

**KAOTİK SİNYALLERDEN MODEL BİLGİSİ OLMADAN
GÜRÜLTÜ TEMİZLENMESİ İÇİN UYARLAMALI
SÜZGEÇ ALGORİTMALARININ PERFORMANS
KARŞILAŞTIRMASI**

Polat BAŞKURT

Yüksek Lisans Tezi

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Kasım-2011

JÜRİ VE ENSTİTÜ ONAYI

Polat BAŞKURT'un "Kaotik Sinyallerden Model Bilgisi Olmadan Gürültü Temizlenmesi İçin Uyarlamalı Süzgeç Algoritmalarının Performans Karşılaştırması" başlıklı Elektrik-Elektronik Mühendisliği Anabilim Dalındaki, Yüksek Lisans Tezi 14.10.2011 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Adı-Soyadı	İmza
Üye (Tez Danışmanı) : Doç. Dr. M. TANKUT ÖZGEN
Üye : Doç. Dr. AYDIN AYBAR
Üye : Doç. Dr. YUSUF OYSAL

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun
..... tarih ve sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

KAOTİK SİNYALLERDEN MODEL BİLGİSİ OLMADAN GÜRÜLTÜ TEMİZLENMESİ İÇİN UYARLAMALI SÜZGEÇ ALGORİTMALARININ PERFORMANS KARŞILAŞTIRMASI

Polat BAŞKURT

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. M. Tankut ÖZGEN

2011, 156 sayfa

Tez kapsamında üzerine Gauss tipi gürültü eklenmiş kaotik sinyaller, model bilgisi olmadan, sadece gürültü sinyalinin değışinti değeri veya onun bir kestirimi bilinerek, farklı uyarlamalı süzgeç algoritmaları ile süzölmüş ve bu algoritmaların performans karşılaştırması yapılmıştır. Süzgeçleme işlemi Evrensel Sonlu Dörtü Tepkili En Küçük Ortalama Karese Hata süzgeci (Evrensel FIR MMSE süzgeci), Wiener süzgeci, LMS-benzeri süzgeç, Polinom Uydurma yöntemi ile gerçekleştirilen süzgeç ve Dalgacık Katsayısı Sıkıştırma yöntemi ile gerçekleştirilen süzgeç vasıtasıyla yapılmıştır. Daha sonra MATLAB program dili kullanılarak tasarlanan süzgeçlerin performans karşılaştırması yapılmıştır. Son olarak gürülsüz (asıl) kaotik sinyalin korelasyon boyutu ile süzgeçlerin çıktılarında elde edilen sinyallerin korelasyon boyutları MATLAB programı ile hesaplanarak, karşılaştırılmıştır.

Anahtar Kelimeler: Kaotik Sinyal, Evrensel FIR MMSE Süzgeç, LMS Benzeri Süzgeç, Polinom Uydurma Yöntemi, Dalgacık Katsayısı Sıkıştırma Yöntemi.

ABSTRACT

Master of Science Thesis

**Comparing Performances of Adaptive
Filter Methods for Model
Independent Noise Cleaning of
Chaotic Signals**

Polat BAŞKURT

Anadolu University

Graduate School of Sciences

Electrical and Electronics Engineering Program

Supervisor: Assoc. Prof. M. Tankut ÖZGEN

2011, 156 pages

In this thesis, Gaussian noise added chaotic signals are filtered using different adaptive filtering algorithms, that utilize the known noise variance only, without any knowledge of system models of chaotic signals, and performances of these filtering algorithms are compared. The filtering process is done by using Universal Finite Impulse Response Minimum Mean Square Error filter (Universal FIR MMSE filter), the Wiener filter and LMS-like filter, filter designed using a Polynomial Fitting method and the filter designed using Wavelet Coefficient Shrinkage method. Using Matlab programming language, performances of the implemented filters are compared. Finally, the correlation dimensions of noise-free chaotic signals and those of filter output signals are calculated using Matlab and compared.

Key Words : Chaotic Signal, Universal FIR MMSE Filter, LMS-like Filter, Polynomial Fitting Method, Wavelet Coefficient Scaling method.

TEŞEKKÜR

Yüksek lisans sürecinde ve tez çalışması esnasında zaman ve mekan gözetmeksizin bana yardımcı olan tez danışmanım Doç. Dr. M. Tankut ÖZGEN'e teşekkürü borç bilirim.

İş ile beraber yüksek lisansa devam etmemi sağlayan şeflerim Yalçın TÜZMEN, Ferdi TUNALI, Serap YENİGÜN ve iş arkadaşlarım İbrahim GİRGİN, Abbas Erkan SARIÇOBAN, Hakan ÇAVUŞER ve Arif MENCET'e teşekkür ederim.

Bu süreçte bana her zaman destek veren aileme ve tezime yardımlarda bulunan arkadaşım Ziya ERCAN ve kuzenim Gizem KATIRCI'ya teşekkür ederim.

Polat BAŞKURT

01.09.2011

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ	xii
ÇİZELGELER DİZİNİ	xv
SİMGELER VE KISALTMALAR DİZİNİ.....	xvi

1.GİRİŞ	1
2. TEORİK ARKA PLAN	4
2.1 Kaotik Sinyal Çeşitleri	4
2.2 Kendinden Uyarlamalı Süzgeç	5
2.2.1 LMS algoritması	5
2.2.2 RLS algoritması	8
2.3 Problemin Tanımı	13
3. DOĞRUSAL UYARLAMALI SÜZGEÇLER	15
3.1 Evrensel FIR MMSE Süzgeci	15
3.2 Wiener Süzgeci	17
3.3 LMS-Benzeri Süzgeç	22

4. DOĞRUSAL OLMAYAN UYARLAMALI SÜZGEÇLEME	
YÖNTEMLERİ	25
4.1 Polinom Uydurma Yöntemi	25
4.2 Dalgacık Katsayısı Sıkıştırma Yöntemi	27
4.2.1 Sürekli dalgacık dönüşümü	29
4.2.2 Ayırık dalgacık dönüşümü	29
4.2.3 Dalgacık katsayısı sıkıştırma yöntemi ile gürültü giderme	32
5.SİMULASYON (BENZETİM) SONUÇLARI	34
5.1 Süzgeç Algoritmalarında Parametre Seçimi.....	35
5.2 Aynı Süzgecin Performansının Farklı Değişinti Değeri Kriteri Altında Karşılaştırılması.....	35
5.2.1 Evrensel FIR MMSE süzgeç simülasyonları.....	36
5.2.2 Wiener süzgeç simülasyonları	37
5.2.3 LMS-benzeri süzgeç simülasyonları.....	39
5.2.4 Polinom uydurma yöntemi simülasyonları.....	40
5.2.5 Dalgacık katsayısı sıkıştırma yöntemi simülasyonları.....	42
5.3 Farklı Süzgeçlerin Performanslarının Aynı Değişinti Değeri Kriteri Altında Karşılaştırılması.....	46
5.3.1 Değişinti değeri=0.1 (var01) için farklı süzgeçlerin performanslarının karşılaştırılması.....	47
5.3.2 Değişinti değeri=1 (var1) için farklı süzgeçlerin performanslarının karşılaştırılması.....	49
5.3.3 Değişinti değeri=2.5 (var2.5) için farklı süzgeçlerin performanslarının karşılaştırılması.....	51

5.3.4 Değişinti değeri=5 (var5) için farklı süzgeçlerin performanslarının karşılaştırılması.....	52
6. KORELASYON BOYUTU	55
7. SONUÇLAR VE TARTIŞMA	62
KAYNAKLAR.....	65
Ek-1 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan Evrensel FIR MMSE Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken)	67
Ek-2 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan Wiener Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken)	69
Ek-3 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan LMS Benzeri Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken)	71
Ek-4 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken)	72
Ek-5 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda=\sigma$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken)	74
Ek-6 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda_j^{[1]}=\sigma_j$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken)	75

Ek-7 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda_j^{[2]} = \sigma_j$ / β) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken).....	77
Ek-8 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan Süzgeç Programlarını Toplu Olarak Çalıştıran Program (Süzgeç Programlarının Değişinti Değerlerini Değiştiren Program).....	79
Ek-9 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan Evrensel FIR MMSE Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması).....	81
Ek-10 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan Wiener Süzgeci Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması).....	83
Ek-11 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan LMS Benzeri Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması).....	85
Ek-12 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması)..	87
Ek-13 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda = \sigma$) ile Tasarlanan Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması).....	89
Ek-14 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan Süzgeç Programlarını Toplu Olarak Çalıştıran Program(Sabit Değişinti Değerleri Altında Süzgeç Programlarını Karşılaştıran Program).....	91

Ek-15 Gürültülü Henon Sinyalini Süzmek için Tasarlanan Evrensel FIR MMSE Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken).....	93
Ek-16 Gürültülü Henon Sinyalini Süzmek için Tasarlanan Wiener Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken).....	95
Ek-17 Gürültülü Henon Sinyalini Süzmek için Tasarlanan LMS Benzeri Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken).....	97
Ek-18 Gürültülü Henon Sinyalini Süzmek için Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken).....	99
Ek-19 Gürültülü Henon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda = \sigma_j$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken)	101
Ek-20 Gürültülü Henon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda_j^{[1]} = \sigma_j$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken).....	103
Ek-21 Gürültülü Henon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda_j^{[2]} = \sigma_j / \beta$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken).....	105
Ek-22 Gürültülü Henon Sinyalini Süzmek için Tasarlanan Süzgeç Programlarını Toplu Olarak Çalıştıran Program (Süzgeç Programlarının Değışinti Değeri Değıştiren Program).....	107
Ek-23 Gürültülü Henon Sinyalini Süzmek için Tasarlanan Evrensel FIR MMSE Süzgeç Programı (Sabit Değışinti Değeri Altında Farklı Süzgeç Programlarının Karşılaştırılması).....	109

Ek-24 Gürültülü Henon Sinyalini Süzmek için Tasarlanan Wiener Süzgeci Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması).....	111
Ek-25 Gürültülü Henon Sinyalini Süzmek için Tasarlanan LMS Benzeri Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması).....	113
Ek-26 Gürültülü Henon Sinyalini Süzmek için Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması).....	115
Ek-27 Gürültülü Henon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda=\sigma$) ile Tasarlanan Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması).....	117
Ek-28 Gürültülü Henon Sinyalini Süzmek için Tasarlanan Süzgeç Programlarını Toplu Olarak Çalıştıran Program(Sabit Değişinti Değerleri Altında Süzgeç Programlarını Karşılaştıran Program).....	119
Ek-29 Gürültüsüz Lojistik Regresyon Sinyalin Korelasyon Boyutunun Hesaplanması	121
Ek-30 Gürültülü Lojistik Regresyon Sinyali için Tasarlanan Evrensel Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması	122
Ek-31 Gürültülü Lojistik Regresyon Sinyali için Tasarlanan Wiener Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması	124
Ek-32 Gürültülü Lojistik Regresyon Sinyali için Tasarlanan LMS Benzeri Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması	126
Ek-33 Gürültülü Lojistik Regresyon Sinyali için Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması	128

Ek-34 Gürültülü Lojistik Regresyon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi ($\lambda=\sigma$) ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması.....	130
Ek-35 Gürültülü Lojistik Regresyon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi ($\lambda_j^{[1]}=\sigma_j$) ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması.....	132
Ek-36 Gürültülü Lojistik Regresyon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi ($\lambda_j^{[2]}=\sigma_j / \beta$) ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması.....	134
Ek-37 Gürültülü Lojistik Regresyon Sinyali için Tasarlanan Süzgeç Çıktılarının Korelasyon Boyutlarını Karşılaştıran Program.....	136
Ek-38 Gürültüsüz Henon Sinyalin Korelasyon Boyutunun Hesaplanması.....	139
Ek-39 Gürültülü Henon Sinyali için Tasarlanan Evrensel Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması	140
Ek-40 Gürültülü Henon Sinyali için Tasarlanan Wiener Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması.....	142
Ek-41 Gürültülü Henon Sinyali için Tasarlanan LMS Benzeri Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması	144
Ek-42 Gürültülü Henon Sinyali için Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması	146
Ek-43 Gürültülü Henon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi ($\lambda=\sigma$) ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması.....	148

Ek-44 Gürültülü Henon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi ($\lambda_j^{[1]} = \sigma_j$) ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması.....	150
Ek-45 Gürültülü Henon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi ($\lambda_j^{[2]} = \sigma_j / \beta$) ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması.....	152
Ek-46 Gürültülü Henon Sinyali için Tasarlanan Süzgeç Çıktılarının Korelasyon Boyutlarını Karşılaştıran Program.....	154

ŞEKİLLER DİZİNİ

Sayfa

1.1. Periyodik (sinüs) (a), kaotik (Lorenz) (b) ve gürültü (rastgele) (c) sinyalleri ve sinyallerin altlarında her sinyalin güç spektrumları	2
2.1. N-taplı enine uyarlamalı süzgeç.....	6
3.1. Enine süzgeç.....	17
4.1. Dalgacık türlerine örnekler.....	28
4.2. Mallat Algoritması.....	31
4.3. Eşik değerinin uygulanması.....	33
5.1. Gürültülü Lojistik regresyon sinyal için tasarlanan Evrensel FIR MMSE süzgecin farklı değişinti değerleri altındaki Ortalama Karesel Hata değerleri gösterimi.....	36
5.2. Gürültülü Henon sinyali için tasarlanan Evrensel FIR MMSE süzgecin farklı değişinti değerleri altındaki Ortalama Karesel Hata değerleri gösterimi	37
5.3. Gürültülü Lojistik regresyon sinyali için tasarlanan Wiener süzgecinin farklı değişinti değerleri altındaki Ortalama Karesel Hata değerleri gösterimi	38
5.4. Gürültülü Henon sinyali için tasarlanan Wiener süzgecinin farklı değişinti değerleri altındaki Ortalama Karesel Hata değerleri gösterimi	38
5.5. Gürültülü Lojistik regresyon sinyali için tasarlanan LMS-benzeri süzgecin farklı değişinti değerleri altındaki Ortalama Karesel Hata değerleri gösterimi.....	39
5.6. Gürültülü Henon sinyali için tasarlanan LMS-benzeri süzgecin farklı değişinti değerleri altındaki Ortalama Karesel Hata değerleri gösterimi	40

5.7. Gürültülü Lojistik regresyon sinyali için Polinom Uydurma yöntemi ile tasarlanan süzgecin farklı değışinti değeri altındaki Ortalama Karesel Hata değeri gösterimi.....	41
5.8. Gürültülü Henon sinyali için Polinom Uydurma yöntemi ile tasarlanan süzgecin farklı değışinti değeri altındaki Ortalama Karesel Hata değeri gösterimi.....	41
5.9. Gürültülü Lojistik regresyon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda = \sigma$) farklı değışinti değeri altındaki Ortalama Karesel Hata değeri gösterimi.....	42
5.10. Gürültülü Henon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda = \sigma$) farklı değışinti değeri altındaki Ortalama Karesel Hata değeri gösterimi.....	43
5.11. Gürültülü Lojistik regresyon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda_j^{[1]} = \sigma_j$) farklı değışinti değeri altındaki Ortalama Karesel Hata değeri gösterim.....	44
5.12. Gürültülü Henon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda_j^{[1]} = \sigma_j$) farklı değışinti değeri altındaki Ortalama Karesel Hata değeri gösterimi.....	44
5.13. Gürültülü Lojistik regresyon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda_j^{[2]} = \sigma_j / \beta$) farklı değışinti değeri altındaki Ortalama Karesel Hata değeri gösterimi.....	45
5.14. Gürültülü Henon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda_j^{[2]} = \sigma_j / \beta$) farklı değışinti değeri altındaki Ortalama Karesel Hata değeri gösterimi.....	46

5.15. Gürültülü Lojistik regresyon sinyali için deęişinti deęeri=0.1(var01) altında süzgeç programlarının Ortalama Karesel Hata deęerleri gösterimi.....	48
5.16. Gürültülü Henon sinyali için deęişinti deęeri=0.1 (var01) altında süzgeç programlarının Ortalama Karesel Hata deęerleri gösterimi.....	48
5.17. Gürültülü Lojistik regresyon sinyali için deęişinti deęeri=1 (var1) altında süzgeç programlarının Ortalama Karesel Hata deęerleri gösterimi.....	49
5.18. Gürültülü Henon sinyali için deęişinti deęeri=1 (var1) altında süzgeç programlarının Ortalama Karesel Hata deęerleri gösterimi.....	50
5.19. Gürültülü Lojistik regresyon sinyali için deęişinti deęeri=2.5 (var2.5) altında süzgeç programlarının performans gösterimi.....	51
5.20. Gürültülü Henon sinyali için deęişinti deęeri=2.5 (var2.5) altında süzgeç programlarının Ortalama Karesel Hata deęerleri gösterimi.....	52
5.21. Gürültülü Lojistik regresyon sinyali için deęişinti deęeri=5 (var5) altında süzgeç programlarının Ortalama Karesel Hata deęerleri gösterimi.....	52
5.22. Gürültülü Henon sinyali için deęişinti deęeri=5 (var5) altında süzgeç programlarının Ortalama Karesel Hata deęerleri gösterimi.....	53
6.1. Öklidyen ve fraktal boyutlar.....	55
6.2. Koch tanesi.....	56

ÇİZELGELER DİZİNİ

Sayfa

- 6.1. Lojistik regresyon sinyali için gürültüsüz sinyal ile süzgeç çıktılarında elde edilen sinyallerin korelasyon boyutlarının karşılaştırılması..... 59
- 6.2. Henon sinyali için gürültüsüz sinyal ile süzgeç çıktılarında elde edilen sinyallerin korelasyon boyutlarının karşılaştırılması..... 60

SİMGELER VE KISALTMALAR DİZİNİ

η : Öğrenme Hızı Parametresi

J_t : Süzgeçten Kaynaklı Anlık Kestirilen Kayıp

J_t : Anlık Karesel Kayıp

L_t : Süzgeç t Anına Kadar Biriken Kayıp

ψ : Dalgacık Analiz Fonksiyonu

ψ : Sürekli Dalgacık Dönüşümü Fonksiyonu

ρ : Heaviside Adım Fonksiyonu

C: Kümenin İki Noktasının Aynı Hücre İçinde Olma Olasılığı

MSE: Mean Square Error - Ortalama Karesel Hata

RLS: Recursive Least Squares - Tekrarlayan En Küçük Kareler

LMS : Least Mean Square - En küçük Ortalama Kareler

MMSE: Minimum Mean Square Error - Minimum Ortalama Karesel Hata

KZFD: Kısa Zamanlı Fourier Dönüşümü

AGS:Alçak Geçiren Süzgeç

YGS:Yüksek Geçiren Süzgeç

SDD: Sürekli Dalgacık Dönüşümü

ADD: Ayrık Dalgacık Dönüşümü

GPA: Grassberger ve Procaccia Algoritması

1.GİRİŞ

Kaos (chaos), mutlak evrensel düzen anlamına gelen ‘cosmos’ sözcüğünün tam tersidir ve eski Yunanca’da mutlak anarşi, kargaşa ve düzensizlik anlamına gelmektedir [1]. Kaos teorisi doğrusal olmayan deterministik sistemlerdeki esrarengiz davranışı araştırır. Dinamik sistemlerde görülen bu alışılmamış durum ilk kez Poincare tarafından 1903 yılında keşfedilmiştir. Poincare başlangıç koşulundaki küçük değişimlerin birbirinden çok farklı çıktılar ortaya çıkarabildiğini keşfetmiştir [2].

Poincare’den yıllar sonra, 1971 yılında Lorenz adlı bir meteoroloji uzmanı hava durumunu bilgisayarda modellemeye çalışmıştır. Lorenz, hava durumunu tahmin etmek için hava olaylarını rakamlara indirgemiş, daha sonra da bilgisayar kullanarak elde ettiği sonuçları incelemiştir. Bir gün daha hızlı sonuca ulaşmak için elde ettiği değerlerden birini, 0,506127 yerine 0,506 olarak almıştır. Bu değişikliğin hava akımları içerisindeki etkisi kelebek kadardır fakat bu küçük değişim sonuçları baştan aşağı değiştirmiştir. Lorenz yayınladığı makalesinde kaotik dinamik sistemi tanımlarken kelebek etkisinden bahsetmiştir.

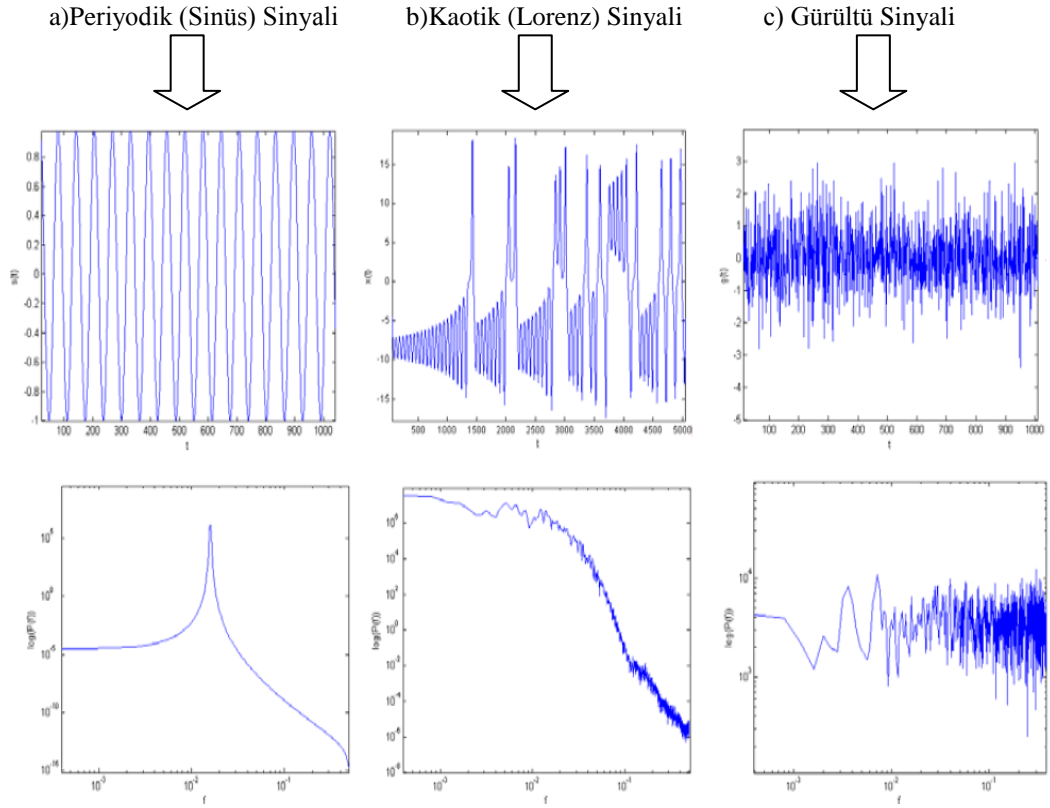
Kaotik sistemler başlangıç değerlerine karşı aşırı duyarlıdırlar. Başlangıç değeri ve denklemleri bilinen deterministik bir sistemin sonraki durumları öngörülebilirken kaotik sistemler için başlangıç değerleri sonsuz duyarlılıkta bilinmiyorsa sistemin ilerleyişini tam olarak kestirmek imkansızdır. Kaotik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

Günlük hayatta sistemlerden yapılan ölçümlere ölçüm aletleri, sinyal kaynağının ideal olmaması vb. gibi birçok sebep etki eder. Bu etkiler ölçümlerde olması istenilmeyen gürültülere sebep olurlar. Elde edilen ölçümlerden daha iyi analiz yapabilmek için sistemin olabildiğince gürültüden ayrıştırılması gerekir.

Kaotik sinyaller geniş güç spektrumuna sahiptirler. Kaotik sinyallerin üzerine çeşitli sebeplerle etki eden gürültüler de geniş güç spektrumuna sahip

olduklarından dolayı üzerine gürültü etki eden kaotik sinyali doğrusal analiz yöntemleri ile gürültüden ayırmak zordur.

Sistemlerden elde edilen gürültülü sinyal doğrusal bir kaynağın çıktısı ise frekans spektrumundaki spektral tepelere bakarak asıl sinyal gürültüden ayrılabilir. Asıl sinyal ile gürültü sinyali farklı (frekans) bantlarında ise gürültüsüz sinyal Fourier analiz ile elde edilebilir. Fakat sinyal kaynağı kaotik ise elde edilmek istenen sinyal de gürültü sinyali de geniş bantlı olacağından dolayı, Fourier analizi bu ayrımı yapmak için yeterli değildir. Şekil (1.1)'de sinüs sinyali, kaotik sinyal ve gürültü sinyali ile bu sinyallerin güç spektrumları görülmektedir.



Şekil 1.1. Periyodik (sinüs) (a), kaotik (Lorenz) (b) ve gürültü (rastgele) (c) sinyalleri ve sinyallerin altlarında her sinyalin güç spektrumları [3].

Üstteki şekile bakıldığında periyodik sinüs sinyalinin güç spektrumunda tek bir frekans bileşenine sahip olduğu, kaotik bir sinyal ile gürültü sinyalinin ise geniş güç spektrumuna sahip olduğu görülmektedir.

Skaler sinyali gürültüden ayırma işlemine “gürültü azaltma” denir. Kaotik sinyalin elde edildiği sistemin denklemleri biliniyorsa, gürültüyü azaltmak için sistem denklemleri kullanılabilir. Sistem denklemleri yerine sistemin önceden ölçülmüş gürültüsüz sinyali biliniyorsa bu sinyal referans sinyali olarak kullanılarak gürültü azaltılabilir. Hem sistem denklemleri hem de referans sinyali olmadan sadece gürültülü sinyal biliniyorsa sinyal modelleri çıkarılarak gürültü bastırılabilir [3]. Gürültüyü bastırmak için kullanılacak bir diğer yöntem ise, bu tezde de anlatıldığı gibi, referans sinyali ve sinyalin elde edildiği sistemin denklemleri hakkında bilgi sahibi olmadan, sadece gürültü sinyali ile ilgili belirli istatistiksel değerler (değişinti (varyans), standart sapma gibi) bilinerek gürültünün azaltılmasıdır. Tez kapsamında, bu sonuncu durum için ortaya atılan belli başlı gürültü bastırma yöntemlerinin performansları incelenmekte ve karşılaştırılmaktadır.

Tez kapsamında vektörler ve matrisler ile skalerlerin karışmaması için vektörler ve matrisler koyu harf ile gösterilmiştir.

2. TEORİK ARKA PLAN

Tez kapsamında, üzerine gürültü eklenmiş kaotik sinyaller işlenmiş ve bunları süzmek için uyarlamalı süzgeç tabanlı algoritmalar kullanılmıştır. Bundan dolayı bu bölümde kaotik sinyaller, kaotik sinyallerin özellikleri, uyarlamalı süzgeçler ve uyarlamalı süzgeç algoritmaları hakkında ön bilgi verilecektir.

2.1 Kaotik Sinyal Çeşitleri

Bu tezde kaotik sinyal olarak ele alınan sinyaller 1-boyutlu Lojistik regresyon sinyali ve 2-boyutlu Henon sinyali'dir. Bu sinyaller daha ayrıntılı incelenirse: Lojistik regresyon sinyalinin genel formu şu şekilde verilir :

(2.1)

Bu form genel olarak nüfus dinamiği çalışmalarında yaygın olarak kullanılmaktadır. Parametre (a) değerine bağlı olarak sistem son derece zengin bir davranış çeşitliği ortaya koyar [4]. Denklem (2.1)'de $a \geq 3.57$ ve $a \neq 3.83$ için sistem kaotik özellik göstermektedir.

Henon sinyalinin genel formu şu şekildedir [5]:

=

(2.2)

Yukarıda denklemin verilen sistem $a=1.4$ ve $b=0.3$ değerleri için kaotik özellik göstermektedir.

2.2 Kendinden Uyarlamalı Süzgeç

Gürültülü sinyalden süzgeçleme işlemi sonunda elde edilecek asıl sinyal hakkında istatistiksel bilginin bulunmadığı durumlarda kendinden uyarlamalı süzgeç algoritmaları kullanılmaktadır. Kendinden uyarlamalı süzgeçler, ilgili sinyalin karakteristik bilgisinin ortamda bulunmadığı durumlarda yinelemeli (recursive) algoritmasıyla, kendi kendini bulunduğu ortama göre ayarlayarak istenilen çıktının oluşturulmasını sağlamaktadır [6].

2.2.1 LMS algoritması

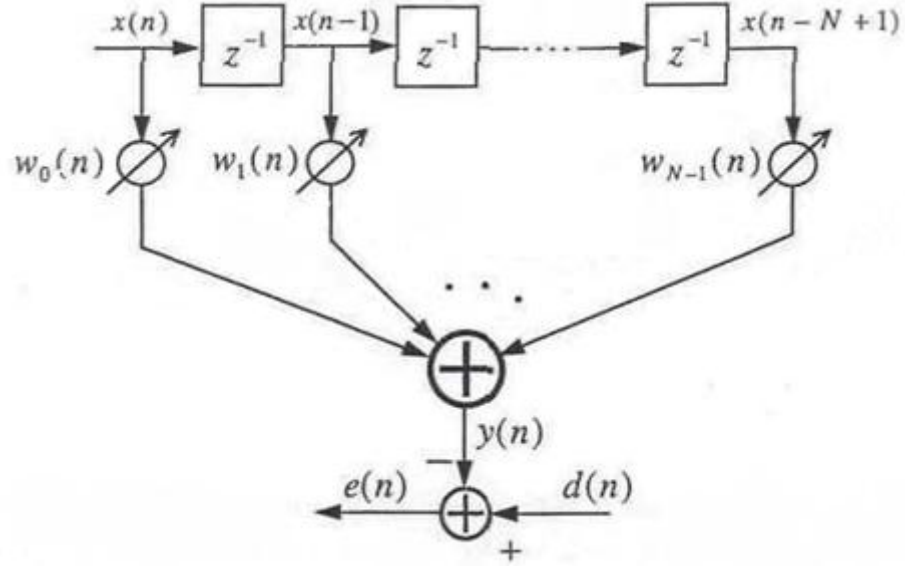
LMS (Least Mean Square - En Küçük Ortalama Kareler) algoritması, maliyet fonksiyonunun gradyan vektörünün anlık kestirimini kullanan en dik azalış algoritmasının bir yaklaşımıdır. Gradyanın kestirimi, tap giriş ve hata sinyalinin örnek değerlerini temel alır. Algoritma süzgeçteki her katsayı üzerinden tekrarlar ve algoritmayı yaklaşık gradyan doğrultusunda ilerletir [7].

N-taplı enine uyarlamalı süzgecin yapısı Şekil 2.1’de gösterilmektedir. Bu şekilde, süzgeç girişi $x(n)$, referans sinyali $d(n)$ ile gösterilirse süzgeç çıktısı

(2.3)

olarak gösterilir. Ağırlık değerleri, $w_0(n)$, $w_1(n)$, ... , $w_{n-1}(n)$, hatayı en aza indirecek şekilde seçilmelidir. Hata sinyali, referans sinyali ile süzgeç çıktısının farkı şeklinde elde edilir.

(2.4)



Şekil 2.1. N-taplı enine uyarlamalı süzgeç [8]

Süzgeçteki ağırlık değerleri, “n” zaman indeksinin fonksiyonlarıdır. Bu, uyarlamalı süzgeçte ağırlık değerlerinin zamanla değişken olduğuna işaret eder. Yani herhangi bir sinyal karakteristiği değişiminde süzgecin ağırlık değerleri değiştirilerek süzgeç çıktısının referans sinyalini takip etmesi sağlanabilir. LMS algoritması ağırlık değerlerini uyarlayarak Ortalama Kare Hata’yı minimize eder. Bu algoritma Wiener süzgecine yaklaşmak için pratik bir şemadır. LMS algoritmasında Wiener-Hopf denklemini çözmeye ihtiyaç yoktur. Geleneksel LMS algoritması, en dik düşüş algoritmasının stokastik yaklaşımıdır. Basit olarak

maliyet fonksiyonunun anlık bayağı kestirimi $\hat{J}(n)$ ’dir

[8]. En dik düşüş algoritmasında $\nabla J(n)$ şeklinde yerine koyarsak

$$(2.5)$$

eşitliği elde edilir.

Denklemden $\mathbf{w}(n)$ vektörüdür, algoritmanın basamak genişliği parametresidir ve ∇ gradyan operatörü sütun vektörü verecek şekilde tanımlanmıştır.

$$\mathbf{w}(n) = \begin{bmatrix} w_0(n) \\ w_1(n) \\ \vdots \\ w_{N-1}(n) \end{bmatrix} \quad (2.6)$$

gradyan vektörünün n 'nci elemanı

$$\frac{\partial J}{\partial w_n} \quad (2.7)$$

olarak elde edilir. Yukarıdaki denklemin en sağ tarafındaki terimin yerine (2.4) yazılırsa ve w_n teriminin w_n teriminden bağımsız olduğu göz önüne alınır

$$\frac{\partial J}{\partial w_n} \quad (2.8)$$

denklemini elde edilir. w_n yerine (2.3) konulursa

$$\frac{\partial J}{\partial w_n} \quad (2.9)$$

ve (2.6) ve (2.9) 'daki denklemler kullanılırsa

$$\frac{\partial J}{\partial w_n} \quad (2.10)$$

denklemine ulaşılır. Yukarıdaki denkleminde

olarak verilmiştir. Denklem (2.10),

(2.5) 'de yerine yazılırsa

$$\frac{\partial J}{\partial w_n} \quad (2.11)$$

denklemini elde edilir. Denklem (2.11), LMS algoritması tekrarlaması olarak tanımlanır. Algoritma her gelen yeni girdi $x(n)$ ve karşılık gelen referans değeri $d(n)$ için süzgeç katsayılarını hesaplar. Denklem (2.3), (2.4) ve (2.11) herbir iterasyonda tanımlanması gereken üç basamağı belirtir.

1) Süzgeç çıktısı

$$y(n) \quad (2.12)$$

2) Kestirim hatası

$$e(n) \quad (2.13)$$

3) Ağırlık uyarlaması

(2.14)

LMS algoritmasını popüler yapan en önemli özellik basitliktir. N örnek uzunluğundaki sinyal vektörü için LMS algoritması, süzgeç çıktısını hesaplarırken N tane çarpma işlemi, skaler-vektör çarpımı için N+1 tane çarpma işlemi olmak üzere toplamda 2N+1 tane çarpma işlemi ve 2N+1 tane toplama işlemi yapmaktadır. LMS algoritmasının bir diğer önemli özelliği ise farklı sinyal koşullarına karşı kararlı ve gürbüz performansa sahip olmasıdır. Bu algoritmanın en büyük problemi yavaş olmasıdır.

2.2.2 RLS algoritması

LMS algoritmasında asıl amaç kestirim hatasının ortalama karesini minimize etmektir. Tekrarlayan En Küçük Kareler algoritmasında ise $n > 0$ için herhangi bir zamanda, ağırlık değerleri hesaplanır ve maliyet fonksiyonunun elde edilen bu ağırlık değerleri yardımıyla minimum olması sağlanır. Matris Ters Alma Önermesi (Matrix Inversion Lemma) olarak bilinen matris eşitliğinden yararlanarak geliştirilmiş ve tekrarlayan en küçük kareler (RLS, Recursive Least Squares) algoritması türetilmiştir. RLS algoritması giriş datası içerisindeki bilgiyi kullanır. Bundan dolayı yakınsama hızı LMS algoritmasına göre daha hızlıdır. Karmaşık hesaplamalarda bu performans artırımını daha belirgin hale gelmektedir. RLS algoritmalarında maliyet fonksiyonu şu şekilde yazılır [9] :

(2.15)

Yukarıdaki denklemde $e(i)$ hata sinyali, $d(i)$ referans sinyali ile $y(i)$ süzgeç çıktısı arasındaki farktır. RLS algoritmasında i anındaki süzgeç girişleri

$x(i), x(i-1), \dots, x(i-M+1)$ şeklinde tanımlanır ve i anındaki hata şu şekilde gösterilir:

(2.16)

Üstteki denklemde \mathbf{a} anındaki tap giriş vektörü

(2.17)

şeklinde verilir. \mathbf{a} anındaki tap ağırlık vektörü ise şu şekilde gösterilir :

(2.18)

Denklem (2.15)'deki ağırlık faktörü şu şekilde tanımlanır :

$$\beta(n,i) \leq 1, \quad (2.19)$$

$\beta(n,i)$ unutmama faktörü veya üssel ağırlık faktörü olarak tanımlanır.

(2.20)

λ değeri pozitif ve birden küçük olduğu zaman ağırlık faktörü yakın hata kestirim değerlerine eski değerlere göre daha fazla önem verir. Yani $\lambda=1$ olduğu zaman algoritma eski değerleri unutmama eğilimi gösterir. Takip etme yeteneği olan bir uyarlamalı algoritma geliştiriliyorsa bu eğilim algoritmada olması istenilen bir özelliktir. λ algoritmanın hafızasıdır ve $\lambda=1$ sonsuz hafızaya işaret eder [9]. Minimize edilmesi gereken maliyet fonksiyonu şu hale gelir :

(2.21)

Denklem (2.21)'deki J maliyet fonksiyonu minimum değerini aldığı anda, ağırlık vektörünün optimum değeri matris formunda şu şekilde yazılabilir :

(2.22)

$M \times M$ boyutundaki \mathbf{R} korelasyon matrisi

(2.23)

şeklinde verilir. Süzgeç girişi ile referans sinyalinin $M \times 1$ boyutundaki çapraz korelasyon vektörü şu şekilde tanımlanır.

(2.24)

Denklem (2.23)'de verilen denklemin en sağ tarafındaki ifadede $i=n$ indeksli terim toplam ifadesinin dışına alınırsa

(2.25)

Denklem (2.25)'in sağ tarafındaki parantezin içi terimine eşittir ve korelasyon matrisini ifade etmektedir. Böylece giriş sinyalinin korelasyon matrisini güncelleyen yineleme denklemi elde edilir :

(2.26)

Yukarıdaki denklemdeki terimi eski korelasyon matrisini gösterir ve dışsal çarpımı güncellemede doğrulama yapılmasını sağlar. Benzer olarak giriş sinyali ile referans sinyali arasındaki çapraz korelasyon vektörü (2.24) kullanılarak hesaplanabilir.

(2.27)

Denklem (2.22)'deki ağırlık vektörü için en küçük kareler kestirimi hesaplanmak istenirse korelasyon matrisinin tersinin alınabilmesi gerekir. Fakat pratikte tap ağırlıklarının sayısı fazla ise işlem miktarı fazla olacağı için matrisin tersinin alınmasından kaçınılır. ağırlık vektörünün $n=1,2,\dots,\infty$ değerleri için tekrarlı olarak hesaplanması gerekmektedir. Yukarıda belirtilen problemleri aşmak için Matris Tersi Alma Önermesi olarak bilinen matris eşitliği kullanılabilir. Bu işlem uygulanırken dikkat edilmesi gereken önemli bir nokta korelasyon matrisini tekil yapmayacak şekilde başlangıç değeri verilmesidir. RLS algoritmasında kullanılan Matris Tersi Alma Önermesi şu şekilde açıklanabilir [9] :

A ve **B**, $M \times M$ boyutunda pozitif tanımlanmış iki matris arasında aşağıdaki ilişkinin olduğu varsayılınsın.

(2.28)

Yukarıdaki denklemde **D** matrisi $N \times N$ boyutunda pozitif tanımlanmıştır ve **C** matrisi $M \times N$ boyutundadır. Matris Tersi Alma Önermesi'ne göre **A** matrisinin tersi şu şekilde gösterilebilir:

(2.29)

Matris Tersine Alma Önermesi, (2.28)'deki gibi tanımlanan A matrisi için (2.29) kullanılarak A matrisinin tersinin ifade edilebileceğini belirtir. Bu kuram aynı zamanda Woodbury eşitliği olarak da bilinir. Tap ağırlık vektörü için en küçük kareler çözümünün hesaplanmasında kullanılan yinelemeli denklemin elde edilmesinde Matris Tersine Alma Önermesi'nden yararlanılabilir. Pozitif tanımlanmış korelasyon matrisi yardımıyla (2.26)'da verilen yinelemeli denkleme Matris Tersine Alma Önermesi uygulanabilir. İlk olarak aşağıdaki tanımlamaların yapılması gerekir.

(2.30)

Yapılan tanımlamalar Matris Tersine Alma Önermesi'nden elde edilen (2.29) yerine yazılırsa, korelasyon matrisinin tersi için yinelemeli denklem elde edilir.

_____ (2.31)

Hesaplama kolaylığı için

(2.32)

ve

_____ (2.33)

dönüşümleri yapılırsa (2.31) şu şekilde yazılabilir.

(2.34)

$M \times M$ boyutlu $P(n)$ ters korelasyon matrisi, $M \times 1$ boyutlu $k(n)$ vektörü kazanç vektörü olarak nitelendirilir. Denklem (2.34) ise RLS algoritmasının Ricatti Denklemi olarak adlandırılır. Denklem (2.33) yeniden düzenlenirse

(2.35)

denklemini elde edilir. Denklem (2.35)'e dikkat edildiğinde denklemin sağ tarafındaki parantez içerisindeki terimin $P(n)$ 'e eşit olduğu görülür. Denklem (2.35) bu bilgi ile yeniden yazılırsa

(2.36)

denklemini elde edilir. Aynı zamanda (2.32) göz önüne alındığında kazanç vektörü için kullanılacak bir eşitlik tanımlanabilir.

(2.37)

Üstteki denklemdeki $k(n)$ vektörü, giriş vektörünün korelasyon matrisinin tersiyle dönüşümü ile elde edilir. Kazanç vektörü tanımlandıktan sonra ağırlık vektörü için en küçük kareler kestirimini güncellemeye yarayacak yinelemeli denklemin türetilmesi gerekmektedir. Denklem (2.22), (2.27), (2.32) kullanılarak ağırlık vektörü:

(2.38)

şeklinde elde edilir. Denklem (2.38)'in sağ tarafındaki ilk $P(n)$ terimi yerine denklem (2.34)'deki karşılığı yazılırsa

(2.39)

denklemine ulaşılır. Son olarak denklem (2.36)'dan elde edilen eşitlik üstteki denklemde yerine yazılırsa ağırlık vektörünün güncellenmesi için gerekli olan denklem türetilebilir.

(2.40)

öncül kestirim hatasını göstermektedir ve şu şekilde ifade edilmektedir:

(2.41)

Burada içsel çarpımı $d(n)$ referans sinyalinin kestirimini göstermektedir. Bu kestirim, ağırlık vektörünün $n-1$ anındaki eski en küçük kareler kestirimine dayanır. Soncul hata, öncül hatadan farklı olarak şu şekilde tanımlanır.

(2.42)

2.3 Problemin Tanımı

Tez kapsamında gürültülü bir kanaldan geçirilen kaotik bir sinyalin, sadece gürültü hakkında belirli istatistiksel veriler bilinerek, süzgeçleme problemi ele alınacaktır [10]. Kestirilmek istenen gerçek değerli sinyal $\{X_t\}_{t \geq 1}$ ve sinyalin geçtiği kanaldaki gürültü $\{N_t\}_{t \geq 1}$ olarak tanımlansın. Kanaldaki gürültünün istatistiğinin t değerinden bağımsız olduğu ve tüm t değerleri için $E\{N_t\}=0$, $E\{N_t^2\}=\sigma^2$ olarak verildiği ve N_t örneklerinin birbirinden istatistiksel olarak bağımsız olduğu varsayalım. Girişi $\{X_t\}_{t \geq 1}$ olan gürültülü kanalın çıktısı $\{Y_t\}_{t \geq 1}$ olarak adlandırılmıştır. Gürültülü kanal girişi ve çıktısı arasındaki ilişki şu şekilde verilir :

$$Y_t = X_t + N_t \quad t=1,2,\dots \quad (2.43)$$

Notasyonda koyu gösterilen harfler d boyutlu sütun vektörlerini göstermektedir. Giriş vektörü $\mathbf{X}_t=[X_t, X_{t-1}, \dots, X_{t-(d-1)}]^T$, gürültü vektörü $\mathbf{N}_t=[N_t, N_{t-1}, \dots, N_{t-(d-1)}]^T$, çıktı vektörü $\mathbf{Y}_t=[Y_t, Y_{t-1}, \dots, Y_{t-(d-1)}]^T$ ve \mathbf{A} olarak tanımlanmıştır. Notasyonda verilen $(.)^T$ notasyonu matris devriğini ifade etmektedir. Problemden \mathbf{X}_t sinyalinin dinamik sistem modelinin bilinmediği ve sadece yukarıda verilen gürültü sinyalinin

değişinti değeri θ 'nin süzgeç tarafından bilindiği varsayılacaktır. Amaç $\{X_t\}_{t \geq 1}$ sinyalini $\{Y_t\}_{t \geq 1}$ gürültülü sinyalinden geri elde etmektir. Tez kapsamında literatürde bu amaçla kullanılan belli başlı süzgeçleme algoritmaları verilmekte ve bunların performansları incelenmekte ve karşılaştırılmaktadır.

3. DOĞRUSAL UYARLAMALI SÜZGEÇLER

Bölüm (2.3)'de verilen problemi çözmek için doğrusal ve doğrusal olmayan süzgeçler kullanılmıştır. Bu bölümde, kullanılan doğrusal süzgeç algoritmaları anlatılacaktır. Doğrusal süzgeç olarak Evrensel Sonlu Zamanlı Darbe Tepkili En Küçük Ortalama Karesel Hata süzgeci (Evrensel FIR MMSE Süzgeci), LMS algoritması baz alınarak türetilen LMS-benzeri süzgeç ve son olarak performans karşılaştırması için sinyal bilgisine sahip olduğu varsayılan Wiener süzgeci kullanılmıştır.

3.1 Evrensel FIR MMSE Süzgeci

Ortalaması sıfır olan gerçek değerli gürültü tarafından bozulmuş gerçek değerli sinyalin kestirimi sinyal işleme ve kestirim konularında temel bir problemdir. Asıl sinyal durağan olduğunda kestirim için genel kriter Ortalama Karesel Hata'dır (MSE, Mean Square Error) ve Wiener'den sonra Minimum Ortalama Karesel Hata (MMSE, Minimum Mean Square Error) üzerinde daha fazla çalışma yapılmıştır [10]. Doğrusal MMSE süzgecinin en bilinen formu FIR formunda olanıdır. FIR süzgecinin yaygınlaşmasının sebebi kullanımının kolay olmasıdır.

Pratikte doğrusal MMSE kestiricisini uygularken iki tür kısıtlama vardır. Bunlardan birincisi genelde sahip olmadığımız sinyalin birinci ve ikinci dereceden istatistiğine ait bilgi, ikincisi ise asıl sinyalin durağan olarak kabul edilmesidir. Oysa pratikte sinyaller durağan hatta stokastik bile olmayabilirler. Gürbüz minimaks ve uyarlamalı süzgeç kullanılarak yukarıda belirtilen kısıtlamalar ele alınabilir. Genel kanı sinyaldeki kararsızlığı optimize etmek için gürbüz kestirici kullanılmasıdır. Fakat gürbüz kestirici sinyaldeki kararsızlık kümesinin en kötü durumuna göre işlem yaptığı için, sinyalden bilgi öğrenilebileceği gerçeğini görmezden gelir. Aynı zamanda çoğu gürbüz kestirici kestirimde büyük

gecikmelere sebep olabilirler. Diğer taraftan uyarlamalı süzgeç, gürültülü sinyalden ve referans sinyalinden bilgi alarak katsayılarını güncelleyebileceği FIR süzgeci oluşturmaya çalışır. Asıl sinyalin kendisi olan referans sinyali hakkında bilgi olmadığından dolayı bu sistem verilen problemi çözmek için doğrudan uygulanabilir değildir. Denetlenmeyen Uyarlamalı Süzgeç (Unsupervised Adaptive Filtering) referans sinyali hakkında bilgi olmadığı durumlarda kullanılabilir, fakat bu süzgecin de uygulanabilmesi için asıl sinyal hakkında belirli istatistiksel verilerin bilinmesine ihtiyaç vardır. Bununla beraber asıl sinyalin istatistiksel özellikleri hakkında bilgi sahibi olunmadığından dolayı bu yöntem de verilen probleme uygulanabilir değildir [10].

Bu bölümde yukarıda verilen yöntemler yerine çevrim-içi öğrenme algoritması kullanılacaktır [10]. Bu sayede asıl sinyal yerine eklenen gürültü hakkında varsayım yapılacaktır. Örneğin eklenen gürültünün ortalamasının sıfır, zamandan bağımsız ve değişimsinin süzgeç tarafından bilindiği varsayılacaktır. Bu varsayımlar altında asıl sinyalin model bilgisinden bağımsız olarak, gözleme dizisinin uzunluğu arttıkça FIR süzgeci kadar iyi performans gösteren süzgeç tasarlanacaktır. Tasarlanan süzgecin performansı ne kadar iyi olursa, asıl sinyali elde etmeye o kadar yaklaşılr. Bu yöntem kullanılarak hem yukarıda belirtilen kısıtlamalar aşılabılır hem de asıl sinyale bağlı kalmadan iyi performans elde edilebilir. Bu bilgiler ışığında Evrensel FIR MMSE süzgecin tanımını şu şekilde verilebilir:

Süzgeç tasarlanırken öncelikle her t değeri için pozitif tanımlanmış olan A_t matrisi şu şekilde verilir.

(3.1)

A_t matrisi kullanılarak yine her t değeri için süzgeç katsayı vektörü şu şekilde bulunabilir:

(3.2)

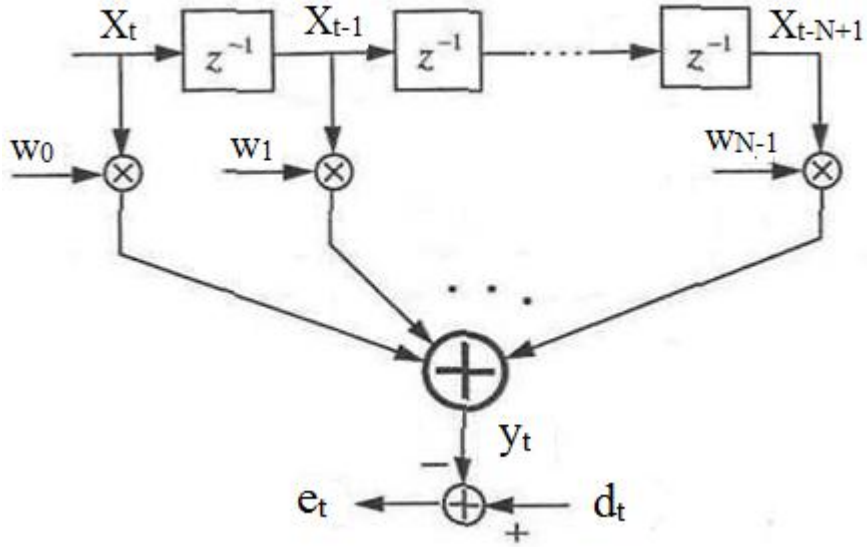
Bu tanımlamalar yapıldıktan sonra süzgeç denklemini

(3.3)

şeklini alır. Denklemde verilen terimi X_t giriş sinyalinin gürültülü gözlemlerine ($\mathbf{Y}^t=[Y_1 \dots Y_t]$) dayanan nedensel kestiricidir. Süzgeç denklemlerinden de görüldüğü gibi süzgeç tasarlanırken asıl sinyal X_t kullanılmamaktadır. Bunun yerine Y_t gürültülü sinyali ile gürültü sinyalinin değışinti değerini içeren \mathbf{c} vektörü kullanılmaktadır.

3.2 Wiener Süzgeci

Bu bölümde Wiener Süzgeci olarak bilinen optimum doğrusal süzgeç incelenecektir. Şekil (3.1)'deki enine süzgece bakılırsa üç aşamadan meydana geldiği görülebilir: depolama, çarpma ve toplama. Bu aşamalar ayrıntılı incelenirse [6] ;



Şekil 3.1 Enine süzgeç [8]

-Depolama: Depolama arka arkaya bağı N-1 gecikme birimini gösterir. Blok diyagramdaki z^{-1} gecikmeyi göstermektedir. Her gecikmeden sonra erişilen farklı noktalara tap noktaları denilmektedir. Tap girişleri $X_t, X_{t-1}, \dots, X_{t-N+1}$ olarak gösterilmektedir.

- Çarpma: Çarpma işlemi tap girişleri $X_t, X_{t-1}, \dots, X_{t-N+1}$ ile tap ağırlıkları w_0, \dots, w_{N-1} arasında içsel çarpım olarak gerçekleştirilir.

-Toplama: Toplama işlemi çarpım çıktılarının toplanarak, süzgeç için çıktı üretilmesini sağlar.

Süzgecin çıktısı Y_t , kestirim hatası ise e_t olarak ifade edilmiştir. Süzgeçteki kestirim hatası azaldıkça süzgecin performansı artmaktadır. Hata sifira yaklaşınca süzgecin çıktısı d_t referans sinyaline yaklaşmaktadır. Kestirim hatasını en aza indirmek için en uygun süzgeç parametrelerini ayarlayacak maliyet fonksiyonu tanımlanırsa bu maliyet fonksiyonunu diğer adıyla performans fonksiyonunu seçerken şu noktalara dikkat edilmesi gerekir.

- 1) Performans fonksiyonu matematiksel olarak takip edilebilir olmalıdır.
- 2) Performans fonksiyonunun bir adet minimum veya yerel maksimuma sahip olması tercih edilir.

Süzgecin analizine olanak sağladığı ve süzgeç algoritmalarının ayarlanmasında kolaylık sağladığı için performans fonksiyonunun takip edilebilirliği önemlidir. Performans fonksiyonu için minimum (veya maksimum) sayısı süzgecin yapısına bağlıdır. Yinelemeli (sonsuz zamanlı darbe tepkili) süzgeçler birden çok minimum (veya maksimum) noktasına sahiptir. Oysa ki, yinelemeli olmayan (sonlu zamanlı darbe tepkili) süzgeçleri, uygun performans fonksiyonu kullanılırsa tek bir tane minimum (veya maksimum) noktası olmasını garanti eder. Sonlu zamanlı darbe tepkili (FIR) süzgeçler girişten çıkışa giderken ileri besleme tekniğini kullanırlar. Sonsuz zamanlı darbe tepkili (IIR) süzgeçler ise girişten çıkışa giden yolda hem ileri besleme hem de geri besleme tekniği kullanırlar. Bununla beraber süzgeç uyarlamalı olduğunda, kendi kararlılık problemleri ile birlikte, uyarlama geri besleme ile birleştiğinde altından kalkması zor bir problem ortaya çıkarır. O yüzden süzgeç tasarlanırken FIR süzgeci IIR

süzgece tercih edilecektir. Wiener süzgecinde performans fonksiyonu şu şekilde seçilebilir [8]:

$$(3.4)$$

Yukarıdaki eşitlikte 'E' notasyonu istatistiksel beklentiyi ifade eder. Aslında performans fonksiyonu Ortalama Karesel Hata kriteri olarak da adlandırılabilir. Bu performans fonksiyonu hem matematiksel olarak elde edilebilir hem de tek global minimuma sahiptir. Wiener süzgecinin performansı Ortalama Karesel Hata ile ölçüldüğünden dolayı istatistiksel beklenti yerine aşağıdaki denklem yazılabilir:

$$(3.5)$$

Evrensel süzgeç bölümündeki (3.3)'den elde edilen denklemde u süzgeç katsayılarını belirtecek şekilde yeniden düzenlenirse

$$(3.6)$$

ve elde edilen denklem, (3.5)'de yerine yazılırsa

–

minimizasyon problemi elde edilir. Yukarıdaki maliyeti minimum kılacak katsayılar şu şekilde bulunabilir :

$$(3.8)$$

Ağırlık vektörü (3.8) ile verilen Wiener süzgeci tasarımında sinyalinin bilindiği varsayılacaktır ve bu süzgeç diğer süzgeçler ile performans karşılaştırması için kullanılacaktır.

Denklem (3.7)'de toplam sembolünün içindeki terim anlık karesel kayıp, J olarak adlandırılırsa

$$(3.9)$$

eşitliği elde edilir. Süzgeçte t anına kadar biriken kayıp

$$L_t(u) = \dots \quad (3.10)$$

olarak gösterilir. Wiener süzgecindeki amaç toplam kaybın minimize edilmesidir. Gürültü değişintisi bilindiğinde 'den LMS-benzeri süzgeçte kullanılacak olan anlık kayıp ifadesi elde edilebilir. Denklem (3.9), problem tanımı bölümündeki (2.43) kullanılarak tekrar düzenlenirse

$$\dots \quad (3.11)$$

eşitliği elde edilir. Bu eşitliğin en sağ tarafındaki parantezin karesi açılırsa

$$\dots \quad (3.12)$$

ve her iki tarafın beklenen değeri alınır

$$\dots \quad (3.13)$$

eşitliği elde edilir. Denklem (2.43) kullanılarak $E\{N_t Y_t\}$ ifadesi aşağıdaki gibi düzenlenebilir:

$$\dots \quad (3.14)$$

Yukarıdaki denklemde beklenen değer içindeki N_t terimi matrisin içine dağıtıldıktan sonra eşitliğin sağ tarafı şu forma ulaşır :

$$E\{N_t Y_t\} \dots \quad (3.15)$$

$E\{N_t\}=0$ ve $E\{N_{t-1}\}=0$ gürültü örnekleri istatistiksel olarak bağımsız olduklarından dolayı

(3.16)

denklemleri elde edilir. Bu bilgiler ışığında, $E\{N_t^2\}=\sigma^2$ eşitliği de kullanılarak, (3.15) şu şekilde yeniden yazılabilir:

$$E\{N_t Y_t\} \quad . \quad (3.17)$$

Denklem (3.13)'de $E\{N_t Y_t\}$ yerine \mathbf{c} vektörü yazılıp yeniden düzenlenirse

(3.18)

elde edilir. Üstteki denklemin sol tarafı algoritmada minimize edilmek istenen karesel hata değerine eşittir. Yani (3.18)'e göre σ^2 değeri biliniyorsa \mathbf{u} ağırlık vektörü seçilerek

(3.19)

ifadesi minimize edilebilir. Beklenen değer hesaplanması için elde istatistiksel bilgi olmadığından beklenen değer kaldırılırsa

(3.20)

elde edilir. Buradaki terimi denklemin (3.6)'da verilen süzgeçten kaynaklı kestirilen anlık kaybı niteler. Denklem (3.20) sonraki bölümde LMS-benzeri süzgeci türetirken kullanılacaktır.

3.3 LMS-Benzeri Süzgeç

Bu bölümde LMS süzgecinin algoritmasından yola çıkılarak türetilen LMS-benzeri Süzgeç algoritması anlatılacaktır. Süzgeç algoritmasının türetim ve süzgeçleme işlemi için bölüm 2.3’de verilen problem ele alınmıştır Şekil 2.1’den yararlanılarak kestirim hatası yapılrısa

$$(3.21)$$

ve elde edilen denklem notasyon olarak düzenlenerek Evrensel FIR MMSE süzgeç bölümünden elde edilen süzgeç denklemi (3.3) kullanılırsa

$$(3.22)$$

denklemi elde edilir. Bu süzgeçten kaynaklanan karesel kayıp (hata) şu şekilde verilir :

$$(3.23)$$

Karesel hata değerini minimum kılacak tap ağırlık değerlerini bulmak için (3.23)’ün ağırlık değerlerine göre kısmi türevleri alınabilir.

$$\begin{array}{ccc} & \text{—} & \text{—} \\ & & \\ \text{—} & & \\ & \text{—} & \text{—} \end{array} \quad (3.24)$$

Hata değerini veren (3.22)’nin tap ağırlıklarına göre türevi alınırsa

$$\begin{array}{ccc} \text{—} & \text{—} &) \\ & & \\ \text{—} & & \end{array} \quad (3.25)$$

elde edilir. Burada LMS algoritmasından elde edilen tap ağırlık güncellemesi denklemi (2.11) ve yukarıdaki (3.24) göz önüne alınırsa

$$— \quad (3.26)$$

denkleminde ulaşılabilir. Üstteki denklemdeki terimi öğrenme hızı parametresidir. Denklem (3.26)'daki kısmi türev işlemi uygulanırsa

$$— \quad (3.27)$$

ve (3.27)'nin en sağındaki terimin yerine (3.25) yazılırsa

$$(3.28)$$

elde edilir. Denklem (3.28) incelendiği zaman LMS algoritması ağırlık vektörünü güncelleyen (2.11) ile aynı formatta olduğu görülebilir. Denklem (3.26)'daki yerine önceki bölümden elde edilen (3.20) yazılabilir. Denklem (3.26), bu şekilde düzenlenir ve yeniden yazılırsa

$$\begin{aligned} &— \\ &— \end{aligned} \quad (3.29)$$

elde edilir. Bu ifade kullanılan \mathbf{Y}_t ve \mathbf{c} vektörleri aşağıda verilmiştir. Bu vektörler $dx1$ boyutundadırlar ;

$$\cdot \quad (3.30)$$

Denklem (3.29)'daki kısmi türev işlemi gerçekleştirilirse

$$(3.31)$$

ve iç taraftaki parantez açılıp yeniden düzenlenirse

$$(3.32)$$

denklemleri elde edilir. Yukarıdaki denklemlerde \mathbf{I} terimi şu şekilde düzenlenebilir :

$$\mathbf{I} = \mathbf{I} \quad (3.33)$$

Yukarıdaki düzenleme (3.32)'ye uygulanırsa ve \mathbf{I} terimi parantez içine alınırsa

$$\mathbf{I} = \mathbf{I} \quad (3.34)$$

elde edilebilir. Yukarıdaki denklem LMS-benzeri süzgeç için ağırlık güncellemesini vermektedir.

Ağırlık güncelleme denklemleri (3.34) ile (3.2) hesaplama miktarı (maliyet) yönünden karşılaştırılsa (3.34)'ün bu yönden (3.2)'den daha iyi olduğu görülebilir. Çünkü (3.2)'nin maliyeti $O(d^2)$ iken, (3.34)'ün maliyeti $O(d)$ mertebesindedir. Yani LMS-benzeri süzgeç için kullanılan ağırlık güncelleme denklemleri (3.34), Evrensel FIR MMSE süzgecinin ağırlık güncelleme denklemleri (3.2)'den daha az hesaplama gerektirmektedir.

4. DOĞRUSAL OLMAYAN UYARLAMALI SÜZGEÇLEME YÖNTEMLERİ

Günlük hayatta ölçülen zaman serileri genelde doğrusal değildir. Bu tezde görülebileceği gibi bazen kaotik de olabilirler. İstenen bilgiyi ölçülen bilgiden ayırmak için yapılan ölçümden gürültünün ayrılması gerekmektedir. Zaman serileri doğrusal olmadığında özellikle de kaotik özellik gösterdiğinde doğrusal alçak geçiren süzgeçler gürültünün azaltılması için uygun bir yöntem olarak görülmemektedir. Çünkü kaotik sinyaller genelde geniş bantlı spektruma sahiptirler ve gürültünün spektrumu ile üst üste gelirler. Bu sebepten dolayı doğrusal alçak geçiren süzgeçler kullanılarak yapılan süzgeçleme işlemi sırasında kaotik sinyal de bozulur. Kaotik sinyalin bozulmasını en aza indirmek için doğrusal olmayan uyarlamalı süzgeçleme yöntemleri kullanılmıştır. Bu bölümde kullanılan doğrusal olmayan uyarlamalı süzgeç yöntemlerinden Polinom Uydurma (Çatma) Yöntemi ve Dalgacık Katsayısı Sıkıştırma Yöntemi anlatılacaktır.

4.1 Polinom Uydurma Yöntemi

Bu bölümde, kullanılan doğrusal olmayan süzgeçleme yöntemlerinden birincisi olan Polinom Uydurma Yöntemi anlatılacaktır. Polinom Uydurma Yöntemi sinyal işlemede ve nümerik analizde yaygın olarak kullanılan bir araçtır. Bu yöntemin uygulanması, en küçük kareler bağlamında rastgele seçilmiş bir sinyal için belirlenen bir dereceden en iyi polinomun bulunması ile gerçekleştirilir. Bu bölümde kullanılan Polinom Uydurma Yöntemi şu şekilde kullanılmıştır. Alınan gürültülü sinyal $2n+1$ nokta içeren segmentlere (pencerelere) bölünür. Segmentlere bölünme esnasında komşu segmentler birbirleri ile $n+1$ noktada üst üste gelecek şekilde ayarlanır [11].

Herbir segment için K 'nıncı dereceden en iyi polinom uydurulur. Polinom uydurma esnasında kullanılan ' K ' değeri $K=0$ için parçalı sabit polinom, $K=1$ değeri için doğrusal polinom uydurmaya denk gelmektedir.

Süzgeçleme işlemi için bölüm 2.3'te verilen problem kullanılırsa ve aynı bölümdeki (2.43)'de verilen gürültülü sinyal için 3'üncü dereceden polinom uydurulursa

$$(4.1)$$

denklemini elde edilir. Yukarıdaki denklemde a_0, a_1, a_2, a_3 , değerleri uydurulan polinomun katsayılarını belirtmektedir. Denklem (4.1) içsel çarpım formunda yazılırsa

$$(4.2)$$

ve bu form $(2n+1)$ örneklilik segmentler için düşünülürse

$$(4.3)$$

denklemini oluşturulur. Üstteki denklemde eşitliğin sol tarafına \mathbf{b} vektörü, eşitliğin sağ tarafındaki ilk terime \mathbf{A} matrisi, sağ taraftaki son terime de \mathbf{a} vektörü denilirse ve katsayıları içeren \mathbf{a} vektörü (4.3)'den çekilirse

$$(4.4)$$

denklemine ulaşılır. Denklem (4.4) ile polinom katsayıları elde edildikten sonra, polinom uydurma için ayrılan segmentlerden, i 'nci ve $i+1$ 'inci segmentler sırasıyla \mathbf{a}_i ve \mathbf{a}_{i+1} olarak adlandırılabilir. Burada $i = 1, 2, \dots, 2n+1$ değerlerini almaktadır. Segmentlerde üst üste binen bölgeler için uydurma şu şekilde tanımlanabilir:

$$(4.5)$$

Yukarıdaki denklemde ω_1 ve ω_2 değerlerini almaktadırlar [11]. ω_1 ve ω_2 terimlerinin genel formu şu şekilde yazılabilir:

$$— \quad (4.6)$$

Buradaki d_j terimi o andaki noktanın sırasıyla $y(i)$ ve $y(i+1)$ merkezlerine olan uzaklığını belirtmektedir. Yani bulunulan nokta ile segmentin merkezi arasındaki ağırlık doğrusal olarak düşmektedir. Bu şekildeki bir ağırlık değeri simetri sağlar ve komşu segmentlerin sınırları etrafındaki atlamaları ve süreksizlikleri ortadan kaldırır.

4.2 Dalgacık Katsayısı Sıkıştırma Yöntemi

Dalgacıklar, bir sinyalin içerdiği bilgiyi farklı frekans bileşenlerine ayıran ve sonra kendi ölçekleriyle eşleştirilmiş bir çözünürlüğe sahip bileşenler üzerinde çalışan matematiksel fonksiyonlardır. Dalgacık yönteminin arkasındaki temel fikir, sinyalin belirlenen bir ölçeğe göre analizinin yapılmasıdır [12].

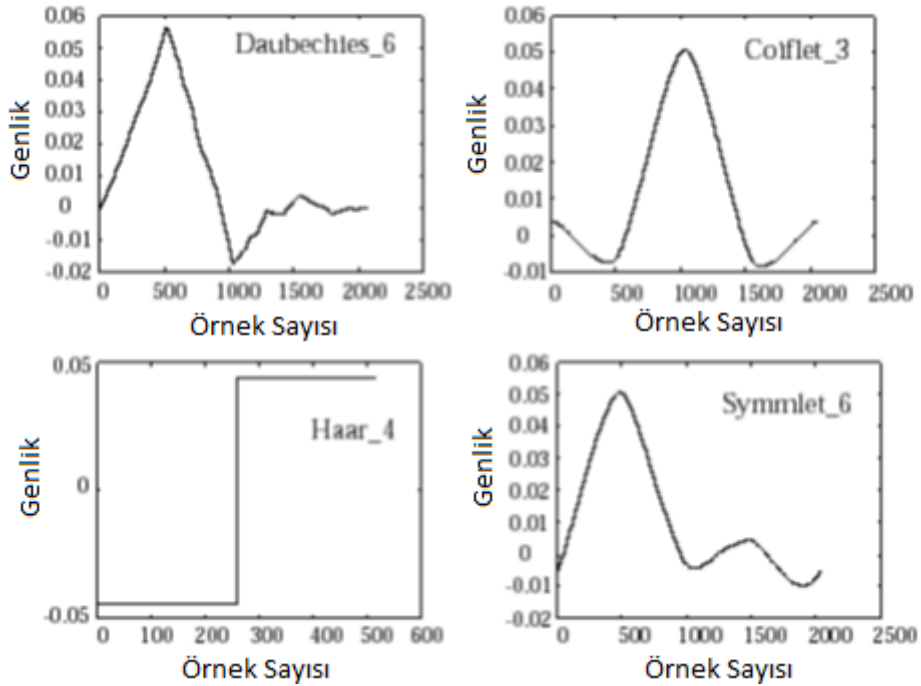
Dalgacık kullanımından önce sinyal analizlerinde Fourier Dönüşümleri yaygın olarak kullanılmaktaydı. Bir sinyal, Fourier Dönüşümü kullanılarak sonsuz sinüs ve kosinüslerin toplamı olarak elde edildiğinden dolayı sinyal içerisindeki frekans bileşenleri Fourier Dönüşümü kullanılarak elde edilebilir. Fakat Fourier dönüşümü ile sinyalden zaman bilgisi kaldırılıp sadece frekans bilgisi elde edildiğinden dolayı sinyal durağan olduğunda güzel sonuçlar vermektedir. Fourier Dönüşümü uygulanan sinyaldeki tüm frekanslar bulunabilir ancak bu frekansların hangi zaman diliminde meydana geldikleri bilinemez [12].

Fourier Dönüşümü bu özelliğinden dolayı durağan olmayan sinyallerde frekans bileşenlerinin hepsini yakalayamamaktır. Durağan olmayan sinyallerde frekans bileşenlerinin hepsini elde edebilmek için bir süre Kısa Zamanlı Fourier Dönüşümü (KZFD) kullanılmıştır. KZFD yöntemi ile sinyal içerisindeki zaman bilgisi alınabilmektedir fakat bu yöntemin sağlıklı kullanımı için zaman çözünürlüğünün sinyal boyunca aynı olması gerektiğinden dolayı sinyal içerisindeki ani frekans değişimlerinin yakalanmasında bu yöntemin yetersiz

kaldığı görülmüştür. Ayrıca KZFD yönteminin kullanımı sırasında sinyali kesmek için kullanılan pencere sinyali ile orjinal sinyal arasında bir konvolüsyon oluşmaktadır ve bu frekans düzleminde kalıntı sinyallere sebep olmaktadır [12].

Dalgacık yönteminin kullanımı ile Fourier Dönüşümü ve KZFD yöntemindeki problemler giderilerek sinyal içerisindeki frekans bileşenlerinin zamanları ve genlikleri belirlenebilir.

Teorik olarak sıfır ortalamalı ve sonlu enerjiye sahip herhangi bir fonksiyon dalgacık sayılabilir [13]. Pratikte dalgacığı seçmek için birçok ölçüt vardır. Dalgacığın zaman ve frekans ortamında iyi konuşlanması için bu ortamlarda hızlı sönümlenmesi gerekir. Dalgacık dönüşümü değişen ölçek değerine göre kullanılan pencereyi değiştirerek yukarıda anlatılan KZFD yöntemindeki zaman ve frekans problemlerini çözmektedir. Aşağıda dalgacık türlerinden örnekler verilmektedir.



Şekil 4.1. Dalgacık türlerine örnekler [13]

Dalgacık Dönüşümleri analiz yöntemlerine göre sayısal uygulamalarda kullanılan Ayrık Dalgacık Dönüşümü (ADD) ve analog uygulamalarda kullanılan Sürekli Dalgacık Dönüşümü (SDD) olarak ikiye ayrılır.

4.2.1 Sürekli dalgacık dönüşümü

Sürekli Dalgacık Dönüşümü (SDD) sinyal ile dalgacık fonksiyonunun kaydırılmış ve ölçeklendirilmiş versiyonlarının çarpılması olarak tanımlanır. Sürekli Dalgacık Dönüşümü şu şekilde verilir :

$$(4.7)$$

Ters Dönüşüm Denklemi

$$— — (4.8)$$

şeklinde elde edilir. Bu denklemde verilen bir dalgacık sabiti olup seçilen dalgacık türüne bağlıdır ve — uygunluk şartını sağlamalıdır [14]. Yukarıda dalgacık dönüşümü denklemlerinde verilen , analiz fonksiyonunu göstermektedir.

Denklem (4.7)'de verilen $x(t)$ sinyalinin, zamanda kaydırılan ve ölçeklendirilen ψ dalgacığı ile çarpılması görülmektedir. Sinyal süreci boyunca kaydırılan ve ölçeklendirilen analiz dalgacığı ψ fonksiyonu şu şekilde ifade edilir.

$$= — (4.9)$$

Burada 'a' ve 'b' değerleri ölçeklendirmeyi ve ötelenmeyi sağlar. — terimi farklı ölçekler için enerji normalizasyonunu yapar. Denklem (4.9)'daki ölçek faktörü 'a' büyüdükçe dalgacık genişler küçüldükçe dalgacık daralır.

4.2.2 Ayrık dalgacık dönüşümü

Ayrık Dalgacık Dönüşümü dalgacık dönüşümünü bilgisayarda gerçeklemek için geliştirilmiştir. Dönüşüm yapılırken ölçek değerleri ikinin üstelleri şeklinde seçilirse, istenmeyen veriler elimine edilerek daha hızlı analiz yapılması sağlanabilir.

Denklem (4.7) ve (4.8)'deki sürekli dalgacık eşitliklerinde 'a' ölçek katsayısı yerine 2^{-j} ve 'b' öteleme katsayısı yerine de $k2^j$ yazılırsa :

$$= — — — (4.10)$$

denklemleri elde edilir. Ayrık dalgacık dönüşümü

(4.11)

ve ayrık dalgacık ters dönüşümü

(4.12)

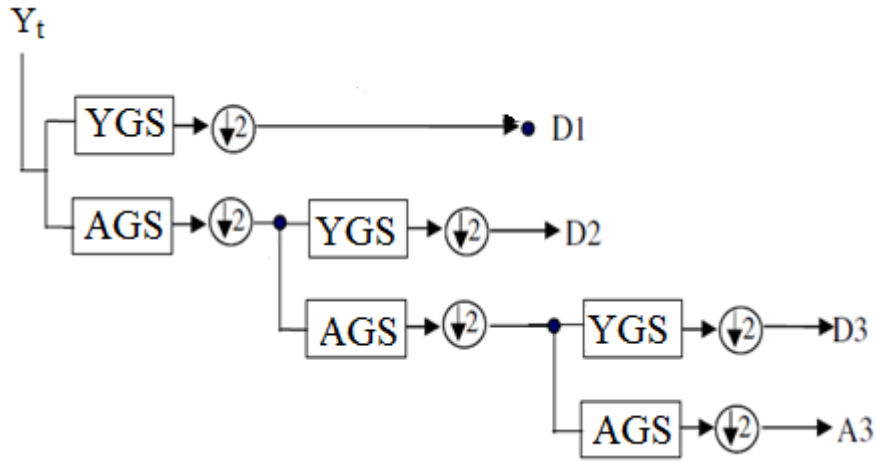
şeklini alır. Burada j indisi dalgacığın ölçeğini belirlemektedir ve j indisinin değeri arttıkça ölçek değeri azalmaktadır.

Dalgacık fonksiyonları bant geçiren bir özellik gösterirler. Fourier dönüşümünün aşağıda (4.13) ile verilen özelliğine göre, bir işareti zaman tanım bölgesinde genişletmek, frekans tanım bölgesinde aynı oranda daraltmaya karşılık gelmektedir [15].

— —

(4.13)

Ayrık dalgacık dönüşümünde (ADD) her seviye arttırımında dalgacık 2 kat genişlemekte ve yukarıdaki eşitliğe göre frekans bandı yarıya düşmektedir. Sürekli Dalgacık Dönüşümü'nde, zaman ekseninde kayarak ilerleyen ve değişen ölçek parametresi ile bu işlemi tüm frekanslar için tekrarlayan dalgacık sayesinde sinyaldeki frekanslar ayrılabilir. ADD'de, bu işlem alçak geçiren ve yüksek geçiren süzgeçleri kullanarak gerçekleştirilir. ADD'de Alçak Geçiren Süzgeç (AGS) düşük frekansları geçirmek için, Yüksek Geçiren Süzgeç (YGS) ise yüksek frekansları geçirmek için kullanılır. Denklem (4.11)'de verilmiş olan ayrık dalgacık dönüşümündeki amaç alçak geçiren ve yüksek geçiren süzgeç katsayılarını bulmaktır [15]. Sinyalin AGS ve YGS kullanılarak altbantlara ayrılması, Mallat Algoritması olarak adlandırılmaktadır ve şekil (4.2)'de gösterilmektedir.



Şekil 4.2. Mallat Algoritması [12]

AGS'den geçirilerek elde edilen sinyale yaklaşım denilmektedir. Çünkü AGS'den geçildikten sonra elde edilen sinyal süzgeçlenmemiş sinyalin ana hatlarını içermektedir. YGS'den geçirilerek elde edilen sinyale ise detay denilmektedir. Örnek olarak insan sesi ele alınırsa; insan sesinden yüksek frekanslar kaldırıldığında konuşmanın içeriği anlaşılabilir. Ancak düşük frekanslar kaldırılırsa içeriği anlaşılmayan, anlam verilemeyen sesler duyulur. Dalgacık dönüşümünde de yaklaşımlar sinyalin orijinalini temsil eder ve yüksek ölçek ile elde edilir. Detaylar ise düşük ölçekteki bilgilerdir ve sinyalin yüksek frekanslı bileşenleridir [12].

Sinyali AGS ve YGS kullanarak altbantlara ayırma işlemi sırasında, her bir ayrıştırma aşamasında orjinal sinyalden alınan örnek sayısı yarıya düşmektedir. Ayrık dalgacık sentezi kullanılarak ayrıştırılan sinyallerden orjinal sinyal tekrar elde edilebilir.

4.2.3 Dalgacık katsayısı sıkıştırma yöntemi ile gürültü giderme

Örnekleme zamanı τ olan ve içinde hem orjinal sinyal X_t hem de ω_t gürültüsünü barındıran Y_t gürültülü sinyali ele alırsak

$$Y_t = X_t + \omega_t \quad t=1, \dots, N \quad (4.13)$$

bu sinyale Dalgacık Katsayısı Sıkıştırma yöntemi 3 aşamada uygulanabilir [11]:

1. Gözlemlenen sinyalin dalgacık dönüşümünün alınması
2. Elde edilen dalgacık katsayılarına eşikleme uygulanması
3. Sinyalin kestiriminin elde edilmesi için ters dalgacık dönüşümü uygulanması

Bu aşamaları daha ayrıntılı incelemek için örnek olarak alınan Y_t gürültülü sinyale Dalgacık Katsayısı Sıkıştırma yöntemi uygulanırsa:

İlk aşamada Y_t gürültülü sinyale dalgacık dönüşümü uygulanarak yaklaşım ve detay katsayıları elde edilebilir.

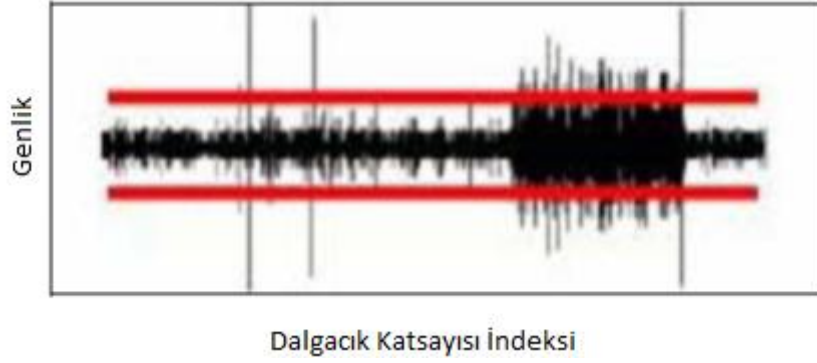
$$Y_t = SA_J + SD_J + SD_{J-1} + \dots + SD_1 \quad (4.14)$$

Yukarıdaki denklemde J en büyük ölçeklendirme olarak belirlenmiştir. SA_J terimi, ' $2^J \tau$ ' zaman ölçeklemesine karşılık gelen yaklaşım katsayılarını, SD_j terimi, $j=J, J-1, \dots, 1$, ' $2^{j-1} \tau$ ' zaman ölçeklemesine karşılık gelen detay katsayılarını göstermektedir.

Daha sonra dalgacık dönüşümü ile elde edilen detay katsayılarına eşikleme uygulanır. Katsayılara uygulanacak eşikleme şu şekilde tanımlanabilir.

$$D(U, \lambda) = \text{sgn}(U) \max(0, |U| - \lambda) \quad (4.14)$$

Burada U dalgacık detay katsayılarını, $\text{sgn}(U)$ ise $U > 0$ için 1, $U < 0$ koşulu için ise -1 değerini alan işaret (signum) fonksiyonunu belirtmektedir. λ terimi eşik değerini belirtmektedir ve 3 tane alternatif yöntem kullanılarak seçilebilir. Aşağıda sinyale eşiklemenin uygulanması görülmektedir.



Şekil 4.3. Eşik değerinin uygulanması [14]

- a. λ terimini belirlemede kullanılan en popüler yöntem olarak

$$\lambda = \sigma \sqrt{\frac{2 \ln N}{N}} \quad (4.15)$$

denklemini verilebilir. Denklemdaki N değeri sinyaldeki toplam nokta sayısını, σ değeri ise gürültünün standart sapma değerini vermektedir.

- b. Eşik değerini bulmak için kullanılan bir diğer yöntem ise σ standart sapma değerinin yerine j ölçeklendirilmesindeki detay katsayılarının standart sapma değeri olan σ_j değerinin konulmasıdır.

$$\lambda_j^{[1]} = \sigma_j \sqrt{\frac{2 \ln N}{N}} \quad (4.16)$$

- c. Son olarak Mallat Algoritmasındaki ilk ölçeklendirmede detay katsayıları için (4.15) denkleminin eşik değeri olarak, diğer ölçeklerde ise aşağıdaki eşik değerinin kullanılması ile elde edilebilir.

$$\lambda_j^{[2]} = \sigma_j \sqrt{\frac{2 \ln N}{N}} / \beta \quad (4.17)$$

Denklemda $j=J$ için $\beta = 1$ ve $j=2, \dots, J-1$ için $\beta = 2^{j-1}$ olarak ifade edilmektedir. Son olarak ters dalgacık dönüşümü uygulanarak sinyalin kestirimi elde edilir.

5.SİMULASYON (BENZETİM) SONUÇLARI

Bu bölümde, önce süzgeç algoritmalarında parametrelerin nasıl seçildiği anlatılacak daha sonra ise iki tane gürültülü kaotik sinyali (Lojistik regresyon sinyali ve Henon sinyali) süzmek için önceki bölümlerde teorik olarak bahsedilen yöntemlerin MATLAB programında performans karşılaştırması yapılacaktır. Performans karşılaştırması iki bölümde incelenecektir. Birinci bölümde her süzgeç farklı değişinti değeri kriteri altında incelenecek, ikinci bölümde ise aynı değişinti (varyans) değerleri için farklı süzgeçlerin performansları karşılaştırılacaktır. Birinci bölüm için yedi farklı süzgeç programının simülasyonu yapılmıştır. Simülasyonu yapılan süzgeç programları sırasıyla Evrensel FIR MMSE süzgeç, Wiener süzgeci, LMS-benzeri süzgeç, Polinom Uydurma yöntemi ve Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeçlerdir. Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeç programı eşik (λ) değerine bağlı olarak üç program ile simule edilmiştir (programlara sırasıyla p1,p2...p7 isimleri verilmiştir). İkinci bölümde ise Dalgacık Katsayısı Sıkıştırma programı için tek eşik (λ) değeri alınarak (faklı değişinti değerleri için Ortalama Karesele Hata değerleri birbirine yakın olduğundan dolayı) beş program ile simülasyon yapılmıştır. Simülasyonlarda değişinti değerleri 0.1, 1, 2.5ve 5 olarak alınmış ve sırasıyla var01,var1,var25 ve var5 şeklinde gösterilmiştir.

Gürültülü Lojistik regresyon sinyalini süzmek için tasarlanan süzgeçlerin algoritmaları Ek-1 - Ek-7'de gösterilmiştir. Ek-8'de bu süzgeç programlarının herbiri için değişinti değerini değiştiren program bulunmaktadır (Ek-8'de süzgeç algoritması seçimi manuel olarak yapılmaktadır). Ek-14'te gürültülü Lojistik regresyon sinyali için sabit değişinti değeri altında süzgeç programlarını karşılaştıran algoritma bulunmaktadır. Ek-9 - Ek-13'te ise Ek-14 tarafından kullanılan süzgeç algoritmaları bulunmaktadır (her değişinti değeri için Ek-9 - Ek-13'teki programlarda manuel olarak değişiklik yapılmaktadır). Benzer şekilde gürültülü Henon sinyalini süzmek için tasarlanan süzgeç algoritmaları Ek-15 - Ek-21'de gösterilmiştir. Ek-22'de bu süzgeç programlarının herbiri için değişinti değerini değiştiren program bulunmaktadır (Ek-22'de süzgeç algoritması seçimi

manuel olarak yapılmaktadır). Ek-28’de gürültülü Henon sinyali için sabit deęişinti deęeri altında (her deęişinti deęeri için Ek-23 - Ek-27’deki programlarda manuel olarak deęişiklik yapılmaktadır) süzgeç programlarını karşılaştıran algoritma bulunmaktadır. Ek-23 - Ek-27’de ise Ek-28 tarafından kullanılan süzgeç algoritmaları bulunmaktadır.

5.1 Süzgeç Algoritmalarında Parametre Seçimi

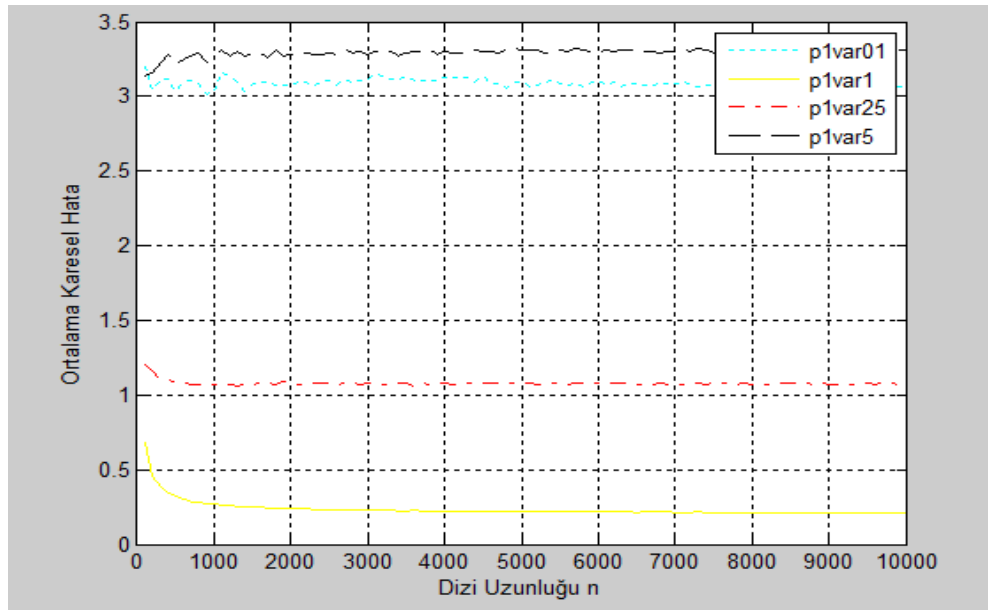
Tasarlanan süzgeçler için parametre seçimi yapılırken süzgeçlerin çıktı sinyalleri (kaotik sinyalin kestirimi) girdi sinyallerinden (gürültülü kaotik sinyal) çıkarılarak, gürültü sinyalinin bir kestirimi elde edilmiş ve bu kestirimin deęişinti deęeri hesaplanmıştır. Hesaplanan bu deęişinti deęerini, bilinen gürültü deęişintisine yakın kılan süzgeç parametre deęerleri, iyi performans sağlayan (MSE deęerlerini küçük kılan veya korelasyon boyutlarını yakın kılan) deęerler olarak süzgeç algoritmalarında kullanılmıştır.

5.2 Aynı Süzgecin Performansının Farklı Deęişinti Deęeri Kriteri Altında Karşılaştırılması

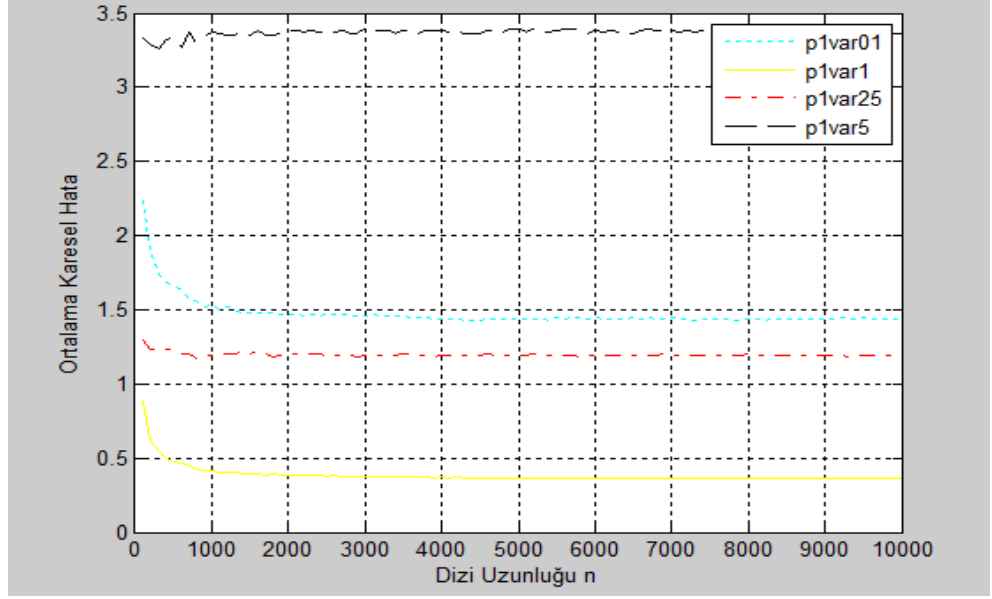
Bu bölümde kaotik sinyallere etkiyen gürültünün deęişinti deęeri deęiştirilerek süzgeçlerin performansları karşılaştırılacaktır. Ortalama Karesel Hata (MSE) deęerleri sinyal uzunluęuna göre çizdirilirken herbir MSE deęeri, o sinyal uzunluęu için yapılan 100 ayrı simülasyon sonucunun aritmetik ortalaması olarak hesaplanmıştır.

5.2.1 Evrensel FIR MMSE süzgeç simülasyonları

1-boyutlu Lojistik regresyon ve 2-boyutlu Henon gürültülü sinyallerini süzmek için tasarlanan Evrensel FIR MMSE süzgecin, değişik değışinti değeri altında verdiği Ortalama Karesel Hata (MSE) değeri sinyal uzunluđuna karşı ařađıdaki řekillerde çizdirilmiştir.



řekil 5.1. Gürültülü Lojistik regresyon sinyali için tasarlanan Evrensel FIR MMSE süzgecin farklı değışinti değeri altındaki Ortalama Karesel Hata değeri gösterimi



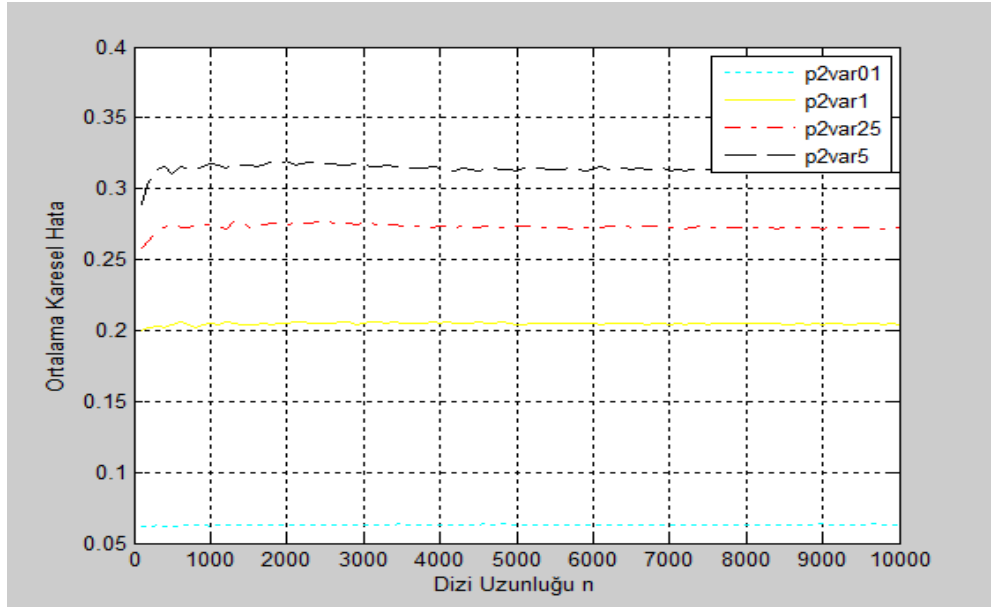
Şekil 5.2. Gürültülü Henon sinyali için tasarlanan Evrensel FIR MMSE süzgecin farklı değışinti değeri altındaki Ortalama Kareysel Hata değeri gösterimi

Yukarıdaki şekillerde görüldüğü gibi gürültülü Lojistik regresyon ve Henon sinyallerini süzmek için tasarlanan Evrensel süzgeç, sırasıyla değışinti değeri=5 (var5) ve değışinti değeri=0.1 (var01) değeri aldığında en yüksek Ortalama Kareysel Hata (MSE) değeri vermektedir. Evrensel FIR MMSE süzgeç değışinti değeri=1 için ise, en küçük MSE değeri vermektedir.

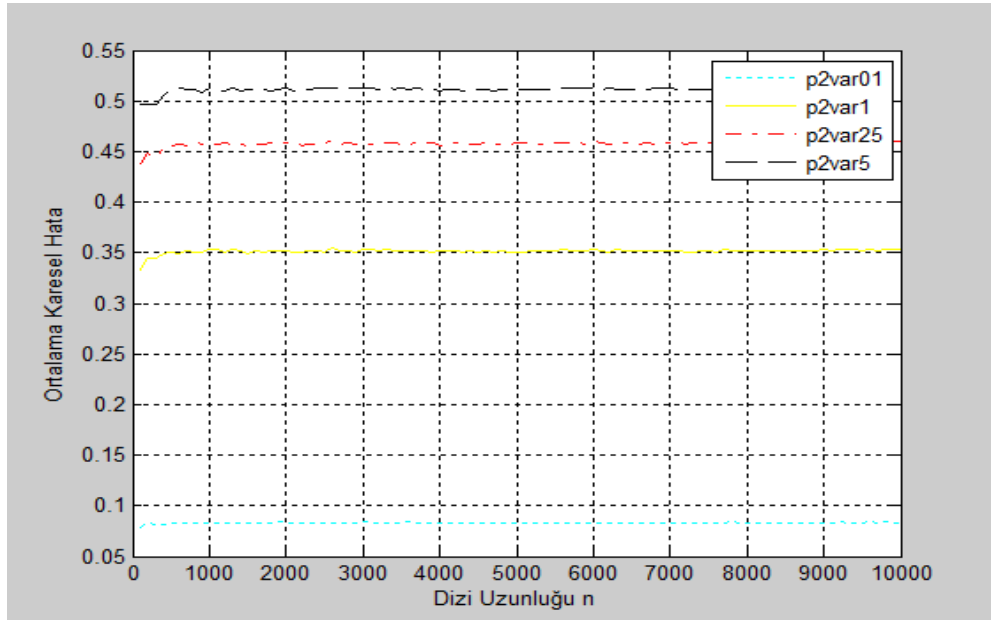
5.2.2 Wiener süzgeç simülasyonları

Wiener süzgeci daha önceden belirtildiği gibi sinyalin tamamen bilindiği varsayılarak tasarlanmıştır. Sinyale ait belirli istatistiksel değeri bilinerek tasarlanan diğer süzgeçlerin MSE değeri ile Wiener süzgecinin MSE değeri karşılaştırılması yapılmıştır.

Lojistik regresyon ve Henon gürültülü sinyallerini süzmek için tasarlanan Wiener süzgecinin değışik değışinti değeri altında verdiği MSE değeri aşağıdaki grafiklerde gösterilmiştir.



Şekil 5.3. Gürültülü Lojistik regresyon sinyali için tasarlanan Wiener süzgecinin farklı değışinti değeri altındaki Ortalama Karese Hata değeri gösterimi

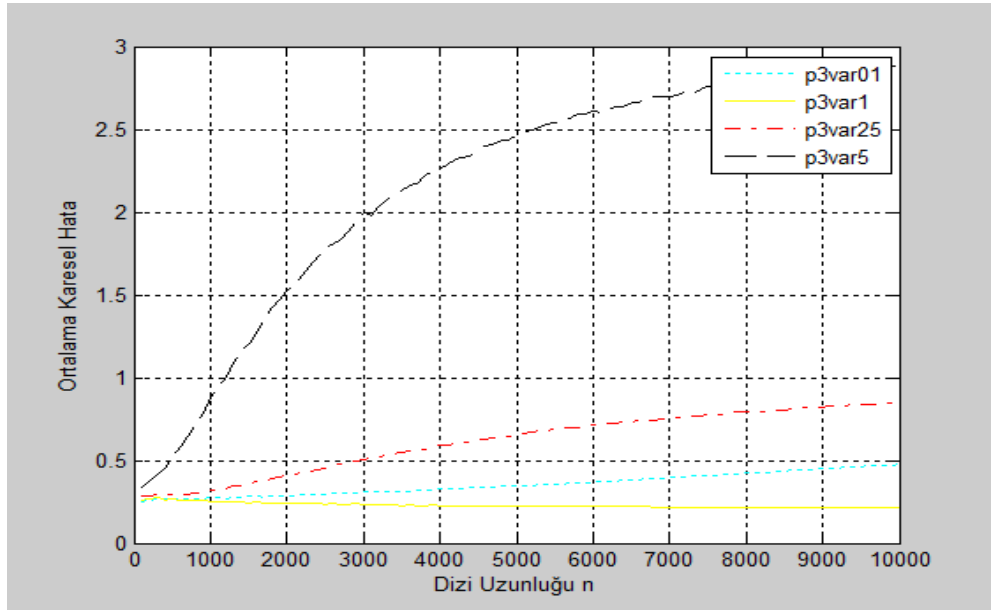


Şekil 5.4. Gürültülü Henon sinyali için tasarlanan Wiener süzgecinin farklı değışinti değeri altındaki Ortalama Karese Hata değeri gösterimi

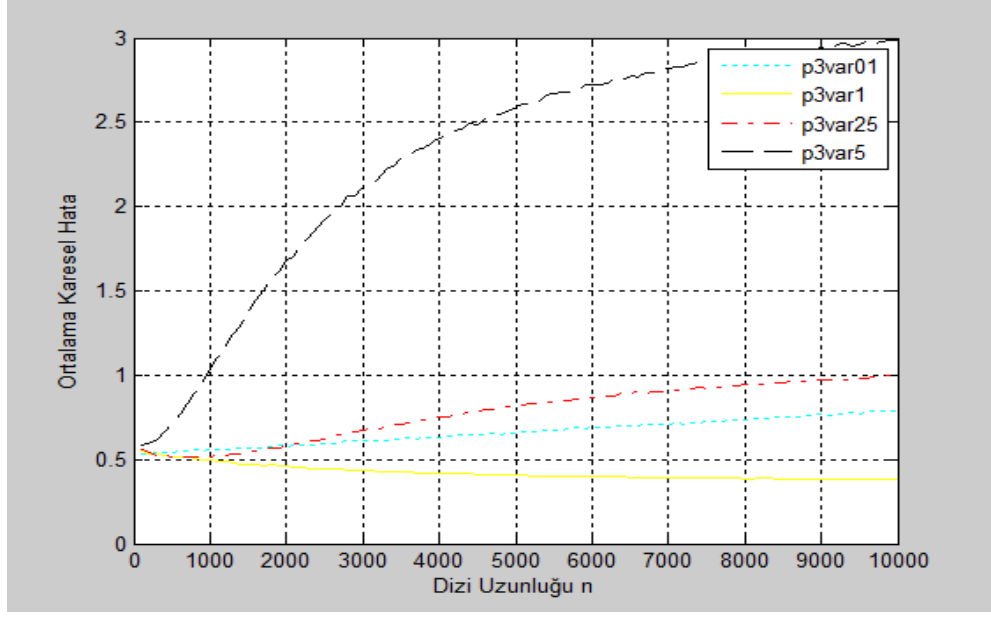
Şekil 5.3 ve Şekil 5.4 'de görüldüğü gibi değişinti değerinin artması Wiener süzgeci için MSE değerinin artmasına sebep olmaktadır.

5.2.3 LMS-benzeri süzgeç simülasyonları

LMS Algoritmasından yola çıkılarak LMS-benzeri süzgeç türetilmiştir. Lojistik regresyon ve Henon gürültülü sinyallerini süzmek için tasarlanan LMS-benzeri süzgecin değişik değişinti değerleri altında verdiği MSE değerleri aşağıdaki grafiklerde gösterilmiştir.



Şekil 5.5. Gürültülü Lojistik regresyon sinyali için tasarlanan LMS-benzeri süzgecin farklı değişinti değerleri altındaki Ortalama Karesel Hata değerleri gösterimi

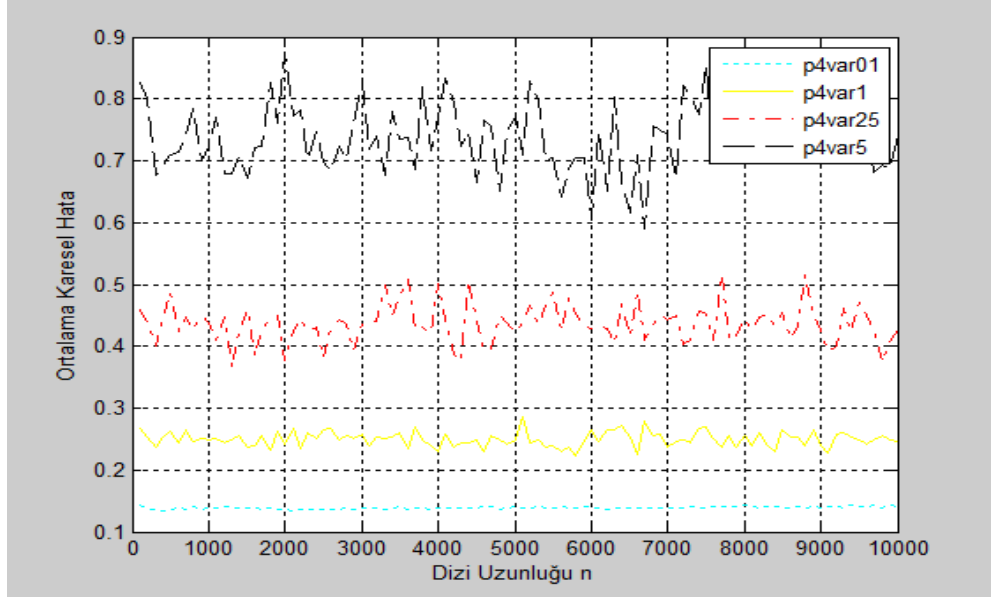


Şekil 5.6. Gürültülü Henon sinyali için tasarlanan LMS-benzeri süzgecin farklı değışinti değeri altındaki Ortalama Kareysel Hata değeri gösterimi

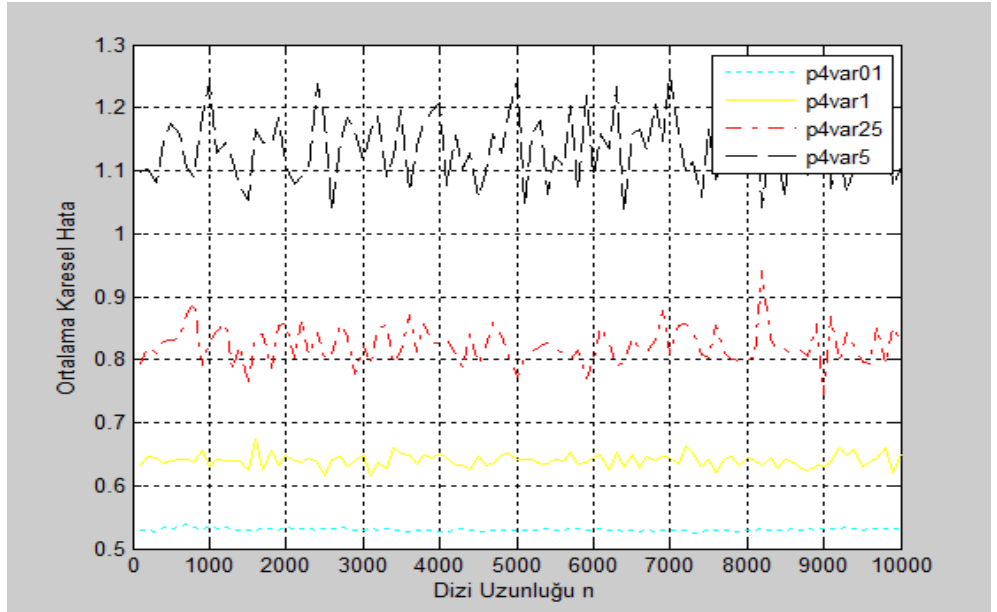
Şekil 5.5 ve Şekil 5.6'da görüldüğü gibi LMS-benzeri süzgeç en iyi performansını değışinti değeri=1 (var1) olduğunda en kötü performansını ise değışinti değeri=0.1(var01) olduğunda göstermiştir.

5.2.4 Polinom uydurma yöntemi simülasyonları

Lojistik regresyon ve Henon gürültülü sinyallerini süzmek için tasarlanan ve doğrusal olmayan süzgeç yöntemleri arasında anlatılan Polinom Uydurma yöntemi ile tasarlanan süzgecin değışik değışinti değeri altında verdiği MSE değeri aşağıdaki grafiklerde gösterilmiştir.



Şekil 5.7. Gürültülü Lojistik regresyon sinyali için Polinom Uydurma yöntemi ile tasarlanan süzgecin farklı değışinti değeri altındaki Ortalama Karese Hata değeri gösterimi

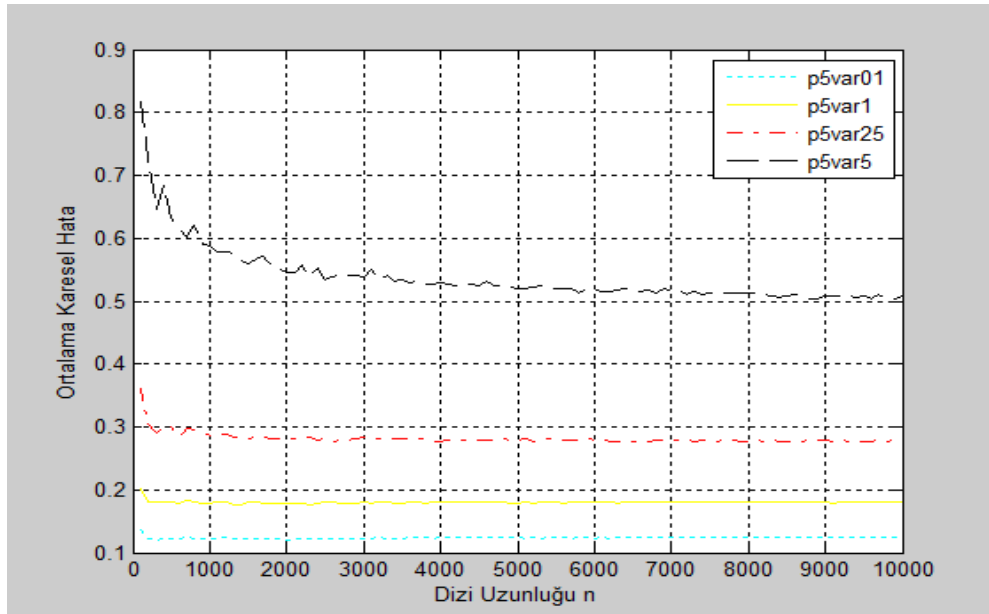


Şekil 5.8. Gürültülü Henon sinyali için Polinom Uydurma yöntemi ile tasarlanan süzgecin farklı değışinti değeri altındaki Ortalama Karese Hata değeri gösterimi

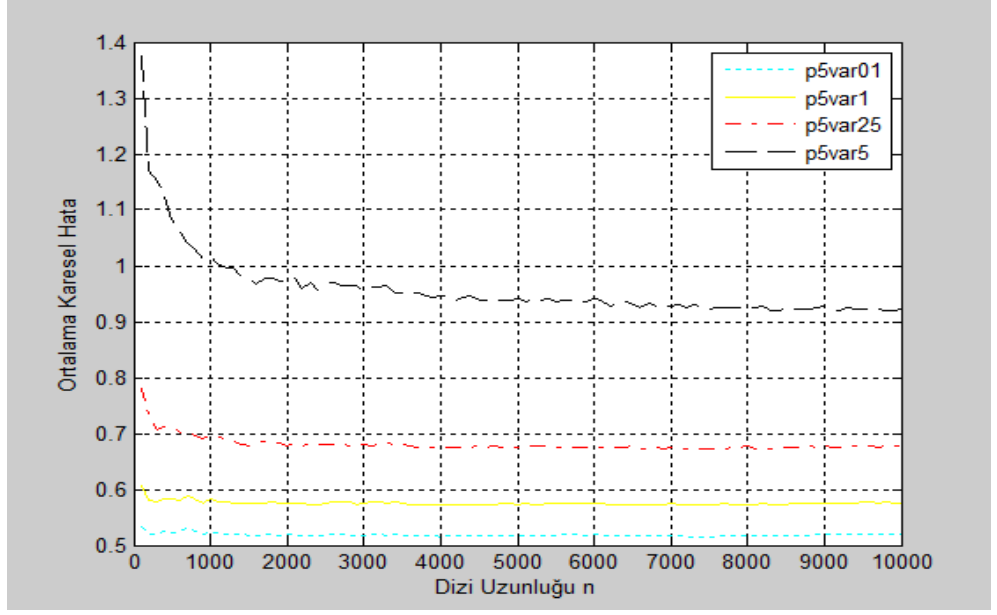
Şekil 5.7 ve Şekil 5.8’ de görüldüğü Polinom Uydurma yöntemi en iyi performansını değişinti değeri=1 (var1) olduğunda, en kötü performansını ise değişinti değeri=0.1 (var01) olduğunda göstermiştir.

5.2.5 Dalgacık katsayısı sıkıştırma yöntemi simülasyonları

Dalgacık Katsayısı Sıkıştırma yöntemi üç farklı program ile gösterilmiştir. Programlardaki farklılık, değişinti değerinin farklı şekilde formüle edilmesinden ileri gelmektedir. Lojistik regresyon ve Henon gürültülü sinyallerini süzmek için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin eşik değeri $\lambda=\sigma$ olarak alındığında elde edilen MSE değerleri aşağıdaki grafiklerde gösterilmiştir.

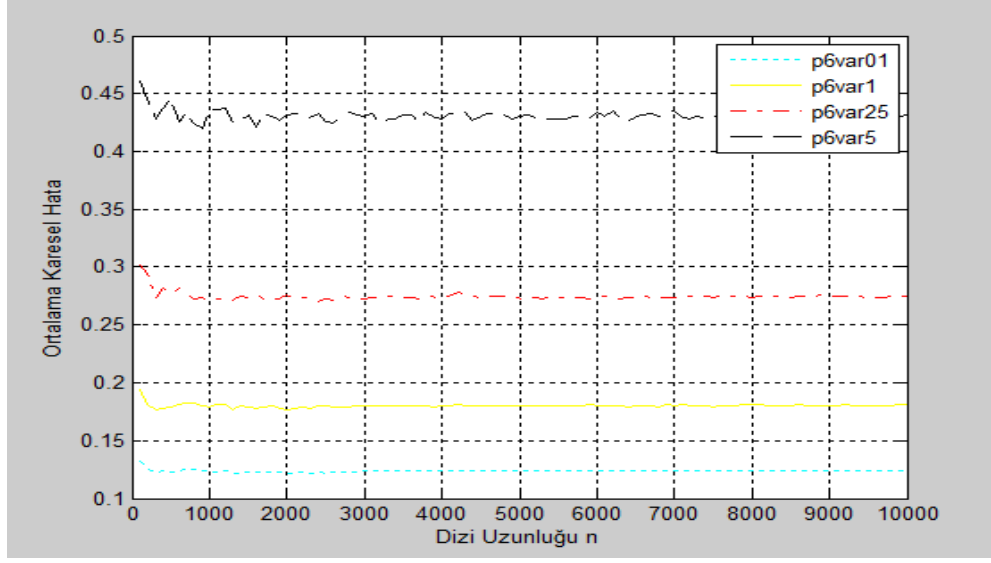


Şekil 5.9. Gürültülü Lojistik regresyon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda=\sigma$) farklı değişinti değerleri altındaki Ortalama Kareysel Hata değerleri gösterimi

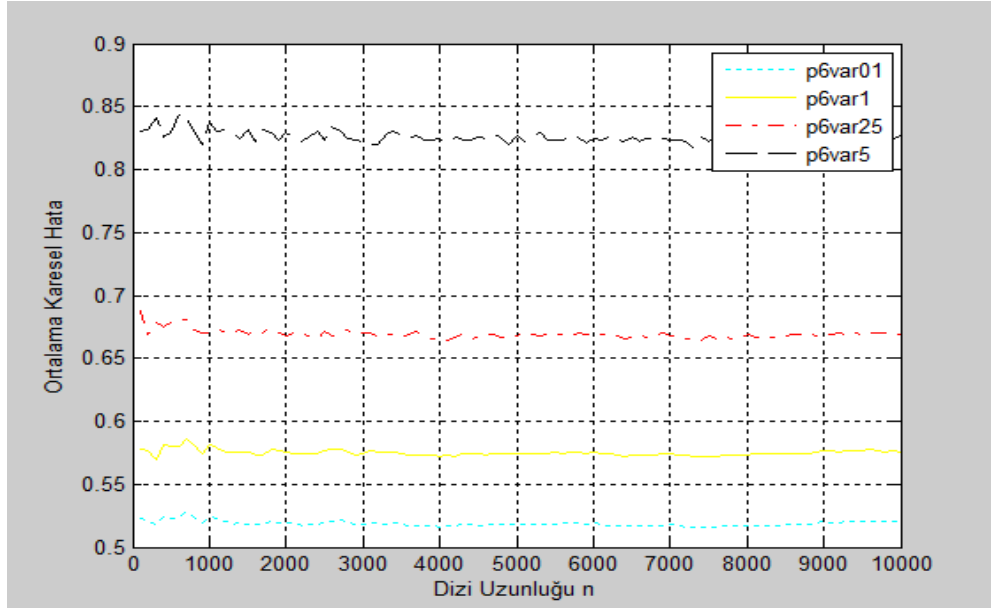


Şekil 5.10. Gürültülü Henon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda = \sigma$) farklı değışinti değeri altındaki Ortalama Karese Hata değeri gösterimi

Lojistik regresyon ve Henon gürültülü sinyallerini süzmek için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeç eşik değeri $\lambda_j^{[1]} = \sigma_j$ olarak alındığında elde edilen MSE değeri aşağıdaki grafiklerde gösterilmiştir.

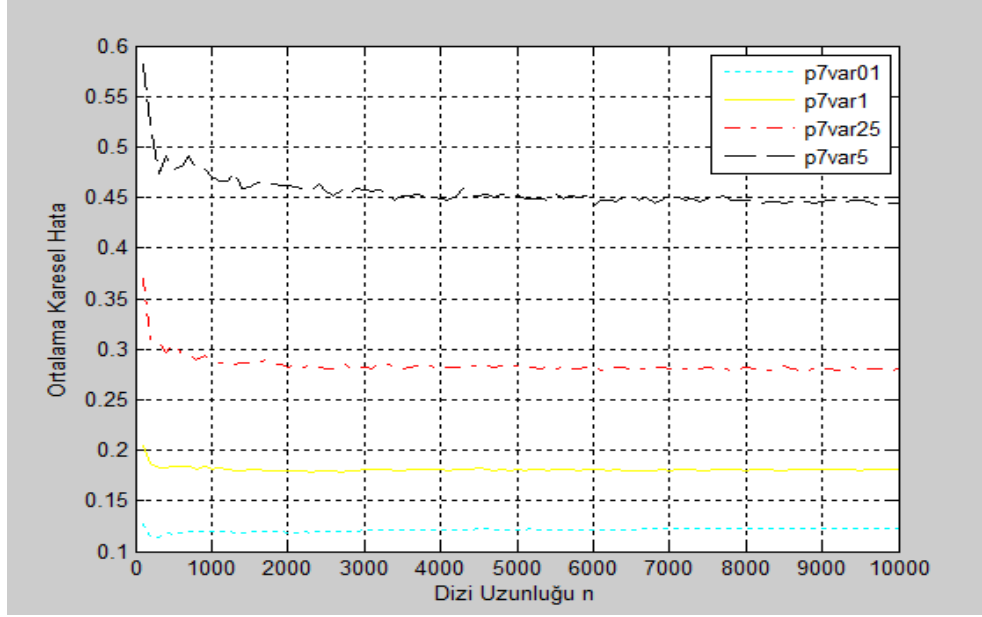


Şekil 5.11. Gürültülü Lojistik regresyon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda_j^{[1]} = \sigma_j$) farklı değışinti değeri altındaki Ortalama Kareysel Hata değeri gösterimi

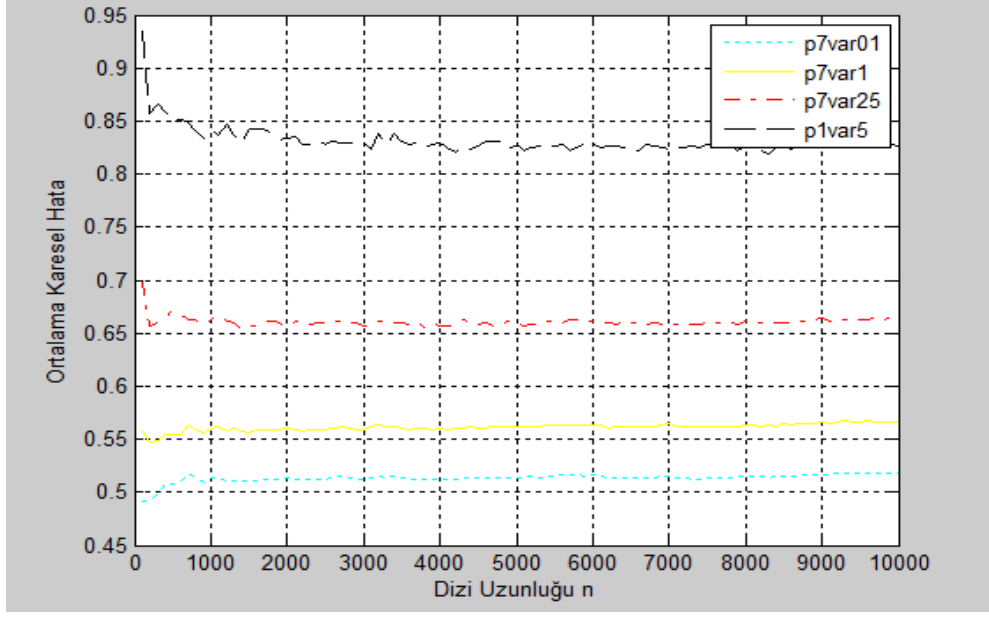


Şekil 5.12. Gürültülü Henon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda_j^{[1]} = \sigma_j$) farklı değışinti değeri altındaki Ortalama Kareysel Hata değeri gösterimi

Lojistik regresyon ve Henon gürültülü sinyallerini süzmek için Dalgacık Katsayısı Sıkıştırma Yöntemi ile tasarlanan süzgeç eşik değeri $\lambda_j^{[2]} = \sigma_j \sqrt{\beta}$ olarak alındığında elde edilen MSE değerleri aşağıdaki grafiklerde gösterilmiştir.



Şekil 5.13. Gürültülü Lojistik regresyon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda_j^{[2]} = \sigma_j \sqrt{\beta}$) farklı değışinti değeri altındaki Ortalama Karese Hata değeri gösterimi



Şekil 5.14. Gürültülü Henon sinyali için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgecin ($\lambda_j^{[2]} = \sigma_j \text{ ---} / \beta$) farklı değışinti değeri altındaki Ortalama Karese Hata değeri gösterimi

Şekil 5.9-5.14'de görüldüğü gibi sırasıyla eşik değeri $\lambda = \sigma \text{ ---}$, $\lambda_j^{[1]} = \sigma_j \text{ ---}$, $\lambda_j^{[2]} = \sigma_j \text{ ---} / \beta$ için Dalgacık Katsayısı Sıkıştırma yöntemi en iyi performansını değışinti değeri=1 (var1) olduğunda, en kötü performansını ise değışinti değeri=0.1 (var01) olduğunda göstermiştir.

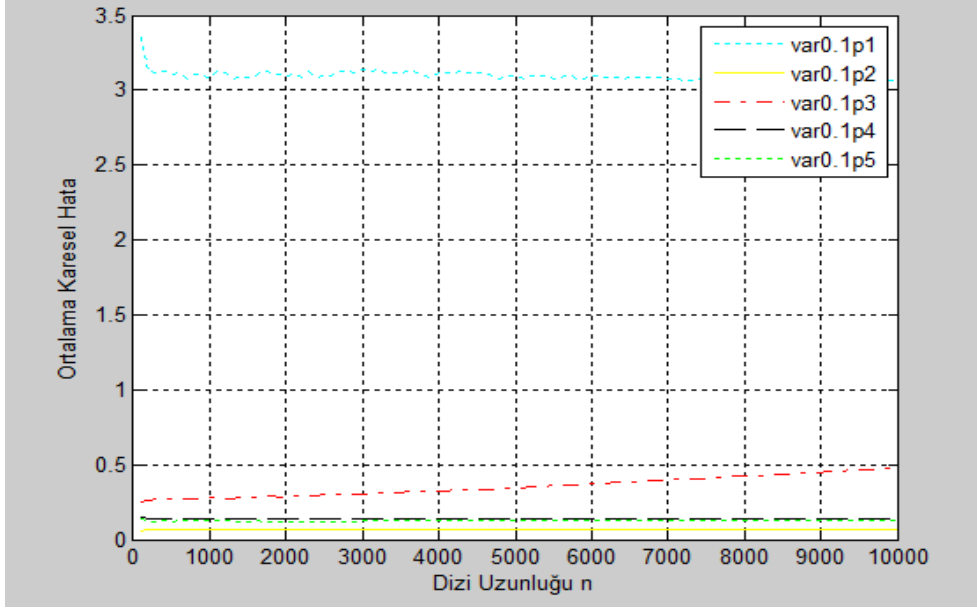
5.3 Farklı Süzgeçlerin Performanslarının Aynı Değışinti Değeri Kriteri Altında Karşılaştırılması

Bu bölümde 1-boyutlu Lojistik regresyon ve 2-boyutlu Henon gürültülü sinyalleri için değışinti değeri sabit tutularak farklı süzgeçlerin aynı değışinti değeriinde verdikleri tepki karşılaştırılacaktır. Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeçler değışik eşik değeri seçim yöntemleri altında birbirlerine yakın MSE değeri verdiğinden dolayı bu yöntem için tek eşik (λ) değeri alınarak beş süzgeç programı ile simulasyon yapılmıştır. Karşılaştırma için

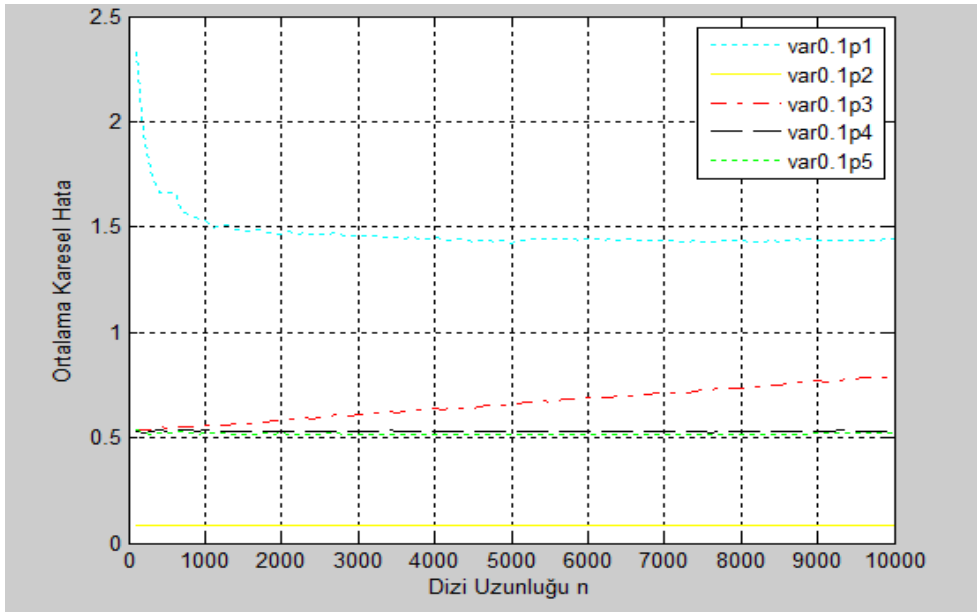
kullanılan beş süzgeç programı sırasıyla Evrensel FIR MMSE süzgeç, Wiener süzgeci, LMS-benzeri süzgeç, Polinom Uydurma yöntemi ve Dalgacık Katsayısı Sıkıştırma yöntemi ($\lambda = \sigma$) ile tasarlanan süzgeçlerdir (süzgeç programları sırası ile p1,p2...p5 olarak adlandırılmıştır). Süzgeçlerin performanslarında dört adet değişinti değeri kullanılmıştır. Değişinti değerleri olarak sırasıyla 0.1,1,2.5 ve 5 değerleri alınmış ve var01, var1,var25,var5 olarak gösterilmiştir. Ortalama Karesel Hata (MSE) değerleri sinyal uzunluğuna göre çizdirilirken herbir MSE değeri, o sinyal uzunluğu için yapılan 100 ayrı simulasyon sonucunun aritmetik ortalaması olarak hesaplanmıştır.

5.3.1 Değişinti değeri=0.1 (var01) için farklı süzgeçlerin performanslarının karşılaştırılması

Lojistik regresyon ve Henon gürültülü sinyali için değişinti değeri=0.1 (var01) olarak alınarak beş tane süzgeç programının simulasyonu yapılmış ve Ortalama Karesel Hata (MSE) değerleri sinyal uzunluğuna karşı aşağıdaki şekillerde çizdirilmiştir.



Şekil 5.15 Gürültülü Lojistik regresyon sinyali için değışinti değeri=0.1 (var01) altında süzgeç programlarının Ortalama Karese Hata değeri gösterimi



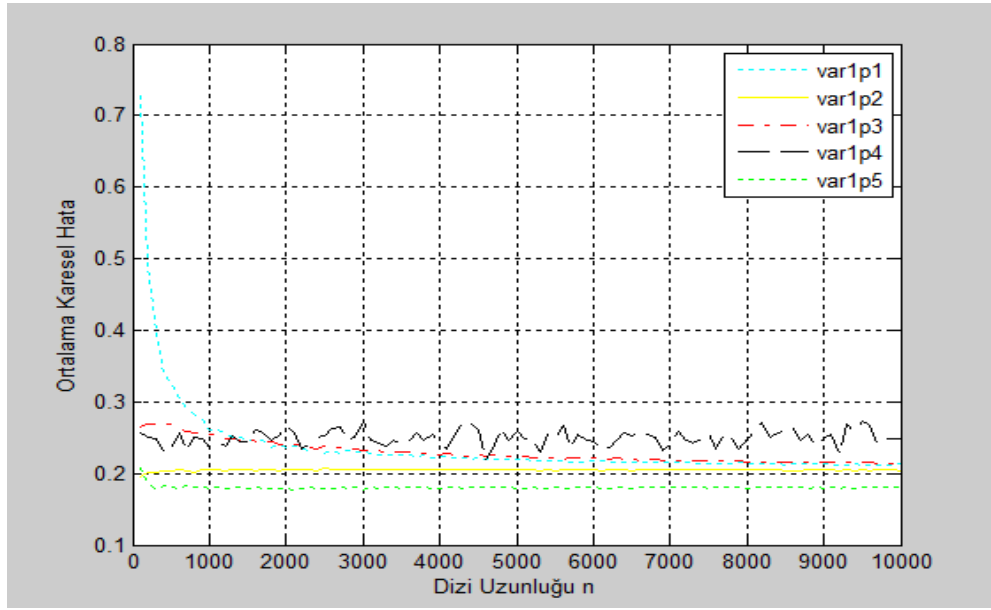
Şekil 5.16. Gürültülü Henon sinyali için değışinti değeri=0.1 (var01) altında süzgeç programlarının Ortalama Karese Hata değeri gösterimi

Şekil 5.15' ve Şekil 5.16'ya göre gürültülü Lojistik regresyon ve Henon sinyalleri için değışinti değeri =0.1 olarak alındığında en iyi süzgeç performansını Wiener süzgeci göstermektedir. Bu süzgecin performansına en fazla yaklaşan

süzgeç Dalgacık Katsayısı yöntemi ile tasarlanan süzgeçtir. Daha sonra ise performans yönünden Polinom Uydurma yöntemi ile tasarlanan süzgeç, LMS-benzeri süzgeç ve Evrensel FIR MMSE süzgeç gelmektedir.

5.3.2 Değişinti değeri=1 (var1) için farklı süzgeçlerin performanslarının karşılaştırılması

Lojistik regresyon gürültülü sinyali için değişinti değeri=1 (var1) olarak alınarak beş tane süzgeç programının simülasyonu yapılmış ve aşağıda gösterilmiştir.

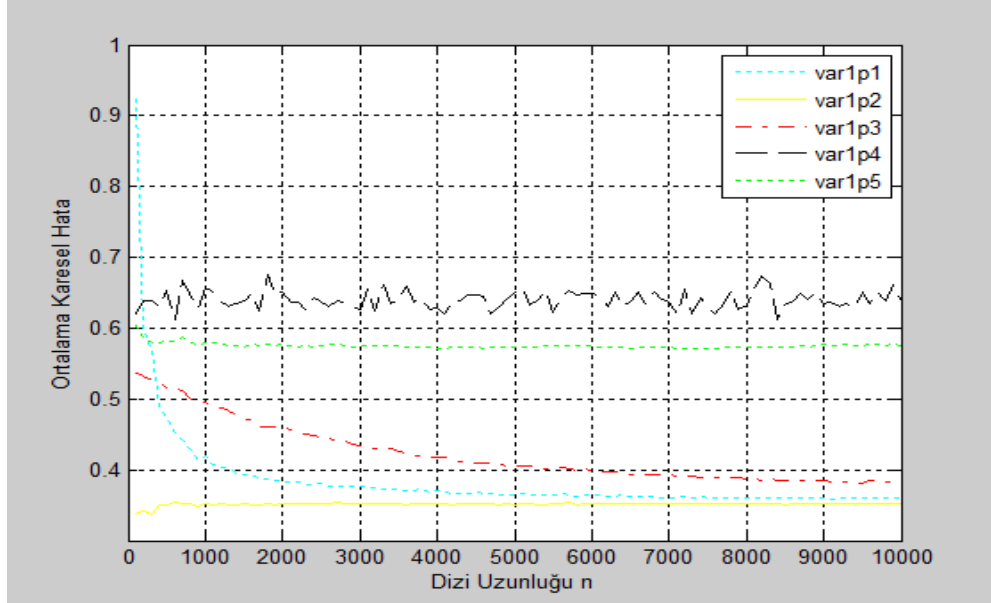


Şekil 5.17. Gürültülü Lojistik regresyon sinyali için değişinti değeri=1 (var1) altında süzgeç programlarının Ortalama Kareysel Hata değerleri gösterimi

Yukarıdaki şekilde görüldüğü gibi Lojistik regresyon sinyali için değişinti değeri =1 olarak alındığında en iyi süzgeç performansını Dalgacık Katsayısı yöntemi ile tasarlanan süzgeç göstermektedir. En iyi ikinci performansı Wiener süzgeci göstermiştir. Daha sonra ise performans yönünden Evrensel FIR MMSE

süzgeç, LMS-benzeri süzgeç, Polinom Uydurma yöntemi ile tasarlanan süzgeç gelmektedir.

Henon gürültülü sinyali için değışinti değeri=1 (var1) olarak alınarak beş tane süzgeç programının simülasyonu yapılmış ve aşağıda gösterilmiştir.

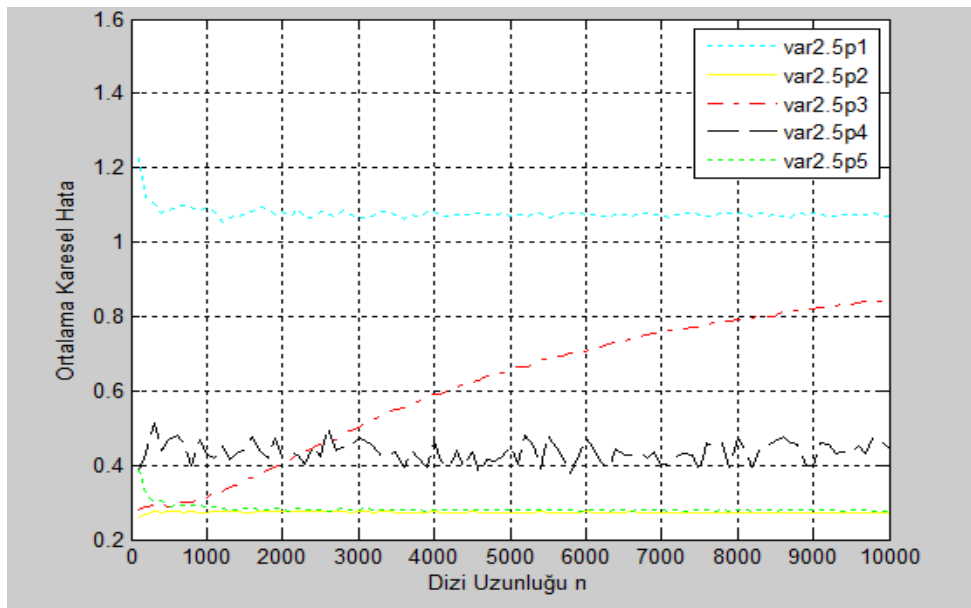


Şekil 5.18. Gürültülü Henon sinyali için değışinti değeri=1 (var1) altında süzgeç programlarının Ortalama Kareysel Hata değerleri gösterimi

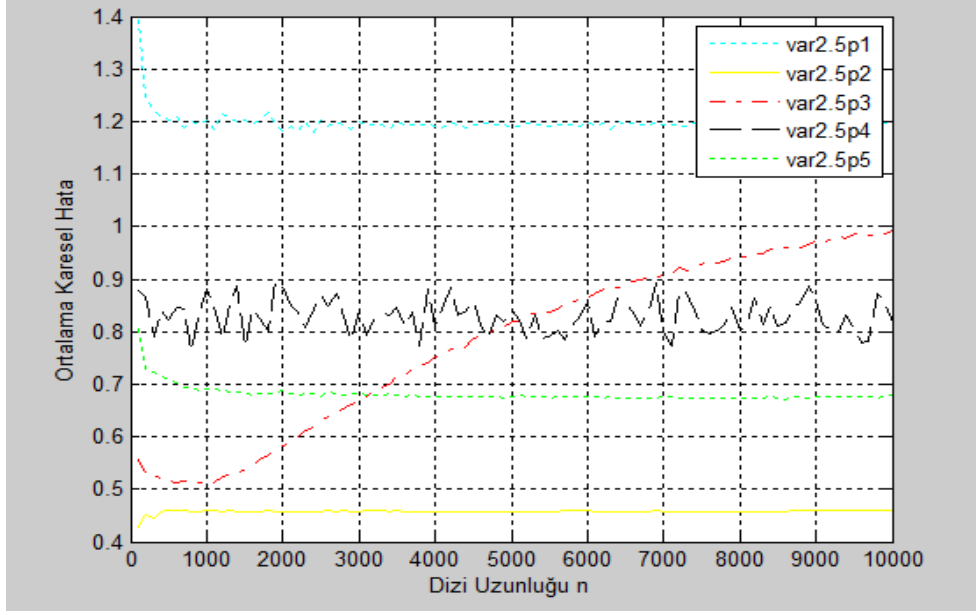
Şekil 5.18’de görüldüğü gibi gürültülü Henon sinyali için değışinti değeri =1 olarak alındığında en iyi süzgeç performansını Wiener süzgeci göstermiştir. Bu süzgecin performansını Evrensel süzgeç, LMS-benzeri süzgeç izlemiştir. En kötü iki performansı gösteren süzgeçler ise Dalgacık Katsayısı yöntemi ile tasarlanan süzgeç ve Polinom Uydurma yöntemi ile tasarlanan süzgeçtir.

5.3.3 Değişinti değeri=2.5 (var2.5) için farklı süzgeçlerin performanslarının karşılaştırılması

Lojistik regresyon gürültülü sinyali ve Henon gürültülü sinyali için değişinti değeri=2.5 (var2.5) olarak alınarak beş tane süzgeç programının simülasyonu yapılmış ve aşağıda gösterilmiştir.



Şekil 5.19. Gürültülü Lojistik regresyon sinyali için değişinti değeri=2.5 (var2.5) altında süzgeç programlarının Ortalama Kareysel Hata değerleri gösterimi

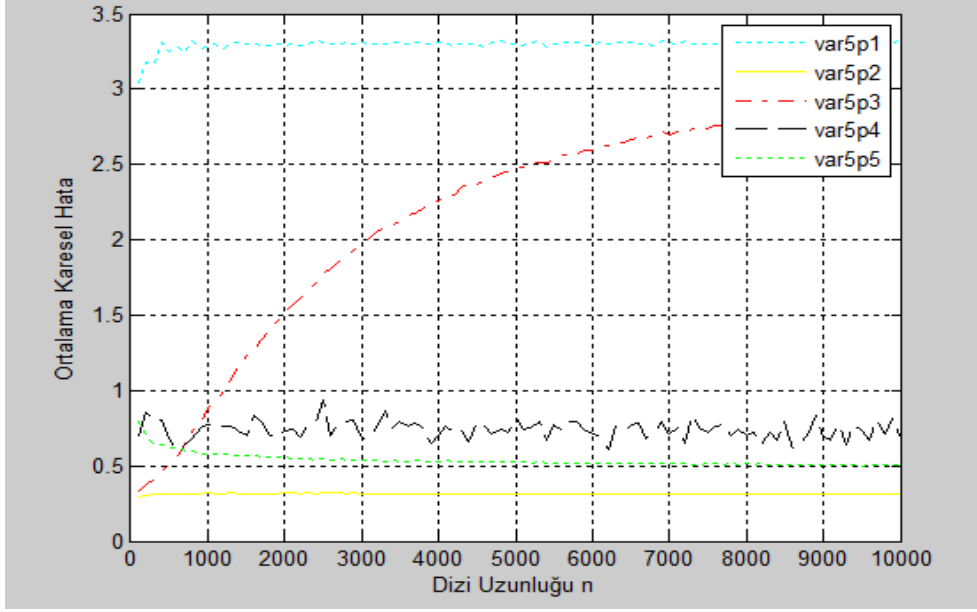


Şekil 5.20. Gürültülü Henon sinyali için değişinti değeri=2.5 (var2.5) altında süzgeç programlarının Ortalama Karesele Hata değerleri gösterimi

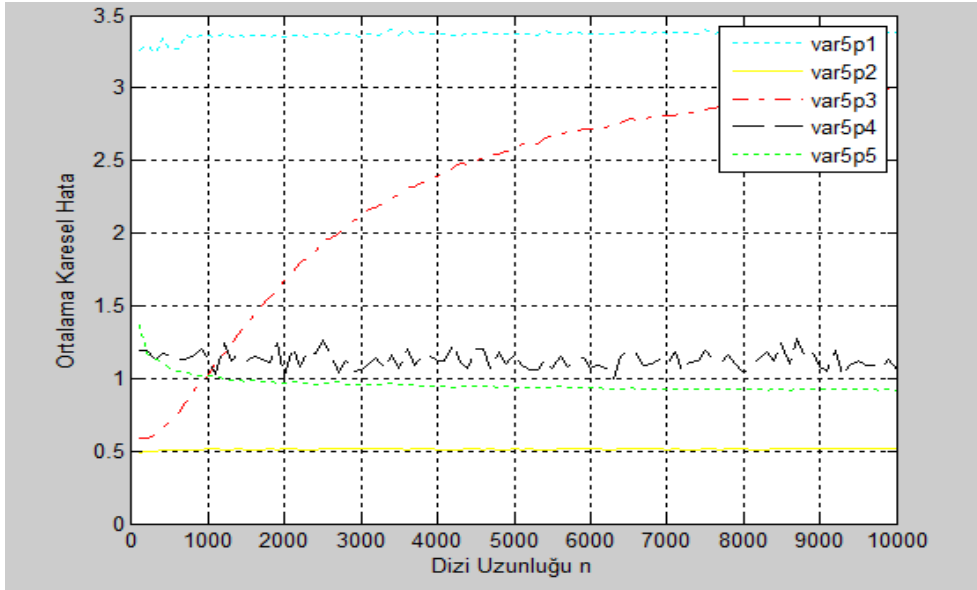
Şekil 5.19 ve şekil 5.20'ye bakıldığında Lojistik regresyon ve Henon gürültülü sinyalleri için değişinti değeri =2.5 olarak alındığında en iyi süzgeç performansını Wiener süzgecin gösterdiği, bu süzgecin performansını Dalgacık Katsayısı yöntemi ile tasarlanan süzgeç, Polinom Uydurma yöntemi ile tasarlanan süzgecin ve LMS-benzeri süzgecin izlediği, en kötü performansı ise Evrensel FIR MMSE süzgecin gösterdiği gözlenmiştir.

5.3.4 Değişinti değeri=5 (var5) için farklı süzgeçlerin performanslarının karşılaştırılması

Lojistik regresyon ve Henon gürültülü sinyalleri için değişinti değeri=5 (var5) olarak alınarak beş tane süzgeç programının simülasyonu yapılmış ve aşağıda gösterilmiştir.



Şekil 5.21. Gürültülü Lojistik regresyon sinyali için değişinti değeri=5 (var5) altında süzgeç programlarının Ortalama Karesel Hata değerleri gösterimi



Şekil 5.22. Gürültülü Henon sinyali için değişinti değeri=5 (var5) altında süzgeç programlarının Ortalama Karesel Hata değerleri gösterimi

Şekil 5.21 ve şekil 5.22'ye bakıldığında Lojistik regresyon ve Henon gürültülü sinyalleri için değişinti değeri =5 olarak alındığında en iyi süzgeç






performansını Wiener süzgecin gösterdiği, bu süzgecin performansını Dalgacık Katsayısı yöntemi ile tasarlanan süzgeç, Polinom Uydurma yöntemi ile tasarlanan süzgecin ve LMS-benzeri süzgecin izlediği, en kötü performansı ise Evrensel FIR MMSE süzgecin gösterdiği gözlenmiştir.

6. KORELASYON BOYUTU

Fraktal geometri, doğadaki karmaşık biçimlerin daha iyi anlaşılmasına yardımcı olan bir geometri dalıdır. Yaklaşık 2000 sene önce ortaya çıkan Öklit geometrisinde boyutların değeri tamsayıdır. Örneğin doğru bir boyutlu, kare, üçgen iki boyutlu ve küp 3 boyutludur. Öklit geometrisi doğal biçimleri açıklama konusunda yetersiz kalmaktadır. Bundan dolayı 20. yüzyılın sonlarına doğru fraktal geometri ortaya çıkmıştır.

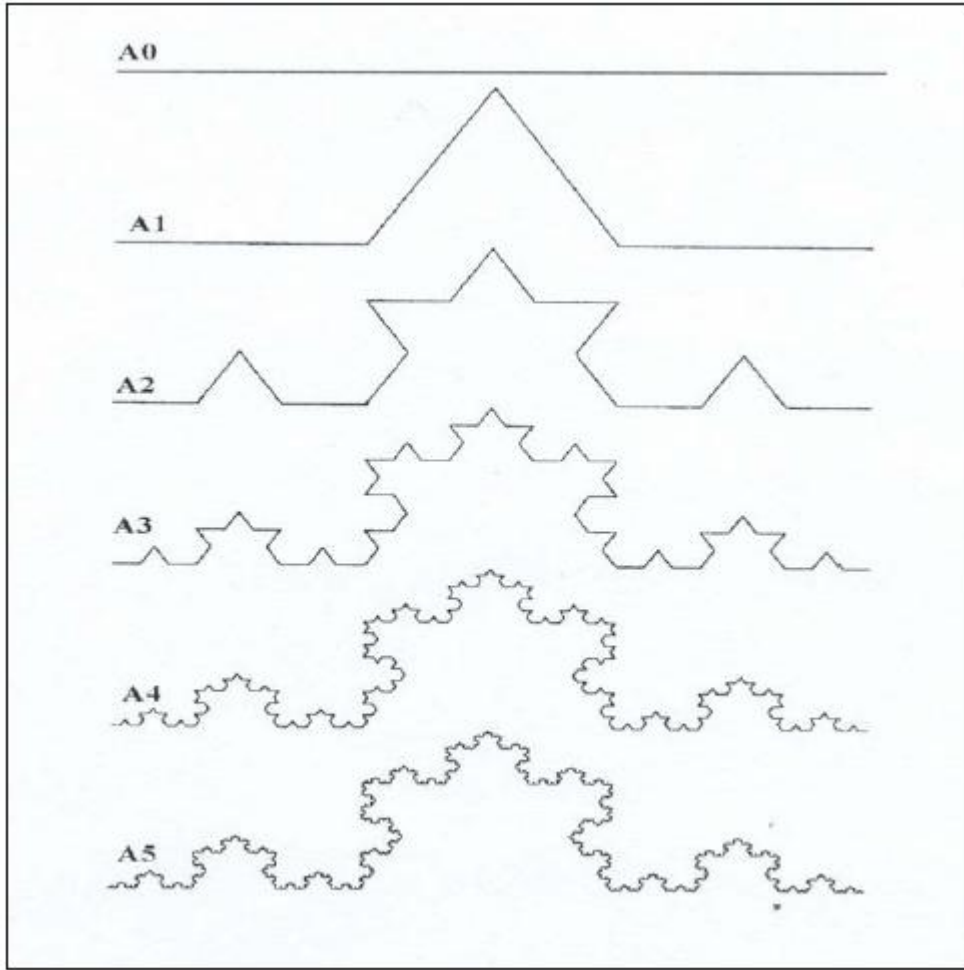
1970’li yıllardan itibaren Mandelbrot, bu konuda çalışmaya başlamıştır. “Siz hiç küre şeklinde bulut, koni şeklinde dağ gördünüz mü” diyen Mandelbrot’a göre doğayı anlamak için yeni bir geometriye ihtiyacımız vardı. Polonya kökenli olan matematikçi “Britanya kıyılarının uzunluğu nedir ?” sorusu üzerine kaos çalışmalarına fraktal boyutu eklemiştir. Latince fractus (kırık, şekilsiz) kelimesinden türettiği fraktal kavramını 1975’te anlatmış ve doğadaki cisimlerin fraktaller ile ölçülebileceğini göstermiştir [1].

Yukarıda da belirtildiği gibi Öklit geometrisinde boyutlar tamsayı değerleri ile ifade edilir. Fakat Fraktal boyutlar tam sayı değerleri almazlar, kesirli değerler alırlar. Aşağıdaki şekilde Öklit geometrisi ve fraktal geometrideki örnek cisimler için boyutlar gösterilmektedir.

EUKLİDYEN BOYUTLAR		FRAKTAL BOYUTLAR	
• (nokta)	0	-----	0.4
—	1		1.4
	2		1.8
	3		2.6

Şekil 6.1. Öklidyen ve fraktal boyutlar [16]

Fraktaller, basit matematiksel işlemlerin sürekli tekrar edilmesi yoluyla hesaplanırlar. En basit fraktallerden biri Koch kar tanesidir. Bir Koch eğrisi inşa etmek için uzunluğu 1 birim olan bir doğru parçası (A_0) ele alınır ve bu doğrunun ortadaki $1/3$ 'lük kısmı atılarak, yerine aynı uzunlukta 2 parça eklenir (A_1) [16]. Bu işlem yeni oluşan her kenar için sonsuza kadar tekrarlanırsa fraktal olan Koch kar tanesi elde edilir. Aşağıdaki şekilde de görülebilen Koch kar tanesi çarpıcı bir özelliğe sahiptir; alan sınırlı, uzunluk sonsuz.



Şekil 6.2. Koch Tanesi [16]

Verilen bir sürecin kaotik olup olmadığını belirlemede kapasite, bilgi, korelasyon vb. boyut değerleri kullanılmaktadır. Kullanılan çeşitli yöntemlerden

korelasyon boyutu hesaplanması, kolaylık nedeniyle tercih edilmektedir. Grassberger-Procaccia Algoritması (GPA), korelasyon integralinin hesaplanması için pratik bir yöntemdir ve korelasyon integralinin ölçekleme bölgesinin eğimi korelasyon boyutuna karşılık gelir [16].

Dinamik bir sistemin sahip olduğu çekeri temsil eden ayrıntılı boyut hesapları vardır ve bu ayrıntılı boyut hesaplarının tamamı D_q boyut spektrumu olarak adlandırılabilir. Boyut spektrumu denklemi şu şekilde verilir.

$$\text{---} \text{---} \text{---} \tag{6.1}$$

$N(r)$, kümeyi örtmesi gereken r büyüklüğündeki (r kenarlı hiperküpler) m boyutlu hücrelerin sayısıdır ve $-\text{---}$ küme içinde bulunan bir noktanın j hiperküpünde olma olasılığıdır. H , küme içindeki noktaların toplam sayısıdır ve H_j , j hiperküpündeki kümenin noktalarının sayısıdır [3].

Homojen çekere sahip kümeler için bütün boyutlar (D_q) eşittir. D_0 kutu sayma (box-counting) boyutu, D_1 bilgi boyutu ve D_2 ilinti boyutu olarak isimlendirilir. Küme boyutu hesaplamalarında iki yol kullanılabilir. Birinci yol kutu sayma yöntemidir. Kutu sayma yönteminin dezavantajı sadece düşük boyutlu kümeler için hesaplanabilmesidir. D_0 kutu sayma boyutu şu şekilde verilebilir [3].

$$\text{---} \tag{6.2}$$

D_1 bilgi boyutunun denklemi

$$\text{---} \tag{6.3}$$

şeklini alır. D_2 ilinti boyutu şöyle ifade edilir.

$$\text{---} \tag{6.4}$$

Burada $C(r)$ terimi ilinti toplamı olarak ifade edilir ve

$$\tag{6.5}$$

şeklinde tanımlanır. $C(r)$ terimi kümenin iki noktasının aynı hücre içinde olma olasılığıdır. Bu yöntem yavaş olduğundan dolayı ve yüksek boyutlu çekicilere uygulanamaması nedeniyle yerine daha hızlı bir algoritma olan Grassberger ve Procaccia Algoritması (GPA) önerilmiştir [3].

GPA'ya göre; bir kümenin iki noktasının (X_i, X_j ve $i \neq j$) aynı r boyutlu hücrede olma olasılığı, iki noktanın r 'ye eşit veya r 'den küçük bir uzaklıkla ayrılmış olma olasılığına yaklaşık olarak eşittir.

$$\frac{\text{---}}{\text{---}} \quad (6.6)$$

Denklemden

$$\text{---} \quad (6.7)$$

ifadesi Öklit farkı olarak hesap edilir ve ρ Heaviside adım fonksiyonudur. Heaviside adım fonksiyonu

$$(6.8)$$

şeklinde tanımlanır. Yani $\rho(f)$, f şartı sağlandığında 1, sağlanmadığında 0 değerini almaktadır. r 'nin değer aralıkları için $C(r)$ hesaplanır ve ilinti boyutu D_2 , $\log(C(r))$ 'nin $\log(r)$ 'ye karşılık çizdirilmesi ile oluşan eğrinin eğimi olarak hesaplanır.

Tez kapsamında (2.43)'de verilen gürültülü sinyale, süzmek için önceki bölümlerde Evrensel Süzgeç, LMS Benzeri Süzgeç, Wiener Süzgeç, Polinom Uydurma Yöntemi ve Dalgacık Katsayısı Sıkıştırma Yöntemi uygulanmıştır. Gürültüsüz sinyal ile gürültülü sinyale uygulanan süzgeçlerin çıktılarının korelasyon boyutları karşılaştırılmıştır ve çizelge 6.1 ve çizelge 6.2'de gösterilmiştir. Simulasyonlarda gürültü değışinti değeri 1 olarak alınmıştır.

Çizelge 6.1. Lojistik regresyon sinyali için gürültüsüz sinyal ile süzgeç çıktılarında elde edilen sinyallerin korelasyon boyutu karşılaştırması

Sinyal Çeşidi	Korelasyon Boyutu
Gürültüsüz Sinyal	0.9803
Evrensel Süzgeç Çıktısı	1,854
LMS Benzeri Süzgeç Çıktısı	1,8108
Wiener Süzgeç Çıktısı	1,7209
Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Çıktısı	0.1491
Dalgacık Katsayısı Sıkıştırma Yöntemi ile Tasarlanan Süzgeç Çıktısı ($\lambda=\sigma$)	1,5347
Dalgacık Katsayısı Sıkıştırma Yöntemi ile Tasarlanan Süzgeç Çıktısı ($\lambda_j^{[1]}=\sigma_j$)	1,5195
Dalgacık Katsayısı Sıkıştırma Yöntemi ile Tasarlanan Süzgeç Çıktısı ($\lambda_j^{[2]}=\sigma_j / \beta$)	1,5615

Yukarıdaki çizelgeye bakıldığında Lojistik regresyon sinyali için gürültüsüz sinyalin korelasyon boyutuna en yakın olarak Dalgacık Katsayısı Sıkıştırma Yöntemi çıktısının ($\lambda_j^{[1]}=\sigma_j$) korelasyon boyutu gösterilebilir.

Çizelge 6.2. Henon sinyali için gürültüsüz sinyal ile süzgeç çıktılarında elde edilen sinyallerin korelasyon boyutu karşılaştırması

Sinyal Çeşidi	Korelasyon Boyutu
Gürültüsüz Sinyal	1,3943
Evrensel Süzgeç Çıktısı	2,8695
LMS Benzeri Süzgeç Çıktısı	2,9986
Wiener Süzgeç Çıktısı	2,6435
Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Çıktısı	0.1491
Dalgacık Katsayısı Sıkıştırma Yöntemi ile Tasarlanan Süzgeç Çıktısı ($\lambda=\sigma$)	1,8748
Dalgacık Katsayısı Sıkıştırma Yöntemi ile Tasarlanan Süzgeç Çıktısı ($\lambda_j^{[1]}=\sigma_j$)	1,8761
Dalgacık Katsayısı Sıkıştırma Yöntemi ile Tasarlanan Süzgeç Çıktısı ($\lambda_j^{[2]}=\sigma_j / \beta$)	1,7488

Üstteki çizelgede Henon sinyali için gürültüsüz sinyalin korelasyon boyutuna en yakın olarak Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeç çıktısının ($\lambda_j^{[2]}=\sigma_j / \beta$) korelasyon boyutu gösterilebilir. Çıktılarının korelasyon boyutları gürültüsüz sinyalinkine daha yakın olan süzgeçler, süzgeçleme işlemi için daha uygun görülmektedir.

Gürültüsüz Lojistik regresyon sinyalinin korelasyon boyutunu hesaplayan program Ek-29'da gösterilmektedir. Gürültülü Lojistik regresyon sinyali için tasarlanan süzgeç çıktılarının korelasyon boyutlarını hesaplayan programlar Ek-30 - Ek-36'da verilmiştir. Lojistik regresyon sinyali için korelasyon boyutlarını karşılaştıran program Ek-37'de gösterilmiştir. Gürültüsüz Henon sinyalinin korelasyon boyutunu hesaplayan program Ek- 38'de gösterilmektedir. Gürültülü Henon sinyali için tasarlanan süzgeç çıktılarının korelasyon boyutlarını

hesaplayan programlar Ek-39 - Ek- 45'de gösterilmiştir. Henon sinyali için korelasyon boyutlarını karşılaştıran program Ek- 46'de verilmiştir

7. SONUÇLAR VE TARTIŞMA

Literatürde kaotik sinyallerin haberleşme sistemlerinde, yüksek frekanslı sinüsler yerine taşıyıcı sinyal olarak kullanılmasını öneren çalışmalar bulunmaktadır. Bilgi veya mesaj sinyalinin modüle ettiği bir kaotik sinyal haberleşme kanalından geçerken üstüne gürültü etki etmektedir. Bu kanaldan taşıyıcı sinyal yollanmadan önce, çıktısından sadece gürültü alınarak, bu gürültünün değişinti değeri hesaplanabilir. Hesaplanan değişinti değeri yardımıyla, bilgiyi taşıyan kaotik sinyalden tezdeki yöntemlerle gürültü temizlenebilir. Daha sonra sistemdeki alıcıda gürültüsü bastırılmış kaotik sinyal demodüle edilerek bilgi sinyali geri elde edilebilir. Eğer taşıyıcı sinyal olarak kaotik bir sinyal seçilirse, tezdeki yöntemlerin haberleşme sistemlerinde gürültü bastırmak için kullanılma potansiyeli bulunmaktadır.

Tez kapsamında 1-boyutlu Lojistik regresyon ve 2-boyutlu Henon kaotik sinyalleri üzerine Gauss tipi gürültü eklenip daha sonra elde edilen gürültülü sinyaller farklı süzgeçler kullanılarak süzölmüş ve süzgeçlerin performans karşılaştırması yapılmıştır. Gürültülü sinyalleri süzmek için sırasıyla Evrensel FIR MMSE süzgeç, Wiener süzgeci, LMS-benzeri süzgeç, Polinom Uydurma yöntemi ile tasarlanan süzgeç ve Dalgacık Katsayısı yöntemi ile tasarlanan süzgeç kullanılmıştır. Bu süzgeçler, Wiener süzgeci dışında, sinyal hakkında bilgi sahibi olmadan sadece gürültü hakkında belirli istatistiksel bilgiler (değişinti, standart sapma değerleri) kullanılarak tasarlanmıştır. Sadece Wiener süzgeci tasarlanırken sinyalin tümü hakkında bilgi sahibi olduğu varsayılmış ve bu süzgeç sadece gürültü hakkında belirli istatistiksel bilgilere sahip olduğu varsayılarak tasarlanan diğer süzgeçlerle performans karşılaştırması için kullanılmıştır.

Performans karşılaştırmasının ilk aşamasında gürültülü Lojistik regresyon ve Henon kaotik sinyalleri için önce herbir süzgecin farklı değişinti değerleri (var0.1, var1, var2.5, var2) altında gösterdikleri performans karşılaştırılmış, daha sonra ise değişinti değeri sabit tutularak farklı süzgeçlerin performansları karşılaştırılmıştır. Gürültülü Lojistik regresyon ve Henon sinyallerine uygulanan

süzgeçlerin farklı deęişinti deęerleri altında performanslarının karşılaştırması sırasında Evrensel FIR MMSE ve LMS-benzeri süzgeç dışındaki dięer tüm süzgeçler için deęişinti deęeri arttıkça Ortalama Karesel Hata'nın (MSE) arttığı gözlemlenmiştir (deęişinti deęeri=0.1 (var01) için en düşük MSE deęeri, deęişinti deęeri=5 (var5) için en yüksek MSE deęeri gözlemlenmiştir). Evrensel FIR MMSE ve LMS-benzeri süzgeçlerde deęişinti deęeri =1 (var1) için düşük MSE deęeri, yani en iyi performans elde edilmiştir.

Performans karşılaştırmasının ikinci aşamasında ise gürültülü Lojistik regresyon ve Henon kaotik sinyalleri için deęişinti deęeri sabit tutularak farklı süzgeç algoritmalarının performans karşılaştırması yapılmıştır. Lojistik regresyon gürültülü kaotik sinyali için tasarlanan süzgeçlerde deęişinti deęeri=1 (var1) dışındaki dięer tüm deęişinti deęerleri için en iyi performansları Wiener süzgeci göstermiştir. Bu deęişinti deęeri (var1) dışındaki dięer tüm deęişinti deęerleri için Wiener süzgecinin performanslarını, Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeç performansları izlemiştir. Deęişinti deęeri=1 (var1) için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeç, Wiener süzgeçten daha iyi performans göstermiştir. Gürültülü Henon kaotik sinyali için tasarlanan süzgeçlerde ise tüm deęişinti deęerleri için en iyi performansı sinyal hakkında bilgi sahibi olarak tasarlanan Wiener süzgeci göstermiştir. Wiener süzgecine en yakın performansı deęişinti deęeri=0.1, 2.5, 5 (var01,var25,var5) için Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeç göstermiştir. Deęişinti deęeri=1 için Wiener süzgecine en yakın performansı Evrensel FIR MMSE süzgeç göstermiştir.

Son olarak korelasyon boyutları incelenmiş ve gürültüsüz (asıl) kaotik sinyal ile süzgeç çıktılarında elde edilen sinyallerin korelasyon boyutlarının karşılaştırması yapılmıştır. Süzgeç çıktısında elde edilen sinyalin korelasyon boyutu ile gürültüsüz sinyalin korelasyon boyutu birbirine yakın olan süzgeçler, süzgeçleme işlemi için daha uygun olarak görülmektedir. Lojistik regresyon sinyali için gürültüsüz kaotik sinyalin korelasyon boyutuna en yakın olarak Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeç ($\lambda_j^{[1]} = \sigma_j$) çıktısının korelasyon boyutu gözlemlenmiştir. Henon gürültüsüz sinyalinin

korelasyon boyutuna en yakın olarak ise Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeç ($\lambda_j^{[2]} = \sigma_j / \beta$) çıktısının korelasyon boyutu gözlenmiştir.

Süzgeçler hızları ve hesaplama maliyetlerine göre incelenirse Evrensel FIR MMSE süzgecin tekrarlayan en küçük kareler tarzında olduğundan dolayı, LMS-benzeri süzgeçten daha fazla işlem gerektirdiği söylenebilir. Bu sebepten dolayı Evrensel FIR MMSE süzgeç, LMS-benzeri süzgece göre daha hızlı yakınsamaktadır. Yani sinyal uzunluğu arttıkça, LMS-benzeri süzgecin MSE değerine göre, Evrensel süzgecin MSE değeri kendi taban değerine daha hızlı düşmektedir. Wiener süzgeci sinyalin tümü hakkında bilgi sahibi olunarak tasarlandığından dolayı genelde performans karşılaştırmasında en iyi düşük MSE değerlerini vermektedir.

LMS-benzeri süzgecin basamak parametresi (μ , öğrenme hızı parametresi) büyük seçilirse, sinyal uzunluğu arttıkça, hata taban değeri civarına daha hızlı düştüğü, ama bu değer etrafında daha geniş salındığı gözlemlenmiştir. Daha küçük seçildiğinde, daha yavaş yakınsadığı, yani taban değeri civarına daha yavaş indiği, ama bu değer etrafında daha az salındığı gözlemlenmiştir. Bu parametre çok büyük seçildiğinde ise, LMS benzeri süzgecin yakınsamadığı, MSE değerinin sinyal uzunluğu arttıkça kararlı bir düşme göstermediği görülmüştür. Polinom Uydurma yöntemi ile tasarlanan süzgeçte, uydurulan polinom derecesi arttıkça MSE değerinin arttığı gözlemlenmiştir. Yine bu yöntem için segment uzunluğu $(2n+1)$ arttıkça MSE değerinin azaldığı gözlemlenmiştir.

Tasarlanan süzgeçlerin hızları incelendiğinde, Dalgacık Katsayısı Sıkıştırma yöntemi ile tasarlanan süzgeçler ilk sırayı almaktadır. Bu süzgeci Polinom Uydurma yöntemi ile tasarlanan süzgeç ve LMS-benzeri süzgeç izlemektedir. Son sıraları ise Wiener Süzgeci ve Evrensel FIR MMSE süzgeci almaktadır.

KAYNAKLAR

- [1] Canan S., *Kaos*, NTVBLM, 23, 23-27, 2011.
- [2] Weinberg V.G. ve ALEXPULOS A., *Examples of a class of chaotic radar signals*, Australian Government Department of Defence- Defence Science and Technology Organisation, DSN-TN-0660, Avusturalya, 2005.
- [3] Yılmaz D. ve Güler N.F. “*Kaotik zaman serisinin analizi üzerine bir araştırma*,” Gazi Üniversitesi Mühendis Mimarlık Fakültesi Dergisi, 21, 759-779, 2006.
- [4] Koçak, K., *Kaotik davranış kriteri olarak fraktal boyut değişimi ve dinamik sistemlere uygulanması*, Doktora Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 1996.
- [5] Rauf, F. ve Hassan, M.A., “Adaptive signal processing techniques for chaotic systems,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 3, 563 – 566, 1993.
- [6] Sevim, O., *Gürültü azaltmada LMS adaptif süzgeçlerin FPGA kullanarak uygulanması*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2007.
- [7] Feldbauer, C., Pernkopf, F. ve Rank, E., *Adaptive filters- A tutorial for the course computational Intelligence*, Signal Processing and Speech Communication Laboratory, 2010.
- [8] Farhang-Boroujeny, B., *Adaptive filters Theory and Application*, John Wiley and Sons, Chichester, İngiltere, 1998.
- [9] Haykin S., *Adaptive filter theory*, Prentice Hall Information and Science Series, NJ, ABD, 1996.
- [10] Moon, T. ve Weissman, T., “Universal FIR MMSE filtering,” *IEEE Transactions on Signal Processing*, 57, 1068 – 1083, 2009.

- [11] Gao, J., Sultan, H., Hu, J. ve Tung, W., “Denoising nonlinear time series by adaptive filtering and wavelet shrinkage: a comparison,” *IEEE Signal Processing Letters*, 19, 237-240, 2010.
- [12] Fidan, H., *Dalgacık dönüşümü tekniğiyle motor arıza tespit*, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta, 2006.
- [13] Açoğlu, E. A., *Sayısal süzgeç tasarımı ve uygulamaları*, Yüksek Lisans Tezi, Başkent Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2008.
- [14] Vatansever, F., Uysal, F. ve Uzun, A., “*Ayrık dalgacık dönüşümü ile gürültü süzme*,” International Conference on Electrical and Electronics Engineering, Bursa, 241-244, 2002.
- [15] Fidan S., *Dalgacık kılavuzunda yayılan elektromanyetik dalganın dalgacık dönüşümü ile modellenmesi*, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara, 2006.
- [16] Yalçın, G. Ç., *Kaotik fiziksel sistemler ve acayip çekiciler*, Yüksek Lisans Tezi, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul, 2005.

Ek-1 Gürültülü Lojistik Regresyon Sinyalini Süzme için Tasarlanan Evrensel FIR MMSE Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken)

```
function ms = torp(N,var)
x=zeros(1,N);
d=5;
h=0;
ms=0;
x(1)=0.1;
w=zeros(d,1);
c=zeros(d,1);
c(1)=1;
crp=zeros(d,d);
sm=zeros(d,1);

for n=2:N
    x(n)=4*x(n-1)*(1-x(n-1));
end

xn=x+sqrt(var)*randn(1,N);
Y=zeros(d,1);Y(1)=xn(1);
h(1)=w.'*Y;

for n=1:N-1
    crp=crp+Y*Y.';
    sm=sm+Y(1)*Y-c;
    A=eye(d,d)+crp;
    w=inv(A)*sm;

    for i=1:d-1
        Y(d+1-i)=Y(d+1-i-1);
    end
```

```
Y(1)=xn(n+1);  
h(n+1)=w.'*Y;  
ms=((1/N)*(x(n)-h(n))^2)+ms;  
end
```

```
ms;  
end
```

**Ek-2 Gürültülü Lojistik Regresyon Sinyalini Süzme için Tasarlanan Wiener
Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken)**

```
function ms2 = torp2(N,var)
d=5;
top1=zeros(d,d);
top2=0;
x=zeros(1,N);
x(1)=0.1;
m=0;
ms2=0;
u=0;

for n=2:N
    x(n)=4*x(n-1)*(1-x(n-1));
end

xn=x+sqrt(var)*randn(1,N);
Y=zeros(d,1);
Y(1)=xn(1);

for n=1:N-1
    top1=top1+Y*Y.';
    top2=top2+x(n)*Y;

    for i=1:d-1
        Y(d+1-i)=Y(d+1-i-1);
    end

    Y(1)=xn(n+1);
end
```

```

u=inv((1/N)*top1)*((1/N)*top2);
Y=zeros(d,1);
Y(1)=xn(1);

for n=1:N-1
    m(n)=u.*Y;

    for i=1:d-1
        Y(d+1-i)=Y(d+1-i-1);
    end

    Y(1)=xn(n+1);
end

for n=1:N-1
    ms2=ms2+1/N*((x(n)-m(n)))^2;

end

ms2;

end

```


Ek-3 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan LMS- Benzeri Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken)

```
function ms3 = torp3(N,var)
d=5;
x=zeros(1,N);
x(1)=0.1;
r=0.0001;
w=0.1*rand(d,1);
c=zeros(d,1);
c(1)=1;
ms3=0;
z=0;
    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
xn=x+sqrt(var)*randn(1,N);
Y=zeros(d,1);
Y(1)=xn(1);
z(1)=w.'*Y;
    for n=1:N-1
        w=w-2*r*Y*(Y'*w)+2*r*(Y(1)*Y-c);
        for i=1:d-1
            Y(d+1-i)=Y(d+1-i-1);
        end
        Y(1)=xn(n+1);
        z(n+1)=w.'*Y;
        ms3=((1/N)*(x(n)-z(n))^2)+ms3;
    end
ms3;
end
```

Ek-4 Gürültülü Lojistik Regresyon Sinyalini Süzme için Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken)

```
function ms4 = torp4(N,var)
d=5;
x=zeros(1,N);
t=0;
x(1)=0.1;
e=16;
y=zeros(1,N);
a=zeros(4,1);
A=zeros(2*e+1,4);
u=ones(2*e+1,1);
f=zeros(2*e+1,1);
g=zeros(2*e+1,1);
j=zeros(2*e+1,1);
s=zeros(2*e+1,1);
segment=0;
y_ext=0;
xn_ext=0;
m=floor(N/e);
y_ext=[zeros(1,N),zeros(1,(m-1)*e+2*e+1-N)];
r=0;
ms4=0;
C=0;

    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end

xn=x+sqrt(var)*randn(1,N);
xn_ext=[xn,zeros(1,(m-1)*e+2*e+1-N)];
```

```

for k=1:2*e+1
    f(k)=k;
    g(k)=k^2;
    j(k)=k^3;
end

A=[u f g j];

    for r=0:floor(N/e)-1
        s=xn_ext(r*e+1:r*e+2*e+1);
        a=pinv(A)*s.';
        t(r*e+1:r*e+2*e+1)=(A*a).';
    end
y_ext(1:e)=t(1:e);

for r=0:floor(N/e)-1
    for l=1:(e+1)

y_ext(e*r+e+1)=(1-(l-1)/e)*t(e*r+(l+e))+((l-1)/e)*t(e*(r+1)+1);
y_ext(e*(r+1)+1:e*(r+2)+1)=y_ext(e*r+1:e*(r+1)+1);
    end
end
y=y_ext(1:N);

    for n=1:N-1
        ms4=ms4+1/N*((x(n)-y(n)))^2;
    end

ms4;
end

```

Ek-5 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda = \sigma \sqrt{\text{var}}$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken)

```
function ms5 = torp5(N,var)
d=5;
mse=0;
ms5=0;
a=0;
z=0;
x=zeros(1,N);
x(1)=0.1;
xn=zeros(1,N);
a=0;
z=0;
    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
xn=x+sqrt(var)*randn(1,N);
J=4;
[C,L]=wavedec(xn,J,'coif1');
    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end
lambda=1*sqrt(2*log(N));
C(z(1)+1:z(J+1))=sign(z(1)+1:z(J+1)).*max(0,abs(C(z(1)+1:z(J+1)))-lambda);
y=waverec(C,L,'coif1');
    for n=1:N-1
        ms5=ms5+1/N*((x(n)-y(n)))^2;
    end
ms5;
end
```

Ek-6 Gürültülü Lojistik Regresyon Sinyalini Süzme için Dalgacık Sıkıştırma Yöntemi ($\lambda_j^{[1]} = \sigma_j$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken)

```
function ms6 = torp6(N,var)
d=5;
a=0;
z=0;
ms6=0;
x=zeros(1,N);
x(1)=0.1;
xn=zeros(1,N);
a=0;
z=0;
    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end

xn=x+sqrt(var)*randn(1,N);
J=4;
[C,L]=wavedec(xn,J,'coif1');

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

vardet4=mean ((C(z(1)+1:z(2))-mean(C(z(1)+1:z(2))))).^2);

lambdet4 =sqrt(vardet4)*sqrt(2*log(N));
```

```

C(z(1)+1:z(2))=sign(C(z(1)+1:z(2))).*max(0,abs(C(z(1)+1:z(2)))-lambdet4);

vardet3=mean((C(z(2)+1:z(3))-mean(C(z(2)+1:z(3))))).^2);
lambdet3 =sqrt(vardet3)*sqrt(2*log(N));
C(z(2)+1:z(3))=sign(C(z(2)+1:z(3))).*max(0,abs(C(z(2)+1:z(3)))-lambdet3);

vardet2=mean((C(z(3)+1:z(4))-mean(C(z(3)+1:z(4))))).^2);
lambdet2 =sqrt(vardet2)*sqrt(2*log(N));
C(z(3)+1:z(4))=sign(C(z(3)+1:z(4))).*max(0,abs(C(z(3)+1:z(4)))-lambdet2);

vardet1=mean((C(z(4)+1:z(5))-mean(C(z(4)+1:z(5))))).^2);
lambdet1 =sqrt(vardet1)*sqrt(2*log(N));
C(z(4)+1:z(5))=sign(C(z(4)+1:z(5))).*max(0,abs(C(z(4)+1:z(5)))-lambdet1);

y=waverec(C,L,'coif1');

for n=1:N-1

ms6=ms6+1/N*((x(n)-y(n)))^2;

end

ms6;

end

```

Ek-7 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda_j^{[2]} = \sigma_j$ / β) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken)

```
function ms7 = torp7(N,var)
d=5;
a=0;
z=0;
ms7=0;
x=zeros(1,N);
x(1)=0.1;
xn=zeros(1,N);
a=0;
z=0;

    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end

xn=x+sqrt(var)*randn(1,N);
J=4;
[C,L]=wavedec(xn,J,'coif1');

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

vardet4=mean ((C(z(1)+1:z(2))-mean(C(z(1)+1:z(2))))).^2);
lambdet4 =sqrt(vardet4)*sqrt(2*log(N));
C(z(1)+1:z(2))=sign(C(z(1)+1:z(2))).*max(0,abs(C(z(1)+1:z(2)))-lambdet4);
```

```

vardet3=mean((C(z(2)+1:z(3))-mean(C(z(2)+1:z(3))))).^2);
lambdet3 =sqrt(vardet3)*sqrt(2*log(N))/log(3);
C(z(2)+1:z(3))=sign(C(z(2)+1:z(3))).*max(0,abs(C(z(2)+1:z(3)))- lambdet3);

```

```

vardet2=mean((C(z(3)+1:z(4))-mean(C(z(3)+1:z(4))))).^2);
lambdet2 =sqrt(vardet2)*sqrt(2*log(N))/log(4);
C(z(3)+1:z(4))=sign(C(z(3)+1:z(4))).*max(0,abs(C(z(3)+1:z(4)))-lambdet2);

```

```

vardet1=mean((C(z(4)+1:z(5))-mean(C(z(4)+1:z(5))))).^2);
lambdet1 =sqrt(vardet1)*sqrt(2*log(N))/sqrt(J);
C(z(4)+1:z(5))=sign(C(z(4)+1:z(5))).*max(0,abs(C(z(4)+1:z(5)))-lambdet1);

```

```

y=waverec(C,L,'coif1');

```

```

for n=1:N-1
    ms7=ms7+1/N*((x(n)-y(n)))^2
end

```

```

ms7;
end

```


Ek-8 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan Süzgeç Programlarını Toplu Olarak Çalıştıran Program (Süzgeç Programlarının Değişinti Değerlerini Değiştiren Program)

```
mse=0;
mseason=zeros(1,100);
p1var01=zeros(1,100);
p1var1=zeros(1,100);
p1var25=zeros(1,100);
p1var5=zeros(1,100);
yu=zeros(1,100);
l=0;
N=0;
y=0;
mseason=0;
    for nibr=1:4
        if nibr==1
            var=0.1
        elseif nibr==2
            p1var01=mseason
            mseason=0;
            var=1
        elseif nibr==3
            p1var1=mseason
            mseason=0;
            var=2.5
        elseif nibr==4
            p1var25=mseason
            mseason=0;
            var=5
        end
    end
```

```

for N=100:100:10000

    for rt=1:100
        mse=mse+ torp(N,var)/100; %Süzgeç programlar manuel olarak
        değiştirilmiş ve sabit değışinti değeri altında farklı süzgeç programları
        incelenmiştir.
    end

    mseson(N/100)=mse;
    mse=0;
end

end

yu=100:100:10000

p1var5=mseson
plot(yu,p1var01,'c:',yu,p1var1,'y-',yu,p1var25,'r-',yu,p1var5,'k--')
xlabel('Dizi Uzunluğu n')
ylabel('Ortalama Karesel Hata')
grid
legend('p1var01','p1var1','p1var25','p1var5',1)

```

Ek-9 Gürültülü Lojistik Regresyon Sinyalini Süzme için Tasarlanan Evrensel FIR MMSE Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması)

```
function ms = horp(N)
x=zeros(1,N);
d=5;
h=0;
ms=0;
x(1)=0.1;
w=zeros(d,1);
c=zeros(d,1);
c(1)=1;
crp=zeros(d,d);
sm=zeros(d,1);

    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
xn=x+sqrt(0.1)*randn(1,N);% );%değişinti değerleri manuel olarak sırasıyla 0.1,
1, 2.5, 5 olarak değiştirilmiştir

Y=zeros(d,1);Y(1)=xn(1);
h(1)=w.'*Y;

    for n=1:N-1
        crp=crp+Y*Y.';
        sm=sm+Y(1)*Y-c;
        A=eye(d,d)+crp;
        w=inv(A)*sm;

        for i=1:d-1
```

```
Y(d+1-i)=Y(d+1-i-1);  
end
```

```
Y(1)=xn(n+1);  
h(n+1)=w.'*Y;
```

```
ms=((1/N)*(x(n)-h(n))^2)+ms;
```

```
end
```

```
ms;
```

```
end
```

Ek-10 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan Wiener Süzgeci Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması)

```
function ms2 = horp2(N)
```

```
d=5;
```

```
top1=zeros(d,d);
```

```
top2=0;
```

```
x=zeros(1,N);
```

```
x(1)=0.1;
```

```
m=0;
```

```
ms2=0;
```

```
u=0;
```

```
for n=2:N
```

```
    x(n)=4*x(n-1)*(1-x(n-1));
```

```
end
```

xn=x+sqrt(0.1)*randn(1,N); %değişinti değerleri manuel olarak sırasıyla 0.1, 1, 2.5, 5 olarak değiştirilmiştir

```
Y=zeros(d,1);
```

```
Y(1)=xn(1);
```

```
for n=1:N-1
```

```
    top1=top1+Y*Y.;
```

```
    top2=top2+x(n)*Y;
```

```
for i=1:d-1
```

```
    Y(d+1-i)=Y(d+1-i-1);
```

```
end
```

```

        Y(1)=xn(n+1);
    end

    u=inv((1/N)*top1)*((1/N)*top2);
    Y=zeros(d,1);
    Y(1)=xn(1);

    for n=1:N-1
        m(n)=u.'*Y;

        for i=1:d-1
            Y(d+1-i)=Y(d+1-i-1);
        end

        Y(1)=xn(n+1);

    end

    for n=1:N-1
        ms2=ms2+1/N*((x(n)-m(n)))^2;
    end

    ms2;

end

```

Ek-11 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan LMS-Benzeri Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması)

```
function ms3 = horp3(N)

d=5;
x=zeros(1,N);
x(1)=0.1;
r=0.0001;
w=0.1*rand(d,1);
c=zeros(d,1);
c(1)=1;
ms3=0;
z=0;

    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end

xn=x+sqrt(0.1)*randn(1,N); %değişinti değerleri manuel olarak sırasıyla 0.1, 1,
2.5, 5 olarak değiştirilmiştir
Y=zeros(d,1);
Y(1)=xn(1);
z(1)=w.'*Y;

    for n=1:N-1

        for i=1:d-1
            Y(d+1-i)=Y(d+1-i-1);
        end

        w=w-2*r*Y*(Y'*w)+2*r*(Y(1)*Y-c);

    end
```

```
Y(1)=xn(n+1);  
z(n+1)=w.'*Y;  
ms3=((1/N)*(x(n)-z(n))^2)+ms3;  
end  
ms3;  
end
```


Ek-12 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Polinom Uydurma Yöntemi ile Tasarlanan Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması)

```
function ms4 = horp4(N)

d=5;
x=zeros(1,N);
t=0;
x(1)=0.1;
e=16;
y=zeros(1,N);
a=zeros(4,1);
A=zeros(2*e+1,4);
u=ones(2*e+1,1);
f=zeros(2*e+1,1);
g=zeros(2*e+1,1);
j=zeros(2*e+1,1);
s=zeros(2*e+1,1);
segment=0;
y_ext=0;
xn_ext=0;
m=floor(N/e);
y_ext=[zeros(1,N),zeros(1,(m-1)*e+2*e+1-N)];
r=0;
ms4=0;
C=0;

    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
```

xn=x+sqrt(0.1)*randn(1,N); %değişinti değerleri manuel olarak sırasıyla 0.1, 1,
2.5, 5 olarak değiştirilmiştir

```
xn_ext=[xn,zeros(1,(m-1)*e+2*e+1-N)];
```

```
for k=1:2*e+1
```

```
    f(k)=k;
```

```
    g(k)=k^2;
```

```
    j(k)=k^3;
```

```
end
```

```
A=[u f g j];
```

```
for r=0:floor(N/e)-1
```

```
    s=xn_ext(r*e+1:r*e+2*e+1);
```

```
    a=pinv(A)*s.;
```

```
    t(r*e+1:r*e+2*e+1)=(A*a).;
```

```
end
```

```
y_ext(1:e)=t(1:e);
```

```
for r=0:floor(N/e)-1
```

```
    for l=1:(e+1)
```

```
        y_ext(e*r+e+1)=(1-(l-1)/e)*t(e*r+(l+e))+((l-1)/e)*t(e*(r+1)+l);
```

```
        y_ext(e*(r+1)+1:e*(r+2)+1)=y_ext(e*r+1:e*(r+1)+1);
```

```
    end
```

```
end
```

```
y=y_ext(1:N);
```

```
for n=1:N-1
```

```
    ms4=ms4+1/N*((x(n)-y(n)))^2;
```

```
end
```

```
ms4;
```

```
end
```

Ek-13 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Dalgacık Sıkıştırma Yöntemi ($\lambda = \sigma$) ile Tasarlanan Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması)

```
function ms5 = horp5(N)

d=5;
mse=0;
ms5=0;
a=0;
z=0;
x=zeros(1,N);
x(1)=0.1;
xn=zeros(1,N);
a=0;
z=0;
    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
xn=x+sqrt(5)*randn(1,N);
J=4;
[C,L]=wavedec(xn,J,'coif1');

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

lambda=1*sqrt(2*log(N));
C(z(1)+1:z(J+1))=sign(z(1)+1:z(J+1)).*max(0,abs(C(z(1)+1:z(J+1)))-lambda);
y=waverec(C,L,'coif1');
```

```
for n=1:N-1
    ms5=ms5+1/N*((x(n)-y(n)))^2;
end
ms5;
end
```

Ek-14 Gürültülü Lojistik Regresyon Sinyalini Süzmek için Tasarlanan Süzgeç Programlarını Toplu Olarak Çalıştıran Program(Sabit Değişinti Değerleri Altında Süzgeç Programlarını Karşılaştıran Program)

```
mse1=0;
mse2=0;
mse3=0;
mse4=0;
mse5=0;
mseason1=zeros(1,100);
mseason2=zeros(1,100);
mseason3=zeros(1,100);
mseason4=zeros(1,100);
mseason5=zeros(1,100);
p1var01=zeros(1,100);
p1var1=zeros(1,100);
p1var25=zeros(1,100);
p1var5=zeros(1,100);
yu=zeros(1,100);
l=0;
N=0;
y=0;

for N=100:100:10000
    for rt=1:100
        mse1=mse1+ horp(N)/100;
        mse2=mse2+ horp2(N)/100;
        mse3=mse3+ horp3(N)/100;
        mse4=mse4+ horp4(N)/100;
        mse5=mse5+ horp5(N)/100;
    end
end
```

```
mseason1(N/100)=mse1;  
mseason2(N/100)=mse2;  
mseason3(N/100)=mse3;  
mseason4(N/100)=mse4;  
mseason5(N/100)=mse5;
```

```
mse1=0;  
mse2=0;  
mse3=0;  
mse4=0;  
mse5=0;
```

```
end
```

```
yu=100:100:10000;  
plot(yu,mseason1,'c:',yu,mseason2,'y-',yu,mseason3,'r-',yu,mseason4,'k--  
,yu,mseason5,'g:')  
xlabel('Dizi Uzunluğu n')  
ylabel('Ortalama Kareysel Hata')  
grid  
legend('var1p1','var1p2','var1p3','var1p4','var1p5',1)
```

**Ek-15 Gürültülü Henon Sinyalini Süzme için Tasarlanan Evrensel FIR
MMSE Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken)**

```
function ms = korp(N,var)

x=zeros(1,N);
d=5;
h=0;
ms=0;
x(1)=0;
x(2)=0;
w=zeros(d,1);

c=zeros(d,1);
c(1)=1;
crp=zeros(d,d);
sm=zeros(d,1);

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end

xn=x+sqrt(var)*randn(1,N);
Y=zeros(d,1);Y(1)=xn(1);
h(1)=w.'*Y;

    for n=1:N-1
        crp=crp+Y*Y.';
        sm=sm+Y(1)*Y-c;
        A=eye(d,d)+crp;
        w=inv(A)*sm;

        for i=1:d-1
```

```
Y(d+1-i)=Y(d+1-i-1);  
end
```

```
Y(1)=xn(n+1);  
h(n+1)=w.'*Y;  
ms=((1/N)*(x(n)-h(n))^2)+ms;
```

```
end
```

```
ms;
```

```
end
```


Ek-16 Gürültülü Henon Sinyalini Süzme için Tasarlanan Wiener Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken)

```
function ms2 = korp2(N,var)
d=5;
top1=zeros(d,d);
top2=0;
x=zeros(1,N);
x(1)=0;
x(2)=0;
m=0;
ms2=0;
u=0;

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end

xn=x+sqrt(var)*randn(1,N);
Y=zeros(d,1);
Y(1)=xn(1);

    for n=1:N-1
        top1=top1+Y*Y.';
        top2=top2+x(n)*Y;

            for i=1:d-1
                Y(d+1-i)=Y(d+1-i-1);
            end

        Y(1)=xn(n+1);
    end
```

```

u=inv((1/N)*top1)*((1/N)*top2);
Y=zeros(d,1);
Y(1)=xn(1);

    for n=1:N-1
        m(n)=u.*Y;

            for i=1:d-1
                Y(d+1-i)=Y(d+1-i-1);
            end

        Y(1)=xn(n+1);

    end

    for n=1:N-1
        ms2=ms2+1/N*((x(n)-m(n)))^2;
    end

ms2;

end

```

**Ek-17 Gürültülü Henon Sinyalini Süzme için Tasarlanan LMS Benzeri
Süzgeç Programı (Süzgeç sabit, değışinti değeri değışken)**

```
function ms3 = korp3(N,var)

d=5;
x=zeros(1,N);
x(1)=0;
x(2)=0;
r=0.0001;
w=0.1*rand(d,1);
c=zeros(d,1);
c(1)=1;
ms3=0;
z=0;

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end

xn=x+sqrt(var)*randn(1,N);
Y=zeros(d,1);
Y(1)=xn(1);
z(1)=w.'*Y;

    for n=1:N-1

        for i=1:d-1
            Y(d+1-i)=Y(d+1-i-1);
        end

        w=w-2*r*Y*(Y'*w)+2*r*(Y(1)*Y-c);

        Y(1)=xn(n+1);
        z(n+1)=w.'*Y;
        ms3=((1/N)*(x(n)-z(n))^2)+ms3;
    end
```

end

ms3;

end

**Ek-18 Gürültülü Henon Sinyalini Süzme için Polinom Uydurma Yöntemi ile
Tasarlanan Süzgeç Programı (Süzgeç sabit, değişinti değeri değişken)**

```
function ms4 = korp4(N,var)

d=5;
x=zeros(1,N);
t=0;
x(1)=0;
x(2)=0;
e=16;
y=zeros(1,N);
a=zeros(4,1);
A=zeros(2*e+1,4);
u=ones(2*e+1,1);
f=zeros(2*e+1,1);
g=zeros(2*e+1,1);
j=zeros(2*e+1,1);
s=zeros(2*e+1,1);
segment=0;
y_ext=0;
xn_ext=0;
m=floor(N/e);
y_ext=[zeros(1,N),zeros(1,(m-1)*e+2*e+1-N)];
r=0;
ms4=0;
C=0;

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end

xn=x+sqrt(var)*randn(1,N);
xn_ext=[xn,zeros(1,(m-1)*e+2*e+1-N)];
```

```

for k=1:2*e+1
    f(k)=k;
    g(k)=k^2;
    j(k)=k^3;
end

```

```
A=[u f g j];
```

```

for r=0:floor(N/e)-1
    s=xn_ext(r*e+1:r*e+2*e+1);
    a=pinv(A)*s.';
    t(r*e+1:r*e+2*e+1)=(A*a).';
end

```

```
y_ext(1:e)=t(1:e);
```

```

for r=0:floor(N/e)-1
    for l=1:(e+1)
        y_ext(e*r+e+1)=(1-(l-1)/e)*t(e*r+(l+e))+((l-1)/e)*t(e*(r+1)+1);
        y_ext(e*(r+1)+1:e*(r+2)+1)=y_ext(e*r+1:e*(r+1)+1);
    end
end

```

```
y=y_ext(1:N);
```

```

for n=1:N-1
    ms4=ms4+1/N*((x(n)-y(n)))^2;
end

```

```

ms4;
end

```

**Ek-19 Gürültülü Henon Sinyalini Süzme için Dalgacık Sıkıştırma Yöntemi
($\lambda = \sigma \sqrt{2 \log(N)}$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit, değışinti
değeri değışken)**

```
function ms5 = korp5(N,var)
d=5;
mse=0;
ms5=0;
a=0;
z=0;
x=zeros(1,N);
x(1)=0;
x(2)=0;
xn=zeros(1,N);
a=0;
z=0;
    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+sqrt(var)*randn(1,N);
J=4;
[C,L]=wavedec(xn,J,'coif1');

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

lambda=1*sqrt(2*log(N));
C(z(1)+1:z(J+1))=sign(z(1)+1:z(J+1)).*max(0,abs(C(z(1)+1:z(J+1)))-lambda);
y=waverec(C,L,'coif1');
```

```
for n=1:N-1
    ms5=ms5+1/N*((x(n)-y(n)))^2;
end
```

```
ms5;
end
```


**Ek-20 Gürültülü Henon Sinyalini Süzme için Dalgacık Sıkıştırma Yöntemi
($\lambda_j^{[1]} = \sigma_j$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit,
değişinti değeri değişken)**

```
function ms6 = korp6(N,var)
d=5;
a=0;
z=0;
ms6=0;
x=zeros(1,N);
x(1)=0;
x(2)=0;
xn=zeros(1,N);
a=0;
z=0;
    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end

xn=x+sqrt(var)*randn(1,N);
J=4;
[C,L]=wavedec(xn,J,'coif1');

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

vardet4=mean ((C(z(1)+1:z(2))-mean(C(z(1)+1:z(2))))).^2);
lambdet4 =sqrt(vardet4)*sqrt(2*log(N));
C(z(1)+1:z(2))=sign(C(z(1)+1:z(2))).*max(0,abs(C(z(1)+1:z(2)))-lambdet4);
```

```

vardet3=mean((C(z(2)+1:z(3))-mean(C(z(2)+1:z(3))))).^2);
lambdet3 =sqrt(vardet3)*sqrt(2*log(N));
C(z(2)+1:z(3))=sign(C(z(2)+1:z(3))).*max(0,abs(C(z(2)+1:z(3)))-lambdet3);

```

```

vardet2=mean((C(z(3)+1:z(4))-mean(C(z(3)+1:z(4))))).^2);
lambdet2 =sqrt(vardet2)*sqrt(2*log(N));
C(z(3)+1:z(4))=sign(C(z(3)+1:z(4))).*max(0,abs(C(z(3)+1:z(4)))-lambdet2);

```

```

vardet1=mean((C(z(4)+1:z(5))-mean(C(z(4)+1:z(5))))).^2);
lambdet1 =sqrt(vardet1)*sqrt(2*log(N));
C(z(4)+1:z(5))=sign(C(z(4)+1:z(5))).*max(0,abs(C(z(4)+1:z(5)))-lambdet1);

```

```

y=waverec(C,L,'coif1');

```

```

    for n=1:N-1
        ms6=ms6+1/N*((x(n)-y(n)))^2;
    end

```

```

ms6;
end

```

Ek-21 Gürültülü Henon Sinyalini Süzme için Dalgacık Sıkıştırma Yöntemi
($\lambda_j^{[2]} = \sigma_j \frac{\text{---}}{\beta}$) ile Tasarlanan Süzgeç Programı (Süzgeç sabit,
değişinti değeri değişken)

```
function ms7 = korp7(N,var)

d=5;
a=0;
z=0;
ms7=0;
x=zeros(1,N);
x(1)=0;
x(2)=0;
xn=zeros(1,N);
a=0;
z=0;

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end

xn=x+sqrt(var)*randn(1,N);
J=4;
[C,L]=wavedec(xn,J,'coif1');

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

vardet4=mean ((C(z(1)+1:z(2))-mean(C(z(1)+1:z(2))))).^2);
lambdet4 =sqrt(vardet4)*sqrt(2*log(N));
```

```

C(z(1)+1:z(2))=sign(C(z(1)+1:z(2))).*max(0,abs(C(z(1)+1:z(2)))-lambdet4);

vardet3=mean((C(z(2)+1:z(3))-mean(C(z(2)+1:z(3))))).^2;
lambdet3 =sqrt(vardet3)*sqrt(2*log(N))/log(3);
C(z(2)+1:z(3))=sign(C(z(2)+1:z(3))).*max(0,abs(C(z(2)+1:z(3)))-lambdet3);

vardet2=mean((C(z(3)+1:z(4))-mean(C(z(3)+1:z(4))))).^2;
lambdet2 =sqrt(vardet2)*sqrt(2*log(N))/log(4);
C(z(3)+1:z(4))=sign(C(z(3)+1:z(4))).*max(0,abs(C(z(3)+1:z(4)))-lambdet2);

vardet1=mean((C(z(4)+1:z(5))-mean(C(z(4)+1:z(5))))).^2;
lambdet1 =sqrt(vardet1)*sqrt(2*log(N))/sqrt(J);
C(z(4)+1:z(5))=sign(C(z(4)+1:z(5))).*max(0,abs(C(z(4)+1:z(5)))-lambdet1);

y=waverec(C,L,'coif1');

    for n=1:N-1
        ms7=ms7+1/N*((x(n)-y(n)))^2;
    end

ms7;

end

```

Ek-22 Gürültülü Henon Sinyalini Süzmek için Tasarlanan Süzgeç Programlarını Toplu Olarak Çalıştıran Program(Süzgeç Programlarının Değişinti Değerlerini Değiştiren Program)

```
mse=0;
mson=zeros(1,100);
p1var01=zeros(1,100);
p1var1=zeros(1,100);
p1var25=zeros(1,100);
p1var5=zeros(1,100);
yu=zeros(1,100);
l=0;
N=0;
y=0;
mson=0;
    for nmbr=1:4

        if nmbr==1
            var=0.1;

        elseif nmbr==2
            p1var01=mson
            mson=0;
            var=1;

        elseif nmbr==3
            p1var1=mson
            mson=0;
            var=2.5 ;

        elseif nmbr==4
            p1var25=mson
            mson=0;
```

```

var=5;

end

for N=100:100:10000
    for rt=1:100
        mse=mse+ korp(N,var)/100; %Süzgeç programlar manuel
olarak değiştirilmiş ve sabit değışinti değeri altında farklı süzgeç programları
incelenmiştir.
    end

    mseson(N/100)=mse;
    mse=0;
end

end

yu=100:100:10000

p1var5=mseson
plot(yu,p1var01,'c:',yu,p1var1,'y-',yu,p1var25,'r-',yu,p1var5,'k--')
xlabel('Dizi Uzunluğu n')
ylabel('Ortalama Karesel Hata')
grid
legend('p7var01','p7var1','p7var25','p7var5',1)

```

Ek-23 Gürültülü Henon Sinyalini Süzme için Tasarlanan Evrensel FIR MMSE Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması)

```
function ms = sorp(N)

x=zeros(1,N);
d=5;
h=0;
ms=0;
x(1)=0;
x(2)=0;
w=zeros(d,1);
c=zeros(d,1);
c(1)=1;
crp=zeros(d,d);
sm=zeros(d,1);

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+sqrt(0.1)*randn(1,N); %değişinti değerleri manuel olarak sırasıyla 0.1, 1,
2.5, 5 olarak değiştirilmiştir

Y=zeros(d,1);Y(1)=xn(1);
h(1)=w.'*Y;

    for n=1:N-1
        crp=crp+Y*Y.';
        sm=sm+Y(1)*Y-c;
        A=eye(d,d)+crp;
        w=inv(A)*sm;
```

```
        for i=1:d-1
            Y(d+1-i)=Y(d+1-i-1);
        end

        Y(1)=xn(n+1);
        h(n+1)=w.*Y;
        ms=((1/N)*(x(n)-h(n))^2)+ms;
    end

ms;

end
```


Ek-24 Gürültülü Henon Sinyalini Süzme için Tasarlanan Wiener Süzgeci Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması)

```
function ms2 = sorp2(N)
```

```
d=5;
```

```
top1=zeros(d,d);
```

```
top2=0;
```

```
x=zeros(1,N);
```

```
x(1)=0;
```

```
x(2)=0;
```

```
m=0;
```

```
ms2=0;
```

```
u=0;
```

```
for n=3:N
```

```
    x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
```

```
end
```

xn=x+sqrt(0.1)*randn(1,N); %değişinti değerleri manuel olarak sırasıyla 0.1, 1, 2.5, 5 olarak değiştirilmiştir

```
Y=zeros(d,1);
```

```
Y(1)=xn(1);
```

```
for n=1:N-1
```

```
    top1=top1+Y*Y.;
```

```
    top2=top2+x(n)*Y;
```

```
for i=1:d-1
```

```
    Y(d+1-i)=Y(d+1-i-1);
```

```
end
```

```
Y(1)=xn(n+1);
```

```
end
```

```
u=inv((1/N)*top1)*((1/N)*top2);
```

```
Y=zeros(d,1);
```

```
Y(1)=xn(1);
```

```
for n=1:N-1
```

```
    m(n)=u.'*Y;
```

```
        for i=1:d-1
```

```
            Y(d+1-i)=Y(d+1-i-1);
```

```
        end
```

```
Y(1)=xn(n+1);
```

```
end
```

```
for n=1:N-1
```

```
    ms2=ms2+1/N*((x(n)-m(n)))^2;
```

```
end
```

```
ms2;
```

```
end
```

**Ek-25 Gürültülü Henon Sinyalini Süzme için Tasarlanan LMS Benzeri
Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç
Programlarının Karşılaştırılması)**

```
function ms3 = sorp3(N)

d=5;
x=zeros(1,N);
x(1)=0;
x(2)=0;
r=0.0001;
w=0.1*rand(d,1);
c=zeros(d,1);
c(1)=1;
ms3=0;
z=0;

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end

xn=x+sqrt(0.1)*randn(1,N); %değişinti değerleri manuel olarak sırasıyla 0.1, 1,
2.5, 5 olarak değiştirilmiştir

Y=zeros(d,1);
Y(1)=xn(1);
z(1)=w.'*Y;

    for n=1:N-1
        w=w-2*r*Y*(Y'*w)+2*r*(Y(1)*Y-c);
        for i=1:d-1
            Y(d+1-i)=Y(d+1-i-1);
        end

        Y(1)=xn(n+1);
        z(n+1)=w.'*Y;
```

```
ms3=((1/N)*(x(n)-z(n))^2)+ms3;
```

```
end
```

```
ms3;
```

```
end
```

**Ek-26 Gürültülü Henon Sinyalini Süzme için Polinom Uydurma Yöntemi ile
Tasarlanan Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı
Süzgeç Programlarının Karşılaştırılması)**

```
function ms4 = sorp4(N)

d=5;
x=zeros(1,N);
t=0;
x(1)=0;
x(2)=0;
e=16;
y=zeros(1,N);
a=zeros(4,1);
A=zeros(2*e+1,4);
u=ones(2*e+1,1);
f=zeros(2*e+1,1);
g=zeros(2*e+1,1);
j=zeros(2*e+1,1);
s=zeros(2*e+1,1);
segment=0;
y_ext=0;
xn_ext=0;
m=floor(N/e);
y_ext=[zeros(1,N),zeros(1,(m-1)*e+2*e+1-N)];
r=0;
ms4=0;
C=0;

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
```

xn=x+sqrt(0.1)*randn(1,N); %değişinti değerleri manuel olarak sırasıyla 0.1, 1,
2.5, 5 olarak değiştirilmiştir

```
xn_ext=[xn,zeros(1,(m-1)*e+2*e+1-N)];
```

```
for k=1:2*e+1
```

```
    f(k)=k;
```

```
    g(k)=k^2;
```

```
    j(k)=k^3;
```

```
end
```

```
A=[u f g j];
```

```
for r=0:floor(N/e)-1
```

```
    s=xn_ext(r*e+1:r*e+2*e+1);
```

```
    a=pinv(A)*s.;
```

```
    t(r*e+1:r*e+2*e+1)=(A*a).;
```

```
end
```

```
y_ext(1:e)=t(1:e);
```

```
for r=0:floor(N/e)-1
```

```
    for l=1:(e+1)
```

```
        y_ext(e*r+e+1)=(1-(l-1)/e)*t(e*r+(l+e))+((l-1)/e)*t(e*(r+1)+l);
```

```
        y_ext(e*(r+1)+1:e*(r+2)+1)=y_ext(e*r+1:e*(r+1)+1);
```

```
    end
```

```
end
```

```
y=y_ext(1:N);
```

```
for n=1:N-1
```

```
    ms4=ms4+1/N*((x(n)-y(n)))^2;
```

```
end
```

```
ms4;
```

```
end
```

Ek-27 Gürültülü Henon Sinyalini Süzme için Tasarlanan Dalgacık Sıkıştırma Yöntemi ($\lambda=\sigma$) ile Tasarlanan Süzgeç Programı (Sabit Değişinti Değerleri Altında Farklı Süzgeç Programlarının Karşılaştırılması)

```
function ms5 = sorp5(N)

d=5;
mse=0;
ms5=0;
a=0;
z=0;
x=zeros(1,N);
x(1)=0;
x(2)=0;
xn=zeros(1,N);
a=0;
z=0;
    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+sqrt(0.1)*randn(1,N); %değişinti değerleri manuel olarak sırasıyla 0.1, 1,
2.5, 5 olarak değiştirilmiştir

J=4;
[C,L]=wavedec(xn,J,'coif1');

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

lambda=1*sqrt(2*log(N));
C(z(1)+1:z(J+1))=sign(z(1)+1:z(J+1)).*max(0,abs(C(z(1)+1:z(J+1)))-lambda);
```

```
y=waverec(C,L,'coif1');  
  for n=1:N-1  
    ms5=ms5+1/N*((x(n)-y(n)))^2;  
  
  end  
  
ms5;  
end
```


Ek-28 Gürültülü Henon Sinyalini Süzmek için Tasarlanan Süzgeç Programlarını Toplu Olarak Çalıştıran Program (Sabit Değişinti Değerleri Altında Süzgeç Programlarını Karşılaştıran Program)

```
mse1=0;
mse2=0;
mse3=0;
mse4=0;
mse5=0;
mseason1=zeros(1,100);
mseason2=zeros(1,100);
mseason3=zeros(1,100);
mseason4=zeros(1,100);
mseason5=zeros(1,100);
p1var01=zeros(1,100);
p1var1=zeros(1,100);
p1var25=zeros(1,100);
p1var5=zeros(1,100);
yu=zeros(1,100);
l=0;
N=0;
y=0;

for N=100:100:10000
    for rt=1:100
        mse1=mse1+ sorp(N)/100;
        mse2=mse2+ sorp2(N)/100;
        mse3=mse3+ sorp3(N)/100;
        mse4=mse4+ sorp4(N)/100;
        mse5=mse5+ sorp5(N)/100;
    end
```

```
mseason1(N/100)=mse1;  
mseason2(N/100)=mse2;  
mseason3(N/100)=mse3;  
mseason4(N/100)=mse4;  
mseason5(N/100)=mse5;
```

```
mse1=0;  
mse2=0;  
mse3=0;  
mse4=0;  
mse5=0;  
end
```

```
yu=100:100:10000;
```

```
plot(yu,mseason1,'c:',yu,mseason2,'y-',yu,mseason3,'r-',yu,mseason4,'k--  
,yu,mseason5,'g:')  
xlabel('Dizi Uzunluğu n')  
ylabel('Ortalama Karesel Hata')  
grid  
legend('var5p1','var5p2','var5p3','var5p4','var5p5',1)
```

Ek-29 Gürültüsüz Lojistik Regresyon Sinyalin Korelasyon Boyutunun Hesaplanması

```
function count=crdm_log_nssz(r,N)

count=0;
x=zeros(1,N);
x(1)=0.1;

    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end

    for t=1:N-2

        for m=t+1:N-1

            if norm([x(t),x(t+1)]-[x(m),x(m+1)])<=r;
                count=count+1;
            end
        end
    end
end
```

**Ek-30 Gürültülü Lojistik Regresyon Sinyali için Tasarlanan Evrensel Süzgeç
Çıktısının Korelasyon Boyutunun Hesaplanması**

```
function count=crdm_log_torp(r,N)
```

```
x=zeros(1,N);
```

```
d=5;
```

```
h=0;
```

```
ms=0;
```

```
x(1)=0.1;
```

```
w=zeros(d,1);
```

```
c=zeros(d,1);
```

```
c(1)=1;
```

```
crp=zeros(d,d);
```

```
sm=zeros(d,1);
```

```
count=0;
```

```
v=0;
```

```
A=0;
```

```
Y=0;
```

```
for n=2:N
```

```
    x(n)=4*x(n-1)*(1-x(n-1));
```

```
end
```

```
xn=x+randn(1,N);
```

```
Y=zeros(d,1);Y(1)=xn(1);
```

```
h(1)=w.*Y;
```

```
for n=1:N-1
```

```
    crp=crp+Y*Y.');
```

```

sm=sm+Y(1)*Y-c;
A=eye(d,d)+crp;
w=inv(A)*sm;

for v=1:d-1
Y(d+1-v)=Y(d+1-v-1);
end

Y(1)=xn(n+1);
h(n+1)=w.*Y;
end
for t=1:N-2
for m=t+1:N-1
if norm([h(t),h(t+1)]-[h(m),h(m+1)])<=r;
count=count+1;
end
end
end
end
end

```

Ek-31 Gürültülü Lojistik Regresyon Sinyali için Tasarlanan Wiener Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması

```
function count=crdm_log_torp2(r,N)
```

```
d=5;
```

```
top1=zeros(d,d);
```

```
top2=0;
```

```
x=zeros(1,N);
```

```
x(1)=0.1;
```

```
m=0;
```

```
ms2=0;
```

```
u=0;
```

```
count=0;
```

```
p=0;
```

```
v=0;
```

```
for n=2:N
```

```
    x(n)=4*x(n-1)*(1-x(n-1));
```

```
end
```

```
xn=x+randn(1,N);
```

```
Y=zeros(d,1);
```

```
Y(1)=xn(1);
```

```
for n=1:N-1
```

```
    top1=top1+Y*Y.');
```

```
    top2=top2+x(n)*Y;
```

```
for v=1:d-1
```

```
    Y(d+1-v)=Y(d+1-v-1);
```

```
end
```

```

Y(1)=xn(n+1);

end

u=inv((1/N)*top1)*((1/N)*top2);
Y=zeros(d,1);
Y(1)=xn(1);
for n=1:N-1
m(n)=u.*Y;
for v=1:d-1
Y(d+1-v)=Y(d+1-v-1);
end
Y(1)=xn(n+1);
end
m(N)=u.*Y;

for t=1:N-2
for p=t+1:N-1
if norm([m(t),m(t+1)]-[m(p),m(p+1)])<=r;
count=count+1;
end
end
end
end
end

```

**Ek-32 Gürültülü Lojistik Regresyon Sinyali için Tasarlanan LMS Benzeri
Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması**

```
function count=crdm_log_torp3(r,N)
d=5;
x=zeros(1,N);
x(1)=0.1;
k=0.0001;
w=0.1*rand(d,1);
c=zeros(d,1);
c(1)=1;
ms3=0;
z=0;
count=0;
v=0;
    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
xn=x+randn(1,N);
Y=zeros(d,1);
Y(1)=xn(1);
z(1)=w.'*Y;
    for n=1:N-1
        w=w-2*k*Y*(Y'*w)+2*k*(Y(1)*Y-c);
        for v=1:d-1
            Y(d+1-v)=Y(d+1-v-1);
        end
        Y(1)=xn(n+1);
        z(n+1)=w.'*Y;
    end
    for t=1:N-2
```



```
    for m=t+1:N-1
        if norm([z(t),z(t+1)]-[z(m),z(m+1)])<=r;
            count=count+1;
        end
    end
end
end
```

**Ek-33 Gürültülü Lojistik Regresyon Sinyali için Polinom Uydurma Yöntemi
ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun
Hesaplanması**

```
function count=crdm_log_torp4(r,N)
d=5;
x=zeros(1,N);
t=0;
k=0;
x(1)=0.1;
e=16;
y=zeros(1,N);
a=zeros(4,1);
A=zeros(2*e+1,4);
u=ones(2*e+1,1);
f=zeros(2*e+1,1);
g=zeros(2*e+1,1);
j=zeros(2*e+1,1);
s=zeros(2*e+1,1);
segment=0;
y_ext=0;
xn_ext=0;
m=floor(N/e);
y_ext=[zeros(1,N),zeros(1,(m-1)*e+2*e+1-N)];
b=0;
ms4=0;
count=0;

    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
```

```

xn=x+randn(1,N);
xn_ext=[xn,zeros(1,(m-1)*e+2*e+1-N)];
    for k=1:2*e+1
        f(k)=k;
        g(k)=k^2;
        j(k)=k^3;
    end
A=[u f g j];
    for b=0:floor(N/e)-1
        s=xn_ext(b*e+1:b*e+2*e+1);
        a=pinv(A)*s.';
        t(b*e+1:b*e+2*e+1)=(A*a).';
    end
y_ext(1:e)=t(1:e);
    for b=0:floor(N/e)-1
        for l=1:(e+1)
            y_ext(e*b+e+1)=(1-(l-1)/e)*t(e*b+(l+e))+((l-1)/e)*t(e*(b+1)+l);
            y_ext(e*(b+1)+1:e*(b+2)+1)=y_ext(e*b+1:e*(b+1)+1);
        end
    end
y=y_ext(1:N);
    for t=1:N-2
        for p=t+1:N-1
            if norm([y(t),y(t+1)]-[y(p),y(p+1)])<=r;
                count=count+1;
            end
        end
    end
end
end

```

Ek-34 Gürültülü Lojistik Regresyon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi ($\lambda = \sigma \sqrt{\quad}$) ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması

```
function count=crdm_log_torp5(r,N)
d=5;
ms5=0;
a=0;
z=0;
x=zeros(1,N);
x(1)=0.1;
xn=zeros(1,N);
a=0;
z=0;
count=0;
P=0;
L=0;
y=0;
    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
xn=x+randn(1,N);
J=4;
[P,L]=wavedec(xn,J,'coif1');

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end
lambda=1*sqrt(2*log(N));
P(z(1)+1:z(J+1))=sign(z(1)+1:z(J+1)).*max(0,abs(P(z(1)+1:z(J+1)))-lambda);
y=waverec(P,L,'coif1');
```

```
for t=1:N-2
    for m=t+1:N-1
        if norm([y(t),y(t+1)]-[y(m),y(m+1)])<=r;
            count=count+1;
        end
    end
end
end
```

Ek-35 Gürültülü Lojistik Regresyon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi ($\lambda_j^{[1]} = \sigma_j$) ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması

```
function count=crdm_log_torp6(r,N)
d=5;
count=0;
a=0;
z=0;
ms6=0;
x=zeros(1,N);
x(1)=0.1;
xn=zeros(1,N);
a=0;
z=0;
count=0;
P=0;
L=0;
y=0;
    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
xn=x+randn(1,N);
J=4;
[P,L]=wavedec(xn,J,'coif1');
    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

vardet4=mean ((P(z(1)+1:z(2))-mean(P(z(1)+1:z(2))))).^2);
lambdet4 =sqrt(vardet4)*sqrt(2*log(N));
P(z(1)+1:z(2))=sign(P(z(1)+1:z(2))).*max(0,abs(P(z(1)+1:z(2)))-lambdet4);
```

```

vardet3=mean((P(z(2)+1:z(3))-mean(P(z(2)+1:z(3))))).^2);
lambdet3 =sqrt(vardet3)*sqrt(2*log(N));
P(z(2)+1:z(3))=sign(P(z(2)+1:z(3))).*max(0,abs(P(z(2)+1:z(3)))-lambdet3);

vardet2=mean((P(z(3)+1:z(4))-mean(P(z(3)+1:z(4))))).^2);
lambdet2 =sqrt(vardet2)*sqrt(2*log(N));
P(z(3)+1:z(4))=sign(P(z(3)+1:z(4))).*max(0,abs(P(z(3)+1:z(4)))-lambdet2);

vardet1=mean((P(z(4)+1:z(5))-mean(P(z(4)+1:z(5))))).^2);
lambdet1 =sqrt(vardet1)*sqrt(2*log(N));
P(z(4)+1:z(5))=sign(P(z(4)+1:z(5))).*max(0,abs(P(z(4)+1:z(5)))-lambdet1);

y=waverec(P,L,'coif1');
    for t=1:N-2
        for m=t+1:N-1
            if norm([y(t),y(t+1)]-[y(m),y(m+1)])<=r;
                count=count+1;
            end
        end
    end
end
end

```

Ek-36 Gürültülü Lojistik Regresyon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi ($\lambda_j^{[2]} = \sigma_j / \beta$) ile Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması

```

function count=crdm_log_torp7(r,N)
d=5;
a=0;
z=0;
ms7=0;
x=zeros(1,N);
x(1)=0.1;
xn=zeros(1,N);
a=0;
z=0;
count=0;
P=0;
L=0;
y=0;
    for n=2:N
        x(n)=4*x(n-1)*(1-x(n-1));
    end
xn=x+randn(1,N);
J=4;
[P,L]=wavedec(xn,J,'coif1');
    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

vardet4=mean ((P(z(1)+1:z(2))-mean(P(z(1)+1:z(2))))).^2);
lambdet4 =sqrt(vardet4)*sqrt(2*log(N));
P(z(1)+1:z(2))=sign(P(z(1)+1:z(2))).*max(0,abs(P(z(1)+1:z(2)))-lambdet4);

```



```

vardet3=mean((P(z(2)+1:z(3))-mean(P(z(2)+1:z(3))))).^2);
lambdet3 =sqrt(vardet3)*sqrt(2*log(N))/log(3);
P(z(2)+1:z(3))=sign(P(z(2)+1:z(3))).*max(0,abs(P(z(2)+1:z(3)))-lambdet3);

vardet2=mean((P(z(3)+1:z(4))-mean(P(z(3)+1:z(4))))).^2);
lambdet2 =sqrt(vardet2)*sqrt(2*log(N))/log(4);
P(z(3)+1:z(4))=sign(P(z(3)+1:z(4))).*max(0,abs(P(z(3)+1:z(4)))-lambdet2);

vardet1=mean((P(z(4)+1:z(5))-mean(P(z(4)+1:z(5))))).^2);
lambdet1 =sqrt(vardet1)*sqrt(2*log(N))/sqrt(J);
P(z(4)+1:z(5))=sign(P(z(4)+1:z(5))).*max(0,abs(P(z(4)+1:z(5)))-lambdet1);
y=waverec(P,L,'coif1');
    for t=1:N-2
        for m=t+1:N-1
            if norm([y(t),y(t+1)]-[y(m),y(m+1)])<=r;
                count=count+1;
            end
        end
    end
end
end

```

Ek-37 Gürültüsüz Lojistik Regresyon Sinyali ile Gürültülü Lojistik Regresyon Sinyali için Tasarlanan Süzgeç Çıktılarının Korelasyon Boyutlarını Karşılaştıran Program

```
N=5000;
D=0;
store=zeros(1,96);
dongu=0;
dongu1=0;
ort_count=0;
C=zeros(1,3);
dongu2=0;
r=0;
D_collect=zeros(1,8);
dongu3=0;;
fark=zeros(1,7);
fark2=0;

for dongu=0:7

    for i=-1:10
        r=10^(-i);
        for dongu1=1:20
            if dongu==0;
                store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_log_nssz(r,N)/20 ;
            end
            if dongu==1;
                store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_log_torp(r,N)/20 ;
            end
            if dongu==2;
                store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_log_torp2(r,N)/20;
            end
            if dongu==3
```

```

store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_log_torp3(r,N)/20;
    end
    if dongu==4
store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_log_torp4(r,N)/20;
    end
    if dongu==5
store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_log_torp5(r,N)/20;
    end
    if dongu==6
store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_log_torp6(r,N)/20;
    end
    if dongu==7
store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_log_torp7(r,N)/20;
    end

end
    if store(dongu*12+i+2)==0
    break
end
end
for dongu2=1:3
C(dongu2)=store(dongu*12+i+2-dongu2)/[(N-1)*(N-2)/2]
end

D_collect(dongu+1)=pinv([log(r*10^(dongu2-2));log(r*10^(dongu2-
1));log(r*10^(dongu2))])*[log(C(1));log(C(2));log(C(3))]

end

for dongu3=2:8
fark(dongu3-1)=abs(D_collect(dongu3)-D_collect(1))

```

```
end
for dongu3=2:8
    fark2=abs(D_collect(dongu3)-D_collect(1))
    if fark2==min(fark)
        fprintf('en optimum süzgeç=%d. süzgeçtir',dongu3-1)
    break
end
end
```

Ek-38 Gürültüsüz Henon Sinyalin Korelasyon Boyutunun Hesaplanması

```
function count=crdm_hen_nssz(r,N)
m=0;
count=0;
D=0;
N=5000;
x=zeros(1,N);
x(1)=0;
x(2)=0;
    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
    for t=1:N-3
        for m=t+1:N-2
            if norm([x(t),x(t+1),x(t+2)]-[x(m),x(m+1),x(m+2)])<=r;
                count=count+1;
            end
        end
    end
end
```

Ek-39 Gürültülü Henon Sinyali için Tasarlanan Evrensel Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması

```
function count=crdm_hen_korp(r,N)

x=zeros(1,N);
d=5;
h=0;
ms=0;
x(1)=0;
x(2)=0;
w=zeros(d,1);
c=zeros(d,1);
c(1)=1;
crp=zeros(d,d);
sm=zeros(d,1);
count=0;
A=0;
v=0;
    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+randn(1,N);
Y=zeros(d,1);Y(1)=xn(1);
h(1)=w.'*Y;
    for n=1:N-1
        crp=crp+Y*Y.';
        sm=sm+Y(1)*Y-c;
        A=eye(d,d)+crp;
        w=inv(A)*sm;
        for v=1:d-1
            Y(d+1-v)=Y(d+1-v-1);
        end
    end
```

```
Y(1)=xn(n+1);
h(n+1)=w.*Y;
end
for t=1:N-3
    for m=t+1:N-2
        if norm([h(t),h(t+1),h(t+2)]-[h(m),h(m+1),h(m+2)])<=r;
            count=count+1;
        end
    end
end
end
```

Ek-40 Gürültülü Henon Sinyali için Tasarlanan Wiener Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması

```
function count=crdm_hen_korp2(r,N)
d=5;
top1=zeros(d,d);
top2=0;
x=zeros(1,N);
x(1)=0;
x(2)=0;
m=0;
ms2=0;
u=0;
count=0;
p=0;
v=0;
    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+randn(1,N);
Y=zeros(d,1);
Y(1)=xn(1);
    for n=1:N-1
        top1=top1+Y*Y.';
        top2=top2+x(n)*Y;

        for v=1:d-1
            Y(d+1-v)=Y(d+1-v-1);
        end
        Y(1)=xn(n+1);
    end
u=inv((1/N)*top1)*((1/N)*top2);
Y=zeros(d,1);
```



```

Y(1)=xn(1);
  for n=1:N-1
    m(n)=u.*Y;

        for v=1:d-1
          Y(d+1-v)=Y(d+1-v-1);
        end
    Y(1)=xn(n+1);
  end
m(N)=u.*Y;
  for t=1:N-3
    for p=t+1:N-2
      if norm([m(t),m(t+1),m(t+2)]-[m(p),m(p+1),m(p+2)])<=r;
        count=count+1;
      end
    end
  end
end

```

**Ek-41 Gürültülü Henon Sinyali için Tasarlanan LMS Benzeri Süzgeç
Çıktısının Korelasyon Boyutunun Hesaplanması**

```
function count=crdm_hen_korp3(r,N)
d=5;
x=zeros(1,N);
x(1)=0;
x(2)=0;
k=0.0001;
w=0.1*rand(d,1);
c=zeros(d,1);
c(1)=1;
ms3=0;
z=0;
count=0;
v=0;
    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+randn(1,N);
Y=zeros(d,1);
Y(1)=xn(1);
z(1)=w.'*Y;
    for n=1:N-1
        w=w-2*k*Y*(Y'*w)+2*k*(Y(1)*Y-c);
        for v=1:d-1
            Y(d+1-v)=Y(d+1-v-1);
        end
        Y(1)=xn(n+1);
        z(n+1)=w.'*Y;
    end
end
```

```
for t=1:N-3
    for m=t+1:N-2
        if norm([z(t),z(t+1),z(t+2)]-[z(m),z(m+1),z(m+2)])<=r;
            count=count+1;
        end
    end
end
end
```

**Ek-42 Gürültülü Henon Sinyali için Polinom Uydurma Yöntemi ile
Tasarlanan Süzgeç Çıktısının Korelasyon Boyutunun Hesaplanması**

```
function count=crdm_hen_korp4(r,N)
d=5;
x=zeros(1,N);
t=0;
x(1)=0;
x(2)=0;
e=16;
y=zeros(1,N);
a=zeros(4,1);
A=zeros(2*e+1,4);
u=ones(2*e+1,1);
f=zeros(2*e+1,1);
g=zeros(2*e+1,1);
j=zeros(2*e+1,1);
s=zeros(2*e+1,1);
segment=0;
y_ext=0;
xn_ext=0;
m=floor(N/e);
y_ext=[zeros(1,N),zeros(1,(m-1)*e+2*e+1-N)];
b=0;
ms4=0;
count=0;
    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+randn(1,N);
xn_ext=[xn,zeros(1,(m-1)*e+2*e+1-N)];
    for k=1:2*e+1
        f(k)=k;
```

```

g(k)=k^2;
j(k)=k^3;
end
A=[u f g j];
for b=0:floor(N/e)-1
    s=xn_ext(b*e+1:b*e+2*e+1);
    a=pinv(A)*s.';
    t(b*e+1:b*e+2*e+1)=(A*a).';
end
y_ext(1:e)=t(1:e);
for b=0:floor(N/e)-1

    for l=1:(e+1)
        y_ext(e*b+e+1)=(1-(l-1)/e)*t(e*b+(l+e))+((l-1)/e)*t(e*(b+1)+l);
        y_ext(e*(b+1)+1:e*(b+2)+1)=y_ext(e*b+1:e*(b+1)+1);
    end
y=y_ext(1:N);

for t=1:N-3
    for p=t+1:N-2
        if norm([y(t),y(t+1),y(t+2)]-[y(p),y(p+1),y(p+2)])<=r;
            count=count+1;
        end
    end
end
end
end

```

**Ek-43 Gürültülü Henon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi
($\lambda = \sigma$) ile Tasarlanan Süzgeç Çıktısının Korelasyon
Boyutunun Hesaplanması**

```
function count=crdm_hen_korp5(r,N)
d=5;
ms5=0;
a=0;
z=0;
x=zeros(1,N);
x(1)=0;
x(2)=0;
xn=zeros(1,N);
a=0;
z=0;
count=0;
P=0;
L=0;
lambda=0;
y=0;
    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+randn(1,N);
J=4;
[P,L]=wavedec(xn,J,'coif1');

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

lambda=1*sqrt(2*log(N));
P(z(1)+1:z(J+1))=sign(z(1)+1:z(J+1)).*max(0,abs(P(z(1)+1:z(J+1)))-lambda);
```

```
y=waverec(P,L,'coif1');  
  for t=1:N-3  
    for m=t+1:N-2  
      if norm([y(t),y(t+1),y(t+2)]-[y(m),y(m+1),y(m+2)])<=r;  
        count=count+1;  
      end  
    end  
  end  
end
```

**Ek-44 Gürültülü Henon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi
($\lambda_j^{[1]} = \sigma_j$) ile Tasarlanan Süzgeç Çıktısının Korelasyon
Boyutunun Hesaplanması**

```
function count=crdm_hen_korp6(r,N)
d=5;
count=0;
a=0;
z=0;
ms6=0;
x=zeros(1,N);
x(1)=0;
x(2)=0;
xn=zeros(1,N);
a=0;
z=0;
count=0;
P=0;
L=0;
lambda=0;
y=0;
vardet4=0;
vardet3=0;
vardet2=0;
vardet1=0;

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+randn(1,N);
J=4;
[P,L]=wavedec(xn,J,'coif1');
```



```

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end

vardet4=mean((P(z(1)+1:z(2))-mean(P(z(1)+1:z(2))))).^2;
lambdet4 =sqrt(vardet4)*sqrt(2*log(N));
P(z(1)+1:z(2))=sign(P(z(1)+1:z(2))).*max(0,abs(P(z(1)+1:z(2)))-lambdet4);

vardet3=mean((P(z(2)+1:z(3))-mean(P(z(2)+1:z(3))))).^2;
lambdet3 =sqrt(vardet3)*sqrt(2*log(N));
P(z(2)+1:z(3))=sign(P(z(2)+1:z(3))).*max(0,abs(P(z(2)+1:z(3)))-lambdet3);

vardet2=mean((P(z(3)+1:z(4))-mean(P(z(3)+1:z(4))))).^2;
lambdet2 =sqrt(vardet2)*sqrt(2*log(N));
P(z(3)+1:z(4))=sign(P(z(3)+1:z(4))).*max(0,abs(P(z(3)+1:z(4)))-lambdet2);

vardet1=mean((P(z(4)+1:z(5))-mean(P(z(4)+1:z(5))))).^2;
lambdet1 =sqrt(vardet1)*sqrt(2*log(N));
P(z(4)+1:z(5))=sign(P(z(4)+1:z(5))).*max(0,abs(P(z(4)+1:z(5)))-lambdet1);

y=waverec(P,L,'coif1');
    for t=1:N-3
        for m=t+1:N-2
            if norm([y(t),y(t+1),y(t+2)]-[y(m),y(m+1),y(m+2)])<=r;
                count=count+1;
            end
        end
    end
end
end
end

```

**Ek-45 Gürültülü Henon Sinyali için Dalgacık Katsayısı Sıkıştırma Yöntemi
($\lambda_j^{[2]} = \sigma_j$ / β) ile Tasarlanan Süzgeç Çıktısının Korelasyon
Boyutunun Hesaplanması**

```
function count=crdm_hen_korp7(r,N)
d=5;
a=0;
z=0;
ms7=0;
x=zeros(1,N);
x(1)=0;
x(2)=0;
xn=zeros(1,N);
a=0;
z=0;
count=0;
P=0;
L=0;
lambda=0;
y=0;
vardet4=0;
vardet3=0;
vardet2=0;
vardet1=0;

    for n=3:N
        x(n)=1-1.4*(x(n-1))^2+0.3*x(n-2);
    end
xn=x+randn(1,N);
J=4;
[P,L]=wavedec(xn,J,'coif1');
```

```

    for p=1:J+1
        a=a+L(p);
        z(p)=a;
    end
vardet4=mean ((P(z(1)+1:z(2))-mean(P(z(1)+1:z(2))))).^2);
lambdet4 =sqrt(vardet4)*sqrt(2*log(N));
P(z(1)+1:z(2))=sign(P(z(1)+1:z(2))).*max(0,abs(P(z(1)+1:z(2)))-lambdet4);

vardet3=mean((P(z(2)+1:z(3))-mean(P(z(2)+1:z(3))))).^2);
lambdet3 =sqrt(vardet3)*sqrt(2*log(N))/log(3);
P(z(2)+1:z(3))=sign(P(z(2)+1:z(3))).*max(0,abs(P(z(2)+1:z(3)))-lambdet3);

vardet2=mean((P(z(3)+1:z(4))-mean(P(z(3)+1:z(4))))).^2);
lambdet2 =sqrt(vardet2)*sqrt(2*log(N))/log(4);
P(z(3)+1:z(4))=sign(P(z(3)+1:z(4))).*max(0,abs(P(z(3)+1:z(4)))-lambdet2);

vardet1=mean((P(z(4)+1:z(5))-mean(P(z(4)+1:z(5))))).^2);
lambdet1 =sqrt(vardet1)*sqrt(2*log(N))/sqrt(J);
P(z(4)+1:z(5))=sign(P(z(4)+1:z(5))).*max(0,abs(P(z(4)+1:z(5)))-lambdet1);

y=waverec(P,L,'coif1');
    for t=1:N-3
        for m=t+1:N-2
            if norm([y(t),y(t+1),y(t+2)]-[y(m),y(m+1),y(m+2)])<=r;
                count=count+1;
            end
        end
    end
end
end

```

**Ek-46 Gürültüsüz Henon Sinyali ile Gürültülü Henon Sinyali için Tasarlanan
Süzgeç Çıktılarının Korelasyon Boyutlarını Karşılaştıran Program**

```
N=5000;
D=0;
store=zeros(1,96);
dongu=0;
dongu1=0;
ort_count=0;
C=zeros(1,3);
dongu2=0;
r=0;
D_collect=zeros(1,8);
dongu3=0;;
fark=zeros(1,7);
fark2=0;

for dongu=0:7
    for i=-1:10
        r=10^(-i)
        for dongu1=1:20
            if dongu==0;
                store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_hen_nssz(r,N)/20;
            end
            if dongu==1;
                store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_hen_korp(r,N)/20;
            end
            if dongu==2;
                store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_hen_korp2(r,N)/20;
            end
            if dongu==3
                store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_hen_korp3(r,N)/20;
            end
            if dongu==4
```

```

store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_hen_korp4(r,N)/20;
    end
    if dongu==5
store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_hen_korp5(r,N)/20;
    end
    if dongu==6
store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_hen_korp6(r,N)/20;
    end
    if dongu==7
store(12*dongu+i+2)=store(12*dongu+i+2)+crdm_hen_korp7(r,N)/20;
    end

    end
    if store(dongu*12+i+2)==0
    break
end
end
for dongu2=1:3
    C(dongu2)=store(dongu*12+i+2-dongu2)/[(N-2)*(N-3)/2];
end

D_collect(dongu+1)=pinv([log(r*10^(dongu2-2));log(r*10^(dongu2-
1));log(r*10^(dongu2))])*[log(C(1));log(C(2));log(C(3))]

end

for dongu3=2:8
fark(dongu3-1)=abs(D_collect(dongu3)-D_collect(1))
end
for dongu3=2:8
fark2=abs(D_collect(dongu3)-D_collect(1))
    if fark2==min(fark)

```

```
fprintf('en optimum süzgeç=%d. süzgeçtir',dongu3-1)
    break
end
end
```