} \end{aligned}$$

burada $e(i)$ tahmin hatasıdır ve aşağıdaki gibi hesaplanmaktadır.

$$\begin{aligned} f_{i-1}(\mathbf{u}(i)) &= h(i)^T \mathbf{a}(i-1) = \sum_{j=1}^{i-1} \mathbf{a}_j(i-1) \kappa(\mathbf{u}(j), \mathbf{u}(i)) \\ e(i) &= d(i) - f_{i-1}(\mathbf{u}(i)) \end{aligned} \quad (5.29)$$

Sonuç olarak, koçan yinelemeli en küçük kareler algoritması ile her bir iterasyonda genişleyen bir radyal taban fonksiyonlu ağ teşkil etmiş bulunmaktayız. O halde öğrenme sürecinin koçan en küçük ortalama kareler algoritmasında olduğu gibi her iterasyonda $\mathbf{u}(i)$ merkezli $r(i)^{-1}e(i)$ katsayılı yeni bir birim oluşturulmaktadır. Koçan en küçük ortalama kareler algoritmasından farklı olarak ise burada, tüm geçmiş katsayılar $-\mathbf{z}(i) r(i)^{-1}e(i)$ ile

güncellenmektedir, nitekim koçan en küçük ortalama kareler algoritmasında geçmiş katsayıların güncellenmesi olgusu mevcut değildir [Engel, Mannorve Meir].

Eğer i anındaki giriş çıkış haritalamasını f_i ile ifade edersek, aşağıdaki sıralı öğrenme kuralını elde ederiz.

$$f_i = f_{i-1} + r(i)^{-1} \left[\kappa(\mathbf{u}(i), \cdot) - \sum_{j=1}^{i-1} \mathbf{z}_j(i) \kappa(\mathbf{u}(j), \cdot) \right] e(i) \quad (5.30)$$

Buradaki $\mathbf{a}(i)$ katsayıları ve $C(i)$ merkezleri eğitim süreci boyunca kayıt edilmelidir. Sonuç olarak i . iterasyonda yapılması gereken güncellemeler de şöyle listelenebilir:

$$\begin{aligned} \mathbf{a}_i(i) &= r(i)^{-1} e(i) \\ \mathbf{a}_j(i) &= \mathbf{a}_j(i-1) - r(i)^{-1} e(i) \mathbf{z}_j(i), \quad j = 1, \dots, i-1 \\ C(i) &= \{C(i-1), \mathbf{u}(i)\} \end{aligned}$$

Dolayısı ile i iterasyonunda bir \mathbf{u} test girdisi uygulanması halinde süzgecin çıkışı da şu şekilde olur:

$$f(\mathbf{u}_*) = \sum_{j=1}^i \mathbf{a}_j(i) \kappa(\mathbf{u}(j), \mathbf{u}_*) \quad (5.31)$$

Koçan yinelemeli en küçük kareler algoritmasını özetlersek Tablo 5.3'ü elde etmiş oluruz.

Başlangıç

$$\mathbf{Q}(1) = (\lambda + \kappa(\mathbf{u}(1), \mathbf{u}(1)))^{-1}, \mathbf{a}(1) = \mathbf{Q}(1)d(1)$$

Hesap

İlerlet $i > 1$:

$$\begin{aligned} \mathbf{h}(i) &= [\kappa(\mathbf{u}(i), \mathbf{u}(1)), \dots, \kappa(\mathbf{u}(i), \mathbf{u}(i-1))]^T \\ \mathbf{z}(i) &= \mathbf{Q}(i-1)\mathbf{h}(i) \\ r(i) &= \lambda + \kappa(\mathbf{u}(i), \mathbf{u}(i)) - \mathbf{z}(i)^T \mathbf{h}(i) \\ \mathbf{Q}(i) &= r(i)^{-1} \begin{bmatrix} \mathbf{Q}(i-1)r(i) + \mathbf{z}(i)\mathbf{z}(i)^T & -\mathbf{z}(i) \\ -\mathbf{z}(i)^T & 1 \end{bmatrix} \\ e(i) &= d(i) - \mathbf{h}(i)^T \mathbf{a}(i-1) \\ \mathbf{a}(i) &= \begin{bmatrix} \mathbf{a}(i-1) - \mathbf{z}(i)r(i)^{-1}e(i) \\ r(i)^{-1}e(i) \end{bmatrix} \end{aligned}$$

Tablo 5.3 Koçan Yinelemeli En Küçük Kareler Algoritması [Liu, Principe ve Haykin]

5.3. ÜSTEL AĞIRLIKLI KOÇAN YİNELEMELİ EN KÜÇÜK KARELER

Benzer bir şekilde unutma faktörünü de algoritmaya dahil edebiliriz. Böylece regülarizasyonla beraber uzak geçmişe ait verilere da daha az önem vererek bilgi sayımı hızlandırabiliriz. Bu durumda maliyet fonksiyonu aşağıdaki gibi olacaktır:

$$\min_{\boldsymbol{\omega}} \sum_{j=1}^l \beta^{l-j} |d(j) - \boldsymbol{\omega}^T \boldsymbol{\varphi}(j)|^2 + \beta^l \lambda \|\boldsymbol{\omega}\|^2 \quad (5.32)$$

ve çözümü de,

$$\boldsymbol{\omega}(i) = [\lambda \beta^i \mathbf{I} + \boldsymbol{\Phi}(i) \mathbf{B}(i) \boldsymbol{\Phi}(i)^T]^{-1} \boldsymbol{\Phi}(i) \mathbf{B}(i) \mathbf{d}(i) \quad (5.33)$$

dir. Sonuç olarak özet halinde yazılır ise, algoritma Tablo 5.4'te gösterildiği duruma gelir.

Başlangıç

$$\mathbf{Q}(1) = (\lambda \beta + \kappa(\mathbf{u}(1), \mathbf{u}(1)))^{-1}, \mathbf{a}(1) = \mathbf{Q}(1) \mathbf{d}(1)$$

Hesap

İlerlet $i > 1$:

$$\begin{aligned} \mathbf{h}(i) &= [\kappa(\mathbf{u}(i), \mathbf{u}(1)), \dots, \kappa(\mathbf{u}(i), \mathbf{u}(i-1))]^T \\ \mathbf{z}(i) &= \mathbf{Q}(i-1) \mathbf{h}(i) \\ r(i) &= \lambda \beta^i + \kappa(\mathbf{u}(i), \mathbf{u}(i)) - \mathbf{z}(i)^T \mathbf{h}(i) \\ \mathbf{Q}(i) &= r(i)^{-1} \begin{bmatrix} \mathbf{Q}(i-1) r(i) + \mathbf{z}(i) \mathbf{z}(i)^T & -\mathbf{z}(i) \\ -\mathbf{z}(i)^T & 1 \end{bmatrix} \\ e(i) &= d(i) - \mathbf{h}(i)^T \mathbf{a}(i-1) \\ \mathbf{a}(i) &= \begin{bmatrix} \mathbf{a}(i-1) - \mathbf{z}(i) r(i)^{-1} e(i) \\ r(i)^{-1} e(i) \end{bmatrix} \end{aligned}$$

Tablo 5.4 Üstel Ağırlıklı Koçan Yinelemeli En Küçük Kareler Algoritması [Liu, Principe ve Haykin]

6. ÖRNEK UYGULAMA VE BİLGİSAYAR BENZETİM SONUÇLARI

Bu bölümde, önceki kısımlarda anlatılan ve en yaygın kullanılan doğrusal süzgeçler olan ortalama en küçük kareler ve yinelenen en küçük kareler algoritmalarının doğrusal olmayan versiyonları için bilgisayar benzetimleri yapılmıştır. Önceden de söz edildiği üzere, bu süzgeçlerin doğrusal olmayan karşılıkları doğuran koçanlı Hilbert uzayında, koçan hilesi yöntemi vasıtasıyla gerçekleştirilmiştir.

Bu algoritmaları kullanarak, örnek uygulamada, bir zaman serisi olan bilginin bir kanal üzerinden iletilmesi ve iletimin ardından kanal denkleştirme işlemi uygulanarak, yeniden elde edilmesi yani tahmini hedeflenmiştir. Bu doğrultuda uygulama ve benzetim çalışmaları Matlab ortamında gerçekleştirilmiştir.

Bu yöntemlerin sınındığı benzetimler için iki farklı zaman serisi kullanılmıştır. Bu zaman serileri belirlenirken, haberleşme işaretlerinde görülen; düzensizlik, geniş bantlılık, aperyodiklik ve uzun zaman aralıkları için tahmin edilemezlik gibi özellikleri yansıttığı için kaotik zaman serilerine başvurulmuştur. Her ikisi de kaotik özellik gösteren dinamik sistemlerden elde edilen bu zaman serileri Mackey-Glass ve Lorenz sistemleridir.

Süzgeç çıkışında elde edilebilmesi istenen bu zaman serileri ile süzgeç çıkışı karşılaştırılmış ve bu karşılaştırma da ortalama kare hata olarak belirtilmiştir. Ortalama hatanın karesi, istatistiksel açıdan bakıldığında varyans ve standart

sapma parametrelerine bağılı olarak elde edildiğinden tahminlerin kıyaslanmasında geçerli bir tekniktir. Ayrıca kare hatanın toplam değeri de, diğeri bir başarıml ölçütü olarak her bir uygulama için hesaplanmıştır. Sonuç olarak hata parametresinin sifira yaklaşması tahminin değeri gerçekte değere ne kadar yakın olduğunu göstermektedir.

Bunun yanı sıra, haberleşme kanalları göz önüne alındığında, uygun bir model seçerken, verici ile alıcının birbirini görmediği durumlar, dolayısı ile alınan işaretin genlik ve fazında büyük değişimlerin meydana gelebileceği durumlar öngörülmüştür. Bu tür durumlarda alınan işaretin genlik ve fazındaki değişimler, 2. Bölümde matematiksel olarak ifade edilen Rayleigh olasılık dağılım fonksiyonu ile modellenenmektedir.

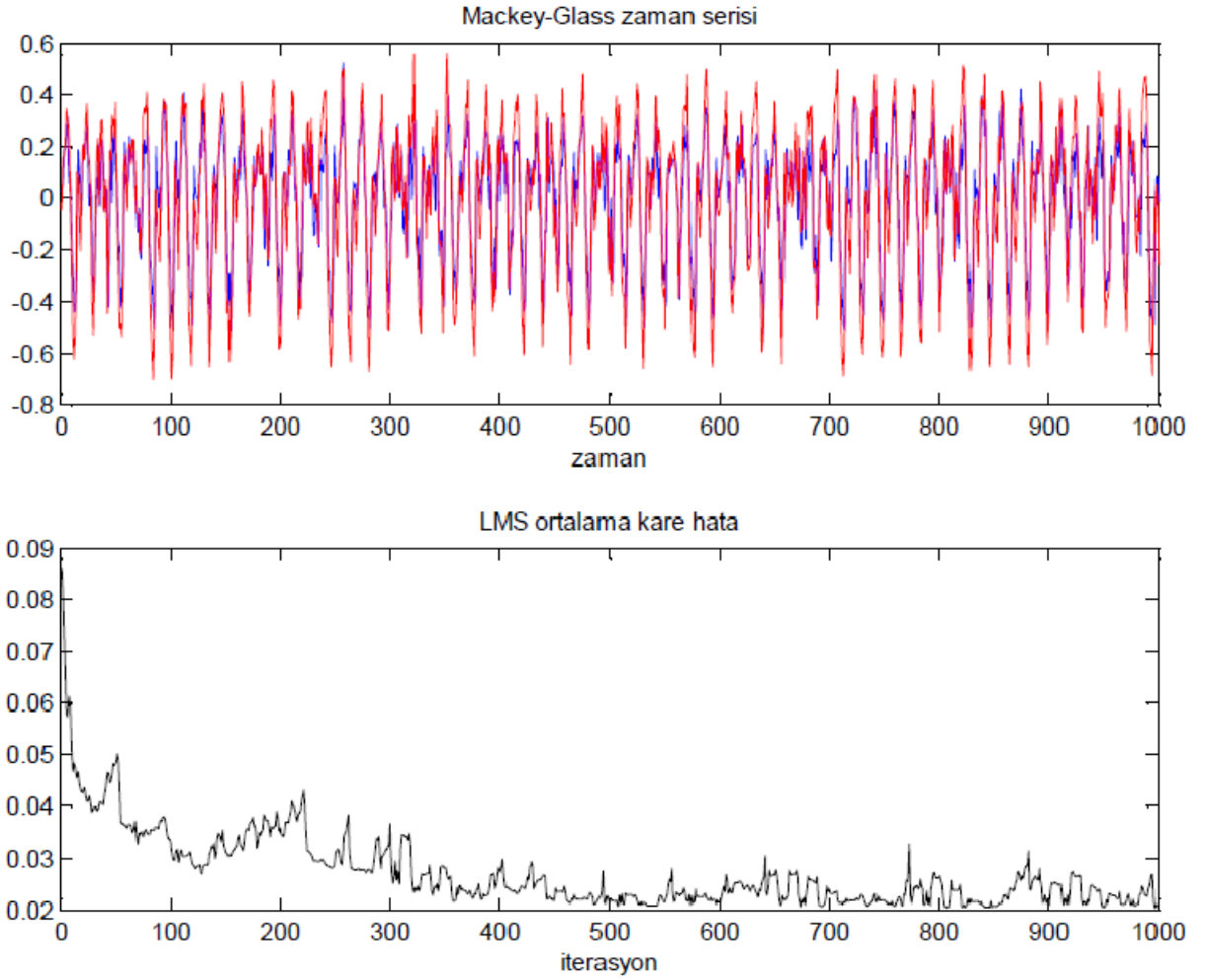
Bu nedenle, model olarak bir Rayleigh kanal modeli kullanılarak geleneksel uyarlanabilir kanal denkleştirme algoritmalarını bu çalışmada önerilen ve en iyi sonuç verdiği düşünülen, koçan uyarlanabilir denkleştirici (koçan yinelenen en küçük kareler algoritması) ile karşılaştırmak amacı ile test edilerek, desibel cinsinden performans sonuçları ortalama ve standart sapma olarak belirtilmiştir.

Böylelikle en küçük ortalama kareler (LMS – Least Mean Squares), koçan en küçük ortalama kareler (KLMS – Kernel Least Mean Squares), yinelenen en küçük kareler (RLS – Recursive Least Squares) ve koçan yinelenen en küçük kareler (KRLS – Kernel Recursive Least Squares) yöntemleri sırasıyla Mackey-Glass ve Lorenz sistemlerine uygulanmıştır ve tüm sonuç ve karşılaştırmalar Şekil 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7 ve 6.8’te belirtilen sırayla gösterilmiştir.

Son aşamada ise Tablo 6.1’de ve Tablo 6.2’de, zaman serileri üzerinden karşılaştırma yapmak için toplam kare hatalar bir arada gösterilmiştir. Burada özellikle kaotik davranışı daha belirsiz olan [Özer ve Zorlu] Lorenz zaman serisi üzerinde yüksek performans elde edilmiştir.

Zaman serileri üzerinden görülen ortalama karesel hataların yanı sıra, karşılaştırma amacı ile toplam kare hatalar da hesaplanmış ve Tablo 6.1’de verilen ilk karşılaştırma tablosu oluşturulmuştur. Buna göre sıfıra yakınlık başarı ölçütüdür; dolayısı ile en başarılı sonuç koçan yinelenen en küçük kareler algoritması ile elde edilmiştir.

Buna göre beklediğimiz gibi, koçan uyarlanabilir süzgeçler doğrusal karşılıklarına göre çok daha iyi performans sergilemekte, doğrusal süzgeçlerde nasıl yinelenen en küçük kareler algoritması en küçük ortalama kareler algoritmasından daha başarılı performans gösteriyor ise, doğrusal olmayan – koçan uyarlanabilir süzgeçlerde de koçan yinelenen en küçük kareler algoritması koçan en küçük ortalama kareler algoritmasından daha başarılı performans göstermektedir.

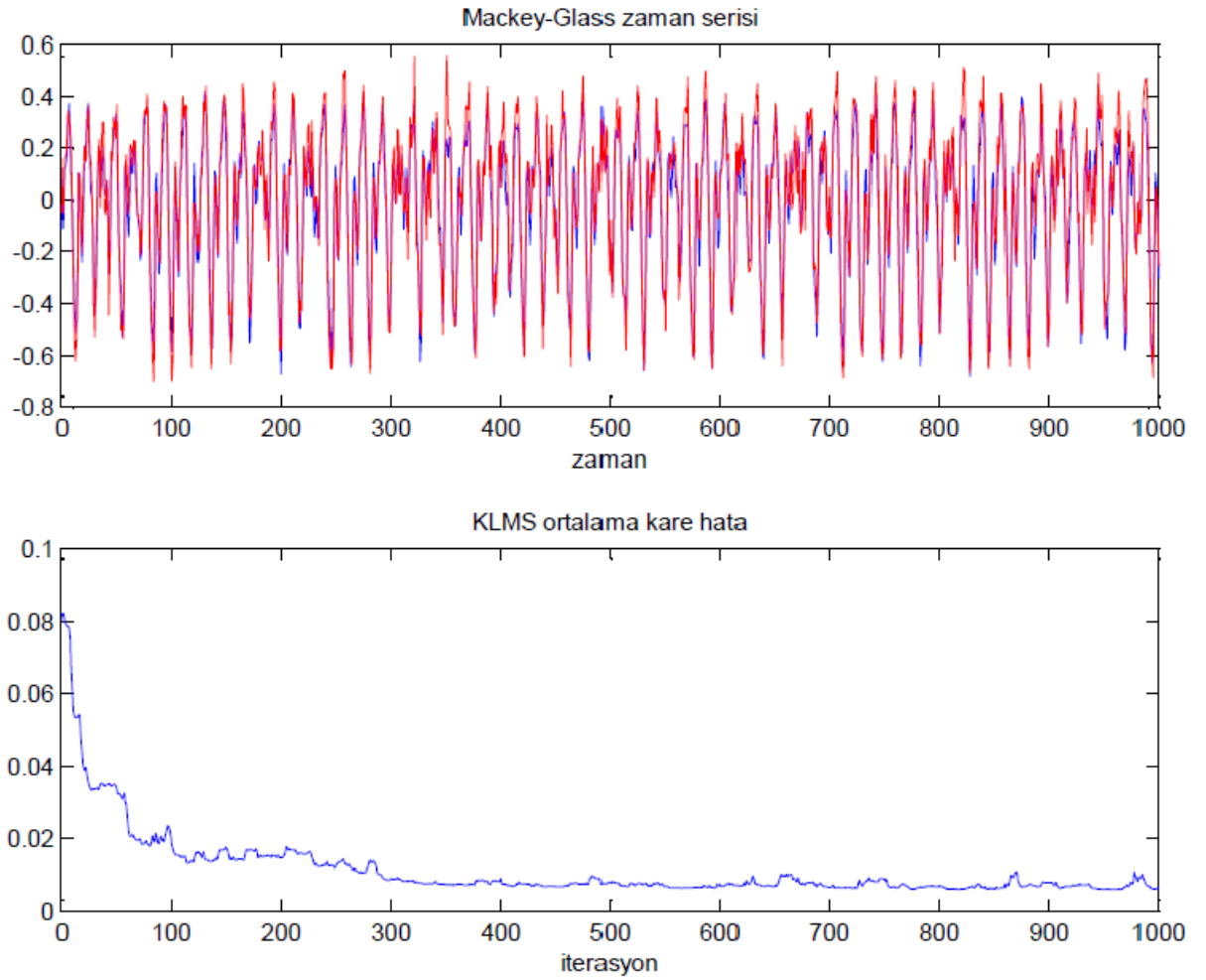


Şekil 6.1

(a) Mackey-Glass zaman serisi ve en küçük ortalama kareler yöntemi ile tahmini

(b) Ortalama kare hata

Toplam kare hata = 27.1431

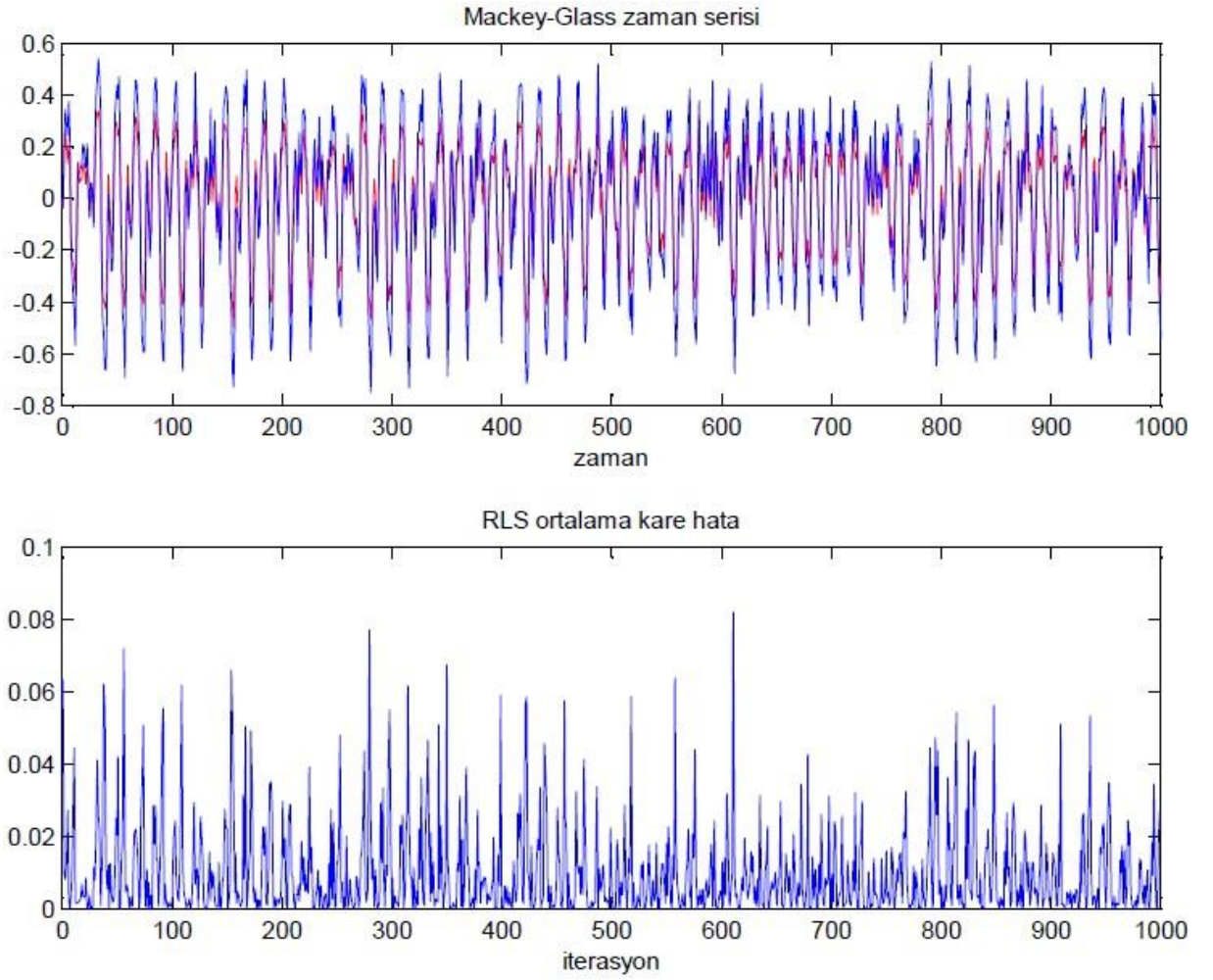


Şekil 6.2

(a) Mackey-Glass zaman serisi ve koçan en küçük ortalama kareler yöntemi ile tahmini

(b) Ortalama kare hata

Toplam kare hata = 11.2618

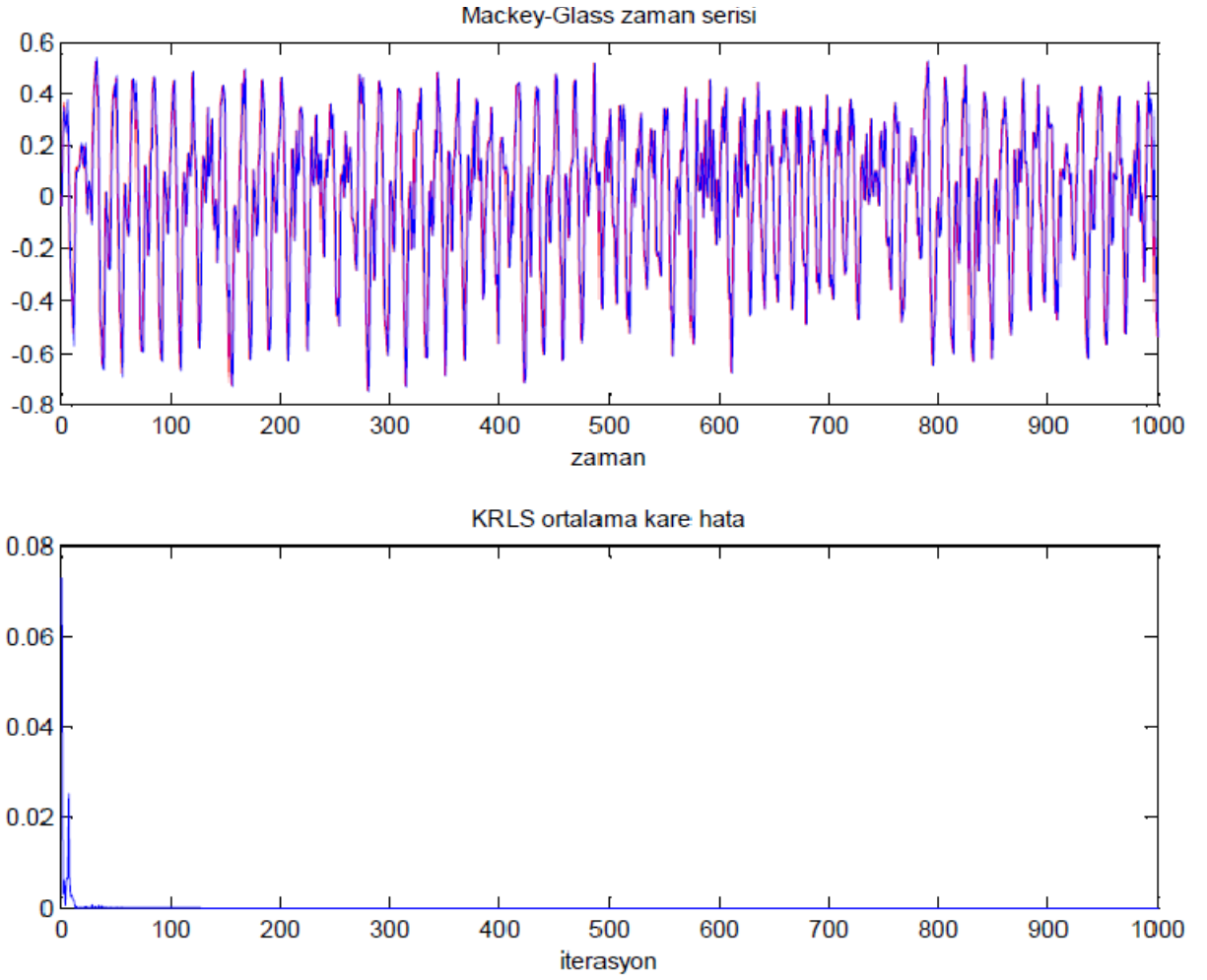


Şekil 6.3

(a) Mackey-Glass zaman serisi ve yinelenen en küçük kareler yöntemi ile tahmini

(b) Ortalama kare hata

Toplam kare hata = 10.0082

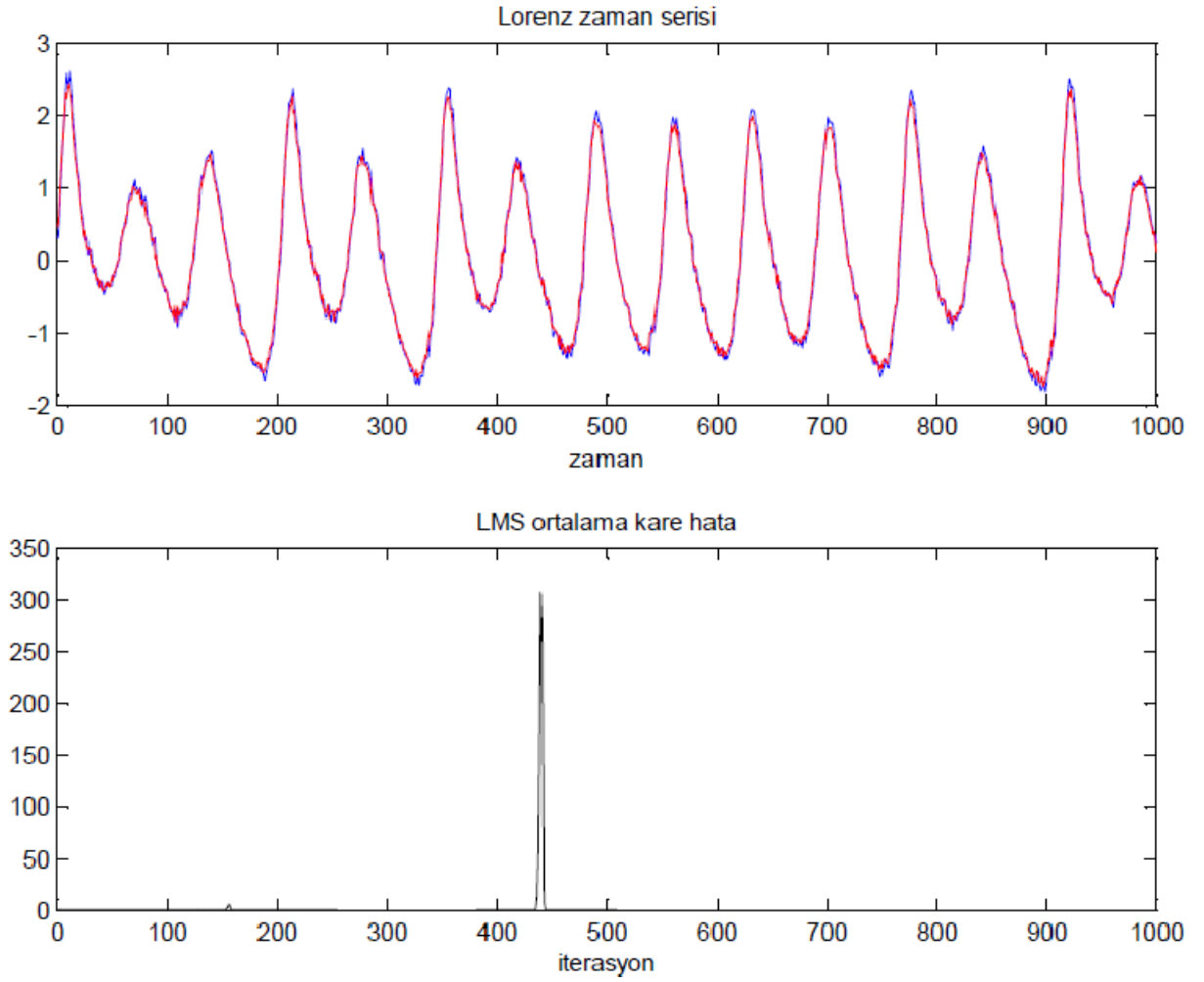


Şekil 6.4

(a) Mackey-Glass zaman serisi ve koçan yinelenen en küçük kareler yöntemi ile tahmini

(b) Ortalama kare hata

Toplam kare hata = 0.1454

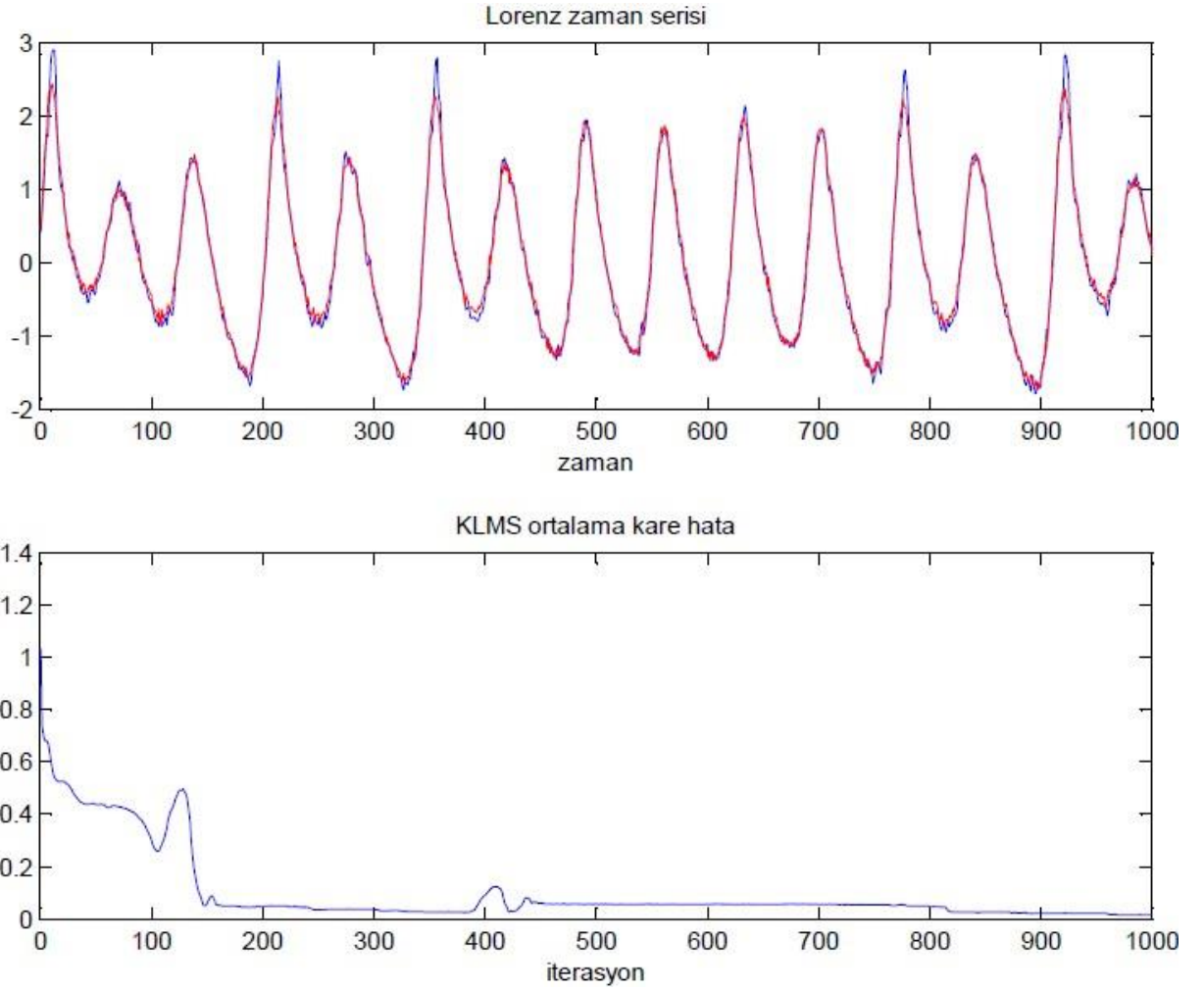


Şekil 6.5

(a) Lorenz zaman serisi ve en küçük ortalama kareler yöntemi ile tahmini

(b) Ortalama kare hata

Toplam kare hata = 972.4862

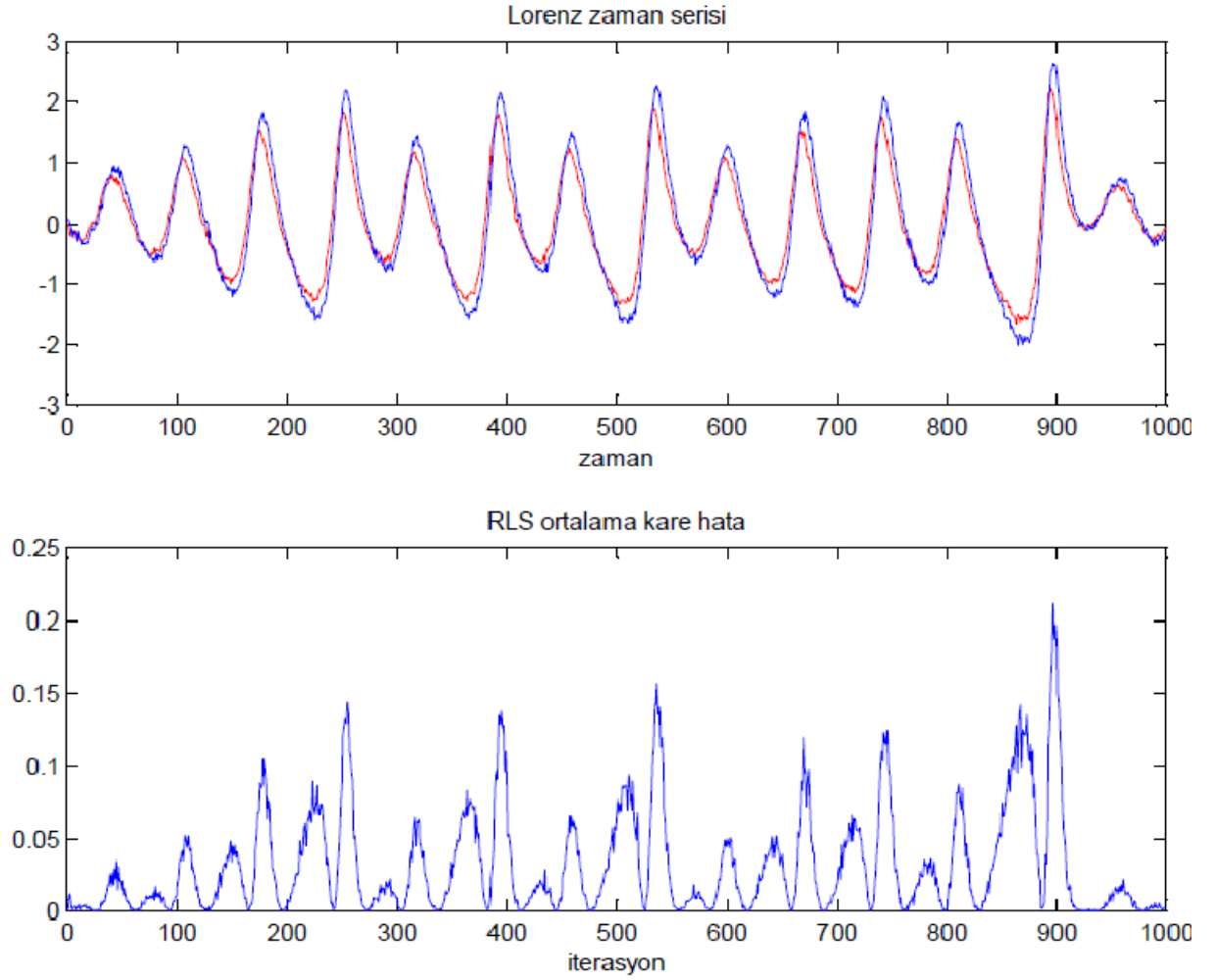


Şekil 6.6

(a) Lorenz zaman serisi ve koçan en küçük ortalama kareler yöntemi ile tahmini

(b) Ortalama kare hata

Toplam kare hata = 99.9426

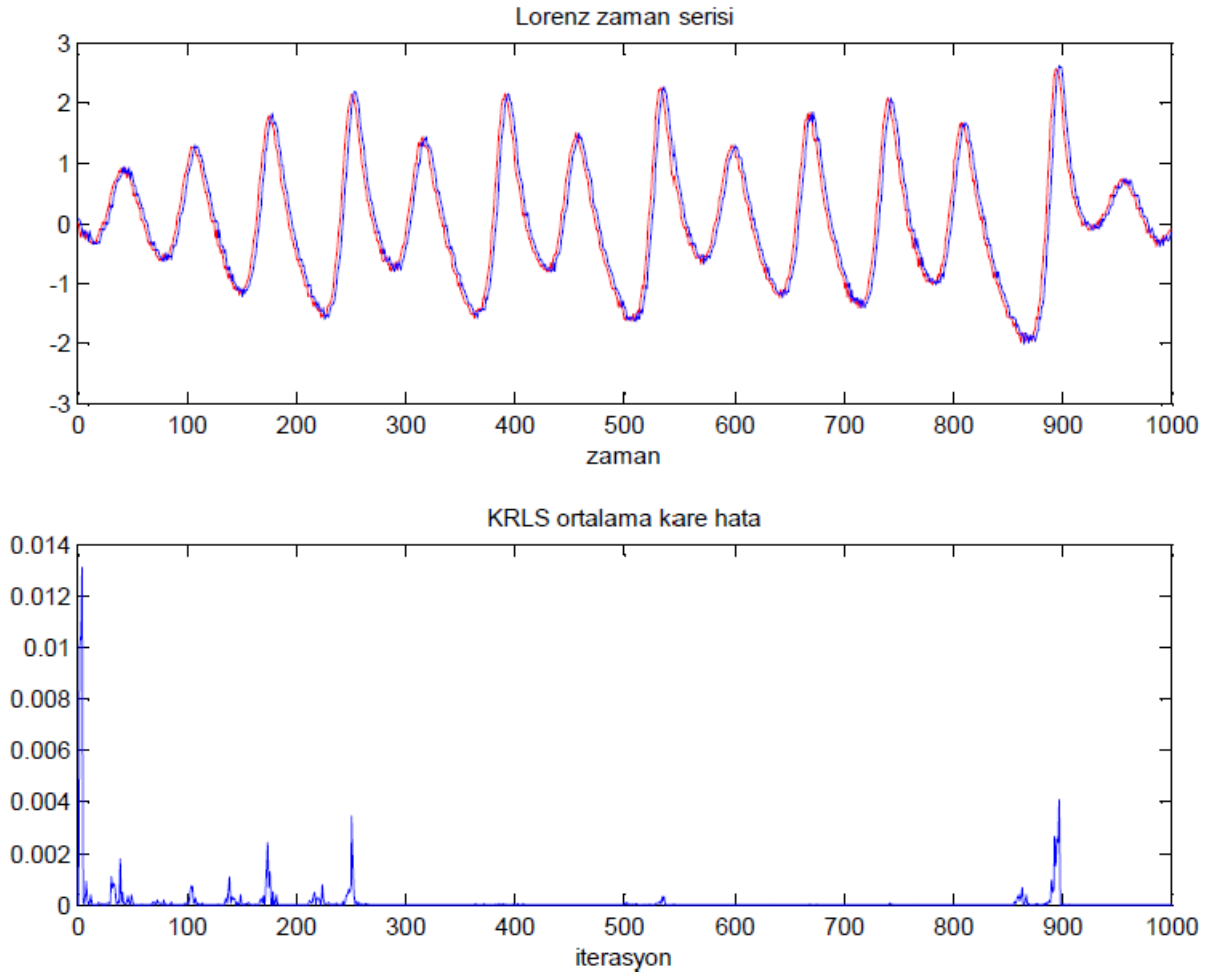


Şekil 6.7

(a) Lorenz zaman serisi ve yinelenen en küçük kareler yöntemi ile tahmini

(b) Ortalama kare hata

Toplam kare hata = 31.0842



Şekil 6.8

(a) Lorenz zaman serisi ve koçan yinelenen en küçük kareler yöntemi ile tahmini

(b) Ortalama kare hata

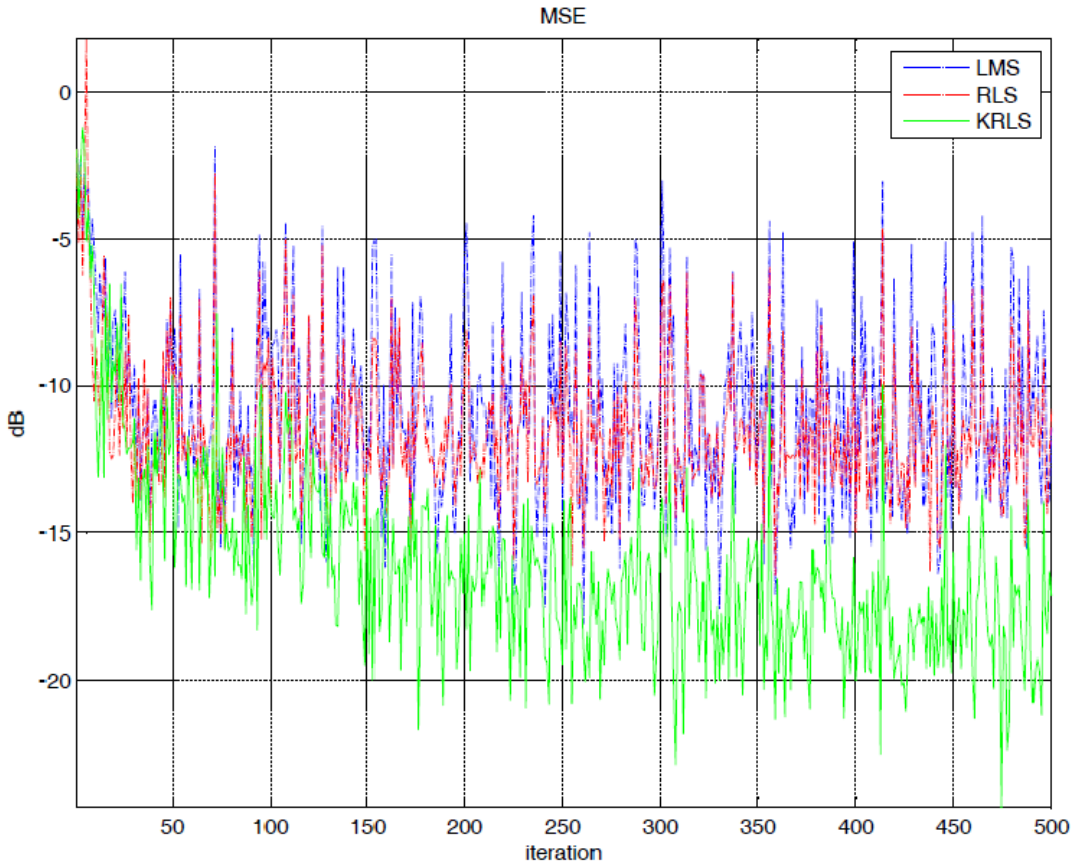
Toplam kare hata = 0.0966

Süzgeç	Toplam Kare Hata (Lorenz serisi)	Toplam Kare Hata (Mackey-Glass serisi)
EKOK (LMS)	972.4862	27.1431
KEKOK (KLMS)	99.9426	11.2618
YEKK (RLS)	31.0842	10.0082
KYEKK (KRLS)	0.0966	0.1454

Tablo 6.1 Karşılaştırma Tablosu

Bu uygulamayı genişleterek, haberleşme kanalları ön plana alınmış ve kanal modeli olarak bir Rayleigh kanal modeli kullanılarak geleneksel uyarlanabilir kanal denkleştirme algoritmalarını bu çalışmada önerilen ve en iyi sonuç verdiği gözlemlenen, koçan uyarlanabilir denkleştirici algoritması (koçan yinelenen en küçük kareler algoritması) ile karşılaştırmak amacı ile tekrar bir teste tabi tutularak desibel cinsinden performans sonuçlarını ortalama ve

standart sapma olarak Tablo 6.2’de belirtilmiş ve Şekil 6.9’da gösterilmiştir. Bu testte, farklı rasgele değerler ile kanal tahmin edilmeye çalışılmış, bu farklı rasgele değerlerden oluşan 4 farklı deney sonucu alınmıştır. Hatanın desibel cinsinden hesaplanmasının amacı farklı tahmin algoritmalarının hatalarını rakamsal olarak karşılaştırılabilir hale getirmektir, dolayısıyla dB değeri düştükçe başarımlar – performans yükselmektedir. Şekil 6.9’dan da net olarak görülebildiği gibi doğrusal algoritmalar ile karşılaştırıldığında koçan yinelenen en küçük kareler algoritmasında hata oldukça düşük seviyelere inmektedir. Bununla birlikte, Tablo 2’deki sonuçlardan yararlanarak da koçan en küçük yinelenmeli kareler algoritmasının farklı rasgele değerler ile çalıştırılan 4 farklı deney sonucunda da oldukça düşük hata değerleri verdiği görülebilmektedir.



Şekil 6.9 Rayleigh kanal modelinde dB cinsinden ortalama karesel hatalar üzerinden performans karşılaştırması

	EKOK (LMS) ort.± std sapma (dBort.kare hata)	YEKK (RLS) ort.± std sapma (dBort.kare hata)	KYEKK (KRLS) ort.± std sapma (dBort.kare hata)
Deney 1	-11.763±3.0606	-11.8272±1.8903	-17.9969±2.0688
Deney 2	-12.9455±3.1145	-13.295±2.0884	-19.5618±2.3944
Deney 3	-11.012±2.7123	-11.4986±1.6779	-17.2634±2.1297
Deney 4	-9.7851±2.6688	-10.2905±1.5668	-16.0566±1.7728

Tablo 6.2 Performans Karşılaştırma Tablosu

7. SONUÇ

Doğrusal olmayan uyarlanabilir süzgeçleme konusu henüz doğrusal uyarlanabilir süzgeçleme gibi bir doyuma ulaşmamış değildir, gelişimini büyük bir ivmeyle sürdüren ve üzerinde birçok çalışma gerçekleştirilen bir konudur. Bu görüş açısından bakılacak olursa, son yıllarda öne çıkan koçan tapanlı yöntemler, hem doğrusal olmayan yöntemler oluşu ve hem de sağlam matematiksel temellere dayanıyor oluşu gibi nedenlerden ötürü, bu çalışmada tercih edilen yöntem koçan tapanlı denkleştirme algoritması olmuştur. Nitekim, bu nedenlerden ötürü de son zamanlarda yapılan birçok araştırma ve uygulama için bu konu oldukça çekici olmuştur.

Bu çalışmada, kanal denkleştirme problemi özelinde, doğrusal koçanlı Hilbert uzayında uyarlanabilir süzgeçleme algoritmaları incelenmiştir. Kanal denkleştirme problemi, bir haberleşme sisteminde alıcıda elde edilen işaretin kanalın tersinden tekrar geçirilmesi, dolayısı ile alıcıda kanal bilgisinin ya da, doğrusal koçanlı algoritmasının tersini elde edilmesi gerekmektedir. Ayrıca, haberleşme kanalının zamanla değişmesi durumunda, kanal denkleştiricinin kanaldaki değişimleri hızlı bir şekilde takip edebilmesi gerekmektedir.

Doğrusal koçanlı Hilbert uzayı, koçan hilesi kullanmak suretiyle iç çarpımlar olarak ifade edilen doğrusal uyarlanabilir süzgeç algoritmalarının doğrusal olmayan karşılıklarının elde edilebilmesi için bir yöntem önermektedir. Dolayısıyla, en iyi bilinen uyarlanabilir süzgeçleme yöntemleri olan en küçük ortalama kareler ve yinelenen en küçük kareler algoritmalarının koçan uygulamaları sunulmuş, uygulama olarak kullanılmış, algoritma performansları değerlendirilmiştir ve bir Rayleigh kanalında denkleştirme probleminin çözümü için performanslar değerlendirilmesi yapılmıştır. Elde

edilen performans sonuçları, koçan tabanlı uyarlanabilir süzgeçleme ile kanal denkleştirme probleminin geleneksel doğrusal uyarlanabilir süzgeçleme yöntemlerine göre çok daha iyi çözülebildiği gözlemlenmiş ve gösterilmiştir.

Bu çalışmada özet olarak, kanal denkleştirme probleminin çözümü için yeni bir doğrusal olmayan uyarlanabilir süzgeçleme tekniği önerilmiş ve denenmiştir. Bu teknikte, süzgeçlemeye ilişkin işlemler doğrusal olarak gerçekleştirilmekte, ancak yapılan koçan dönüşümü sayesinde doğrusal olmayan bir nitelik kazanmaktadır. Bu da, hem doğrusal olmayan süzgeçleme algoritmalarının hesapsal yoğunluklarından kurtulmak ve de doğrusal olmayan karşılıklarına göre çok daha gelişmiş olan doğrusal işlem/hesaplama literatürünü kullanabilmek gibi avantajları beraberinde getirmektedir.

Sonuç olarak, elde edilen başarımlar/performans açısından bakıldığında ,umut verici olmuştur. Bu da koçan uyarlanabilir süzgeç algoritmalarının doğrusal karşılıkları yerine kullanılmasının hem mümkün hem de gerekli olduğunu göstermektedir. Elbette, bu yöntemler gerçek zamanlı sistemlere uygulanmadan, nihai bir karara ulaşmak yanıltıcı olabilir. Buradan yola çıkarak da, gelecek çalışma olarak burada kullanılan algoritmaların gerçek zamanlı sistemlerde kullanımının gerçekleştirilmesi hedeflenmektedir. Nitekim, bu çalışmanın devamı bir doktora çalışması olarak düşünülmektedir.

8. KAYNAKÇA

Abel, A.,Schwarz, W., Chaos Communications Principles, Schemes and Systems Analysis, Proceedings of the IEEE, Vol. 90, No.5, 2002.

AronszajnN.,Theory of reproducing kernels . Transactions of the American Mathematical Society, 68 : 337 – 404 , 1950

Başkurt, P., Kaotik Sinyallerden Model Bilgisi Olmadan Gürültü Temizlenmesi İçin Uyarlamalı Süzgeç Algoritmalarının Performans Karşılaştırması, Yüksek Lisans Tezi, Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Eskişehir, 2011

Birlikbaş, S.,Adaptif Dijital Filtre Yapıları , Öğrenme Algoritmaları Ve Uygulamaları ,Yuksek Lisans Tezi, Elazığ ,Fırat Üniversitesi 2000

Burges C. J. C., A tutorial on support vector machines for pattern recognition . Data Mining and Knowledge Discovery, 2 (2): 121 – 167 , 1998

Çakar. Ö., Fonksiyonel Analize Giriş I, ErwinKreyszig'den uyarlama ders notu, Ankara Üniversitesi, 2007

Çetinel, G., Kaotik Haberleşme Sistemlerinde Gözü Kapalı Kanal Denkleştirme, Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya, 2010

Engel, S. Mannor, and R. Meir, The Kernel Recursive Least Squares Algorithm, IEEE Transactions on Signal Processing, volume 52, no. 8, pp. 2275–2285, Aug. 2004

Feldbauer, C., Pernkopf, F. ve Rank, E., Adaptive filters - A tutorial for the course computational Intelligence, Signal Processing and Speech Communication Laboratory, 2010.

Farhang-Boroujeny, B., Adaptive filters Theory and Application, John Wiley and Sons, Chichester, İngiltere, 1998.

Gretton A., Introduction to RKHS, and some simple kernel algorithms, Lecture Notes, February, 2014

Haykin S., Adaptive filter theory, Prentice Hall Information and Science Series, NJ, ABD, 1996.

Liu W., Adaptive Filtering in Reproducing Kernel Hilbert Spaces, PhD Dissertation, University of Florida, 2008

Liu, W., Principe, J. C., Haykin S., Kernel Adaptive Filtering: A Comprehensive Introduction, John Wiley & Sons, 2010

Liu, W., Principe, J. C., Haykin S., Kernel Adaptive Filtering: A Comprehensive Introduction, John Wiley&Sons, 2010

Liu W., Pokharel P., Principe J., "The Kernel Least-Mean-Square Algorithm," IEEE Transactions on Signal Processing, Volume 56, Issue 2, 2008

Mahmut Y., Kablosuz İletişim Sistemlerinde Zaman-Frekans Yaklaşımı ile Kanal Modelleme ve Kestirimi, Doktora Tezi, İstanbul Üniversitesi, Temmuz, 2011

Özer, Ş., Zorlu H., Doğrusal Olmayan PAR Sistemler Kullanılarak Kaotik Zaman Serisi Kestirimi, Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi, Cilt 27, No 2, 323-331, 2012

Özşahin, A. T., OFDM Sistemlerinde Yapay Sinir Ağları Kullanarak Kanal Dengeleme, Yüksek Lisans Tezi, Erciyes Üniversitesi, 2006

Pokharel P., Liu W., Principe J., "Kernel LMS," International Conference on Acoustics, Speech, and Signal Processing, Honolulu, Hawaii, 2007

Pokharel P., Liu W., Principe J., "Kernel Least Mean Square Algorithm with Constrained Growth," Signal Processing, Volume 89, Issue 3, March 2009

Sevim, O., Gürültü azaltmada LMS adaptif süzgeçlerin FPGA kullanarak uygulanması, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2007.

Sklar B., Rayleigh Fading Channels in Mobile Digital Communication Systems, IEEE Communications Magazine, 90- 109, 1997

Stavroulakis, P., Chaos Applications in Telecommunications, Taylor & Francis Group, 2006.

Stüber, G. L., Principles of Mobile Communication. Kluwer Academic Publishers, 2000

Sunan, E., Uyarlanabilir Süzgeçler Ve Uygulama Alanlarının İncelenmesi, Yüksek Lisans Semineri. Samsun Ondokuz Mayıs Üniversitesi, 2006

Widrow B. ve M. E. Hoff. Adaptive switching circuits. IRE WESCON Convention Record, 4:96–104, 1960.

EK

Kod.1

Mackey-Glass zaman serisi için LMS ve koçan LMS algoritmalarının karşılaştırılması

```

-----
clear all; close all; clc

% zaman gecikmesi
TD = 10;
% koçan parametresi
a = 1;
% gürültü standart sapması
np = .05;
% veri boyutu
N_tr = 1000;
N_te = 1000;

% veri yükleme
load MK30

MK30 = MK30+np*randn(size(MK30));
MK30 = MK30 - mean(MK30);

train_set = MK30(1501:3000);
test_set = MK30(3401:4950);

X = zeros(TD,N_tr);
for k=1:N_tr
    X(:,k) = train_set(k:k+TD-1)';
end
T = train_set(TD+1:TD+N_tr);

X_te = zeros(TD,N_te);
for k=1:N_te
    X_te(:,k) = test_set(k:k+TD-1)';
end
T_te = test_set(TD+1:TD+N_te);

mse_te_l = zeros(N_tr,1);

% LMS
% öğrenme hızı
lr_l = .3;
w1 = zeros(1,TD);
e_l = zeros(N_tr,1);
for n=1:N_tr
    y = w1*X(:,n);
    e_l(n) = T(n) - y;
    w1 = w1 + lr_l*e_l(n)*X(:,n)';

    % test
    err_te = T_te'-(w1*X_te);
    mse_te_l(n) = mean(err_te.^2);
end
% LMS sonu
sum(mse_te_l)

figure(1)
subplot(211),plot(w1*X_te),hold on,plot(T_te,'r'),xlabel('zaman'),...
    title('Mackey-Glass zaman serisi'),hold off
subplot(212),plot(mse_te_l,'k'),title('LMS ortalama kare hata'),...
    xlabel('iterasyon')

% Koçan LMS

% öğrenme hızı

```

```

lr_k = .3;

e_k = zeros(N_tr,1);
y = zeros(N_tr,1);
mse_te_k = zeros(N_tr,1);

e_k(1) = T(1);
y(1) = 0;
mse_te_k(1) = mean(T_te.^2);
for n=2:N_tr
    % eğitim
    ii = 1:n-1;
    y(n) = lr_k*e_k(ii)*(exp(-sum((X(:,n)*ones(1,n-1)-X(:,ii)).^2)))';
    e_k(n) = T(n) - y(n);

    % test
    y_te = zeros(N_te,1);
    for jj = 1:N_te
        y_te(jj) = lr_k*e_k(1:n)*(exp(-sum((X_te(:,jj)*ones(1,n)-X(:,1:n)).^2)))';
    end
    err = T_te - y_te;
    mse_te_k(n) = mean(err.^2);

end

sum(mse_te_k)

figure(2)
subplot(211),plot(y_te),hold on,plot(T_te,'r'),xlabel('zaman'),...
    title('Mackey-Glass zaman serisi'),hold off
subplot(212),plot(mse_te_k),title('KLMS ortalama kare hata'),...
    xlabel('iterasyon')
% Koçan LMS sonu

```

Kod.2

Lorenz zaman serisi için LMS ve koçan LMS algoritmalarının karşılaştırılması

```

-----
clear all; close all; clc

% zaman gecikmesi
TD = 4;
% koçan parametresi
a = 1;
% gürültü standart sapması
np = 0.05;
% veri boyutu
N_tr = 1000;
N_te = 1000;

% veri yükleme
load lorenz

lorenz2 = lorenz2 - mean(lorenz2);
lorenz2 = lorenz2/std(lorenz2);

MK30=lorenz2';

MK30 = MK30+np*randn(size(MK30));

train_set = MK30(1501:4500);
test_set = MK30(6071:8050);

X = zeros(TD,N_tr);
for k=1:N_tr
    X(:,k) = train_set(k:k+TD-1)';

```

```

end
T = train_set(TD+1:TD+N_tr);

X_te = zeros(TD,N_te);
for k=1:N_te
    X_te(:,k) = test_set(k:k+TD-1)';
end
T_te = test_set(TD+1:TD+N_te);

mse_te_l = zeros(N_tr,1);

% LMS

lr_l = .11;% öğrenme hızı
w1 = zeros(1,TD);
e_l = zeros(N_tr,1);
for n=1:N_tr
    y = w1*X(:,n);
    e_l(n) = T(n) - y;
    w1 = w1 + lr_l*e_l(n)*X(:,n)';

    % test
    err_te = T_te'-(w1*X_te);
    mse_te_l(n) = mean(err_te.^2);
end
% LMS sonu
sum(mse_te_l)

figure(1)
subplot(211),plot(w1*X_te),hold on,plot(T_te,'r'),xlabel('zaman'),...
    title('Lorenz zaman serisi'),hold off
subplot(212),plot(mse_te_l,'k'),title('LMS ortalama kare hata'),...
    xlabel('iterasyon')

% Koçan LMS

% öğrenme hızı
lr_k = .4;

e_k = zeros(N_tr,1);
y = zeros(N_tr,1);
mse_te_k = zeros(N_tr,1);

e_k(1) = T(1);
y(1) = 0;
mse_te_k(1) = mean(T_te.^2);

for n=2:N_tr
    % eğitim
    ii = 1:n-1;
    y(n) = lr_k*e_k(ii)'*(exp(-sum((X(:,n)*ones(1,n-1)-X(:,ii)).^2)))';
    e_k(n) = T(n) - y(n);

    % test
    y_te = zeros(N_te,1);
    for jj = 1:N_te
        y_te(jj) = lr_k*e_k(1:n)'*(exp(-sum((X_te(:,jj)*ones(1,n)-X(:,1:n)).^2)))';
    end
    err = T_te - y_te;
    mse_te_k(n) = mean(err.^2);
end

figure(2)
subplot(211),plot(y_te),hold on,plot(T_te,'r'),xlabel('zaman'),...
    title('Lorenz zaman serisi'),hold off
subplot(212),plot(mse_te_k),title('KLMS ortalama kare hata'),...
    xlabel('iterasyon')
% Koçan LMS sonu

sum(mse_te_k)

```

Kod.3

Mackey-Glass zaman serisi için RLS ve koçan KRLS algoritmalarının karşılaştırılması

```

-----
close all; clear all; clc

load MK30

np=0.05;

MK30 = MK30';
MK30 = MK30+np*randn(size(MK30));
lorenz2 = MK30 - mean(MK30);

trainSize = 1000;
inDim = 3;

% Koçan parametreleri
typeKernel = 'Gauss';
paramKernel = .01;

L = 1;

ensLearnRls = zeros(trainSize,1);
ensLearnKrls = zeros(trainSize,1);

for k = 1

    inputSignal = lorenz2(k*trainSize:k*trainSize+trainSize+inDim+1);

    trainInput = zeros(inDim,trainSize);
    for kk = 1:trainSize
        trainInput(:,kk) = inputSignal(kk:kk+inDim-1);
    end

    % Eğitim işareti
    trainTarget = zeros(trainSize,1);
    for ii=1:trainSize
        trainTarget(ii) = inputSignal(ii+inDim-1);
    end

    % RLS
    epsilon = 1e-3;
    pInitialRls = (1/epsilon)*eye(inDim);
    forgettingFactorRls = 5;

    [weightVectorRls,learningCurveRls,rec_rls]= ...
        RLS(trainInput,trainTarget,pInitialRls,forgettingFactorRls);

    % RLS sonu

    % Koçan RLS
    regularKrls = 0.001;
    forgetKrls = 1;

    [expansionCoefficientKrls,learningCurveKrls,rec] = ...
        KRLS(trainInput,trainTarget,typeKernel,paramKernel,regularKrls,forgetKrls);

    % Koçan RLS sonu

    ensLearnRls = ensLearnRls + learningCurveRls;
    ensLearnKrls = ensLearnKrls + learningCurveKrls;

end

figure(1)
subplot(211),plot(rec_rls,'r'),hold on,plot(trainInput(3,:)),...

```

```

        xlabel('zaman'),title('Mackey-Glass zaman serisi'),hold off
subplot(212),plot(ensLearnRls(2:end)/L),...
        title('RLS ortalama kare hata'),xlabel('iterasyon')

sum(ensLearnRls(2:end)/L)

figure(2)
subplot(211),plot(rec,'r'),hold on,plot(trainInput(3,:)),...
        xlabel('zaman'),title('Mackey-Glass zaman serisi'),hold off
subplot(212),plot(ensLearnKrls(2:end)/L),...
        title('KRLS ortalama kare hata'),xlabel('iterasyon')

sum(ensLearnKrls(2:end)/L)

```

Kod.4

Lorenz zaman serisi için RLS ve koçan KRLS algoritmalarının karşılaştırılması

```

-----
close all; clear all; clc

load lorenz.mat

lorenz2 = lorenz2 - mean(lorenz2);
lorenz2 = lorenz2/std(lorenz2);

np=0.05;

lorenz2 = lorenz2 + np*randn(size(lorenz2));

trainSize = 1000;
inputDim = 5;

% Koçan parametreleri
typeKernel = 'Gauss';
paramKernel = .1;

L = 1;

ensLearnRls = zeros(trainSize,1);
ensLearnKrls = zeros(trainSize,1);

for k = 8

    inputSignal = lorenz2(k*trainSize:k*trainSize+trainSize+inputDim+1);

    trainInput = zeros(inputDim,trainSize);
    for kk = 1:trainSize
        trainInput(:,kk) = inputSignal(kk:kk+inputDim-1);
    end

    % Eğitim işareti
    trainTarget = zeros(trainSize,1);
    for ii=1:trainSize
        trainTarget(ii) = inputSignal(ii+inputDim-1);
    end

    % RLS

    epsilon = 1e-3;
    pInitialRls = (1/epsilon)*eye(inputDim);
    forgettingFactorRls = 2;

```

```

[weightVectorRls, learningCurveRls, rec_rls]= ...
    RLS(trainInput, trainTarget, pInitialRls, forgettingFactorRls);

% RLS sonu

% Koçan RLS

regularKrls = 0.001;
forgetKrls = 1;

[expansionCoefficientKrls, learningCurveKrls, rec] = ...
    KRLS(trainInput, trainTarget, typeKernel, paramKernel, regularKrls, forgetKrls);

% Koçan RLS sonu

ensLearnRls = ensLearnRls + learningCurveRls;
ensLearnKrls = ensLearnKrls + learningCurveKrls;

end

figure(1)
subplot(211), plot(rec_rls, 'r'), hold on, plot(trainInput(3, :)), ...
    xlabel('zaman'), title('Lorenz zaman serisi'), hold off
subplot(212), plot(ensLearnRls(1:end)/L), ...
    title('RLS ortalama kare hata'), xlabel('iterasyon')

sum(ensLearnRls(1:end)/L)

figure(2)
subplot(211), plot(rec, 'r'), hold on, plot(trainInput(3, :)), ...
    xlabel('zaman'), title('Lorenz zaman serisi'), hold off
subplot(212), plot(ensLearnKrls(1:end)/L), ...
    title('KRLS ortalama kare hata'), xlabel('iterasyon')

sum(ensLearnKrls(1:end)/L)

```


ÖZET

Doğrusal olmayan uyarlanabilir süzgeçleme alanının aksine doğrusal uyarlanabilir süzgeç kuramları bir doyum noktasına ulaşmıştır. Koçan yöntemler doğrusal olmayan ve sağlam matematiksel temelleri olan yöntemlerdir. Bu yüzden de son zamanlarda yapılan birçok araştırma için oldukça cezbedici olmuştur.

Bu çalışmada, haberleşme sistemlerinde kanal denkleştirme problemi özelinde, doğuran koçanlı Hilbert uzayında uyarlanabilir süzgeçleme algoritmaları incelenmiştir. Bu bağlamda bir Rayleigh kanal modeli kullanılmış ve uygulama örnekleri bu yönde yürütülmüştür.

Doğuran koçanlı Hilbert uzayı, koçan hilesi kullanmak suretiyle iç çarpımlar olarak ifade edilen doğrusal uyarlanabilir süzgeç algoritmalarının doğrusal olmayan karşılıklarının elde edilebilmesi için bir yöntem ortaya koymaktadır. Dolayısı ile, en iyi bilinen uyarlanabilir süzgeçleme yöntemleri olan en küçük ortalama kareler ve yinelenen en küçük kareler algoritmalarının koçan uygulamaları sunulmuş ve kanal denkleştirme problemi için performansları değerlendirilmiştir.

Anahtar Kelimeler: Kanal Denkleştirme, Uyarlanabilir Süzgeçleme, Doğuran Koçanlı Hilbert Uzayı

SUMMARY

The theory of linear adaptive filtering has reached a saturation, unlike the area of nonlinear adaptive filtering. Kernel methods which are nonlinear and have solid mathematical foundations. This makes them very appealing in recent research studies.

In this study, adaptive filtering, algorithms in reproducing kernel Hilbert spaces (RKHS) are investigated in the case of channel equalization problem in communication systems. In this concept, a Rayleigh channel model is incorporated, and applications examples are conducted in this direction.

The reproducing kernel Hilbert space provides a method for obtaining nonlinear counterparts of linear adaptive filtering algorithms which are expressed in terms of inner products by incorporating kernel trick. Thus, kernel extensions for well-known adaptive filtering methods, the least-mean-square, and the recursive-least-squares are presented and their performances in channel equalization problem are studied.

Key Words: Channel Equalization, Adaptive Filtering, Reproducing Kernel Hilbert Space