

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**GRADYAN İNİŞ YÖNTEMİ VE KONTROL SİSTEMLERİNDE
UYGULAMALARI**



YÜKSEK LİSANS

Nagihan YAĞMUR

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Barış Baykant ALAGÖZ

TEMMUZ-2020

**T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**GRADYAN İNİŞ YÖNTEMİ VE KONTROL SİSTEMLERİNDE
UYGULAMALARI**

YÜKSEK LİSANS

**Nagihan YAĞMUR
36183619028**

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Barış Baykant ALAGÖZ

TEMMUZ-2020

TEŐEKKÜR VE ÖNSÖZ

Bu tez alıőmasının her aőamasında yardım, öneri, bilgi, tecrübe ve desteklerini esirgemededen beni her konuda yönlendiren danışman hocam Sayın Do. Dr. Barıő Baykant ALAGÖZ'e,

Tüm hayatım boyunca olduėu gibi tez alıőmam süresince de benden her türlü desteklerini esirgemeyen aileme ve arkadaşlarıma,

teőekkür ederim.



ONUR SÖZÜ

Doktora veya yüksek lisans tezi olarak sunduđum “Gradyan İniş Yöntemi ve Kontrol Sistemlerinde Uygulaması” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuđunu belirtir, bunu onurumla doğrularım.

Nagihan Yağmur



İÇİNDEKİLER

TEŞEKKÜR VE ÖNSÖZ	i
ONUR SÖZÜ	ii
İÇİNDEKİLER.....	iii
ÇİZELGELER DİZİNİ.....	v
ŞEKİLLER DİZİNİ.....	vi
SEMBOLLER VE KISALTMALAR DİZİNİ.....	viii
ÖZET	ix
ABSTRACT	x
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	3
2.1 Optimizasyon ve Temel Bilgiler	3
2.1.1 Optimizasyon test fonksiyonları	6
2.1.1.1 De Jong fonksiyonu	6
2.1.1.2 Rastrigin fonksiyonu	7
2.2 Gradyan İniş Algoritması	7
2.3 Gradyan İniş Algoritması Türleri	16
2.3.1 Algoritmaların kıyaslanması	22
2.4 Kontrol Sistemlerinin Temelleri	27
2.4.1 Kontrol sistemi çeşitleri	28
2.4.1.1 Açık çevrim kontrol sistemi (Geri beslemesiz sistemler).....	28
2.4.1.2 Kapalı çevrim kontrol sistemi (Geri beslemeli sistemler).....	29
2.4.2 Sistem modellemesi ve kontrol	29
2.4.3 Transfer fonksiyonları	30
2.4.4 Kararlılık nedir ?.....	31
2.5 Lyapunov Kararlılık	32
2.6 Direk Gradyan İniş Yöntemi	34
2.7 Adaptif Gradyan İniş Kontrol.....	34
3. SÜREKLİ ZAMAN GRADYAN İNİŞ DİNAMİĞİ LYAPUNOV KARARLILIĞI VE NÜMERİK GRADYAN İNİŞ YÖNTEMİ ÇÖZÜMLERİ İLE KIYASLAMA	41
3.1 Sürekli Zaman Gradyan İniş Dinamiği ve Nümerik Gradyan İniş Yöntemi...41	
3.2 Gradyan İniş Dinamiği Kararlılık Analizi	42
3.3 Örnek Nümerik Analiz	43
4. KARARLI, BİRİNCİ DERECE, ZAMAN GECİKMELİ DİNAMİK SİSTEM MODELLERİNİN ZAMANLA DEĞİŞEN FIR FİLTRE MODELİNE GÖRE ADAPTİF GRADYAN İNİŞ KONTROLÜ	48
4.1 Teorik Arka Plan	48
4.2 Zamanla Değişen FIR Filtre Modeli Varsayımı ile Adaptif Gradyan İniş Kontrolü 49	
4.3 Simülasyon Çalışması	50
5. BİRİNCİ MERTEBE ZAMAN GECİKMELİ SİSTEM DİNAMİĞİNİN GRADYAN İNİŞ YÖNTEMİNE DAYALI ADAPTİF IIR FİLTRELER İLE MODELLENMESİ	55
5.1 Teorik Arkaplan.....	56
5.2 Adaptif IIR Filtre Yapısının Gradyan İniş Yöntemi Çözümlemesi.....	57
5.3 Nümerik Çalışma	58
6. SONUÇ VE ÖNERİLER	71

7. KAYNAKÇA	73
8. ÖZGEÇMİŞ.....	81



ÇİZELGELER DİZİNİ

Çizelge 2.1: Optimizasyon problem çeşitleri.	4
Çizelge 3.1: Nümerik çözümün ortalama mutlak hata değerleri.....	46
Çizelge 4.1 : Kontrol performans testleri için parametrik olarak değişen plant modelleri ve bu modellerin $G_1(s)$ modeline göre değişim oranları.....	51
Çizelge 4.2 : Plant fonksiyonlarının ortalama mutlak hataları (MAE).	54
Çizelge 5.1 : Farklı zaman gecikmesine sahip BMZG sistem modelleri.....	59
Çizelge 5.2 : Farklı zaman gecikmeli BMZG sistemlerin adaptif IIR filtre ile modellenme performansları.....	70



ŞEKİLLER DİZİNİ

Şekil 2.1 : Kısıtlamalı – kısıtlamasız problem figürü.	4
Şekil 2.2 : Tek değişkenli – çok değişkenli problem figürü.	5
Şekil 2.3 : Ayrık değişkenli – sürekli değişkenli optimizasyon problemi.	5
Şekil 2.4 : Doğrusal ve doğrusal olmayan eğriler.	5
Şekil 2.5 : İki boyutlu De Jong fonksiyonu.	7
Şekil 2.6 : Rastrigin fonksiyonunun iki boyutlu gösterimi.	7
Şekil 2.7 : Türevin tanımının geometrik yorumu.	8
Şekil 2.8 : Kritik nokta türleri.	9
Şekil 2.9 : Öğrenme katsayısı değerine göre optimum noktaya ilerleyiş.	11
Şekil 2.10 : Çeşitli öğrenme katsayılarının yakınsama üzerindeki etkisi.	11
Şekil 2.11 : Başlangıç noktası seçimine göre minimum noktaya gitme güzergahı.	13
Şekil 2.12 : De Jong fonksiyonunun gradyan iniş yöntemi ile optimizasyonu; (a) 2 boyutlu De Jong fonksiyonu gösterimi, (b) Gradyan iniş yöntemi çözümlerinin De Jong fonksiyonu üzerindeki ilerleyişi, (c) Optimizasyon sırasında hata fonksiyonu değişimi.	14
Şekil 2.13 : Rastrigin fonksiyonunu gradyan iniş yöntemi ile optimize etme; (a) Gradyan iniş yöntemi ile optimize edilmiş Rastrigin fonksiyonu, (b) Optimizasyon sırasında hata fonksiyonu değişimi, (c) ve (d) x_1 ve x_2 değişimi.	15
Şekil 2.14 : Gradyan iniş yöntemi yakınsama örneği, (a) momentum uygulanmamış, (b) momentum uygulanmış.	17
Şekil 2.15 : De Jong fonksiyonunun momentum yöntemi ile optimizasyonu; (a) Momentum yöntemi çözümlerinin De Jong fonksiyonu üzerinde ilerleyişi, (b) Optimizasyon sırasında x_1 ve x_2 değerlerinin değişimi, (c) Optimizasyon sırasında hata fonksiyonu değişimi.	18
Şekil 2.16 : De Jong fonksiyonunun adagrad yöntemi ile optimizasyonu; (a) Adagrad yöntemi çözümlerinin De Jong fonksiyonu üzerinde ilerleyişi, (b) Optimizasyon sırasında x_1 ve x_2 değerlerinin değişimi, (c) Optimizasyon sırasında hata fonksiyonu değişimi.	20
Şekil 2.17 : De Jong fonksiyonunun rmsprop yöntemi ile optimizasyonu; (a) Rmsprop yöntemi ile optimize edilmiş De Jong fonksiyonu üzerinde çözüm noktalarının ilerleyişi, (b) Optimizasyon sırasında x_1 ve x_2 değerlerinin değişimi, (c) Optimizasyon sırasında hata fonksiyonu değişimi.	21
Şekil 2.18 : Öğrenme katsayısı $\eta = 0.01$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.	22
Şekil 2.19 : Öğrenme katsayısı $\eta = 0.1$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.	23
Şekil 2.20 : Öğrenme katsayısı $\eta = 0.2$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.	24
Şekil 2.21 : Öğrenme katsayısı $\eta = 0.5$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.	25
Şekil 2.22 : Öğrenme katsayısı $\eta = 1$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.	26
Şekil 2.23 : Kontrol sistemi temel öğeleri.	27
Şekil 2.24 : Açık çevrim kontrol sistemlerinin öğeleri.	29
Şekil 2.25 : Negatif geribeslemeli kapalı çevrim kontrol sistemi.	29

Şekil 2.26 : Açık çevrim kontrol sistemi blok diyagram gösterimi.	30
Şekil 2.27 : Kapalı çevrim kontrol sistemi blok diyagram gösterimi	31
Şekil 2.28 : Genel MRAC yapısı [23].....	35
Şekil 2.29 : Önerilen RBAK sistem blok diyagramı [23]	36
Şekil 2.30 : Dördüncü dereceden anlık giriş – çıkış ilişkisi için gradyan iniş kontrolün blok diyagramı [96]	40
Şekil 3.1 : Nümerik gradyan iniş çözümünün üç boyutlu yörüngesi	45
Şekil 3.2 : Nümerik gradyan iniş çözümünün iki boyutlu yörüngesi.....	45
Şekil 3.3 : Farklı örnekleme periyotları için nümerik çözüm ile analitik çözüm karşılaştırılması.....	46
Şekil 3.4 : Anlık mutlak hata dağılımı	47
Şekil 4.1 : Matlab Simulink kontrol sistemi simülasyon modeli	51
Şekil 4.2 : Orijinal plant fonksiyonu $G_1(s)$ ve parametrik değişmiş varyantları $G_2(s)$, $G_3(s)$ ve $G_4(s)$ fonksiyonları için PI kontrol sisteminin yanıtlarını gösteren kontrol simülasyon sonuçları.	52
Şekil 4.3 : Orijinal plant fonksiyonu ve parametrik değişmiş varyantları için önerilen adaptif gradyan iniş kontrol sisteminin simülasyon sonuçları.	53
Şekil 4.4 : $G_2(s)$ ve $G_4(s)$ için her iki kontrolördeki kontrol hataları.....	54
Şekil 5.1 : Dinamik sistem modellemesi için geliştirilen sistemin blok diyagramı	58
Şekil 5.2 : BMZG sistem modellemesi için geliştirilen simulink simülasyon ortamı	59
Şekil 5.3 : Kare dalga giriş işareti için $G_0(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki başlangıç anı cevapları, (c) 1400 - 1500 sn aralığındaki cevaplar	60
Şekil 5.4 : $G_0(s)$ fonksiyonu için model hatasının değişimi	61
Şekil 5.5 : Kare dalga giriş işareti için $G_1(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki yakın görüntüsü, (c) 1400 - 1500 sn aralığındaki yakın görüntüsü.....	62
Şekil 5.6 : $G_1(s)$ fonksiyonu için model hatasının değişimi.....	63
Şekil 5.7 : Kare dalga giriş işareti için $G_5(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki yakın görüntüsü, (c) 1400 - 1500 sn aralığındaki yakın görüntüsü.....	64
Şekil 5.8 : $G_5(s)$ fonksiyonu için model hatasının değişimi	65
Şekil 5.9 : Kare dalga giriş işareti için $G_{10}(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki yakın görüntüsü, (c) 1400 - 1500 sn aralığındaki yakın görüntüsü.....	66
Şekil 5.10 : $G_{10}(s)$ fonksiyonu için model hatasının değişimi	67
Şekil 5.11 : Kare dalga giriş işareti için $G_{20}(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki yakın görüntüsü, (c) 1400 - 1500 sn aralığındaki yakın görüntüsü.....	68
Şekil 5.12 : $G_{20}(s)$ fonksiyonu için model hatasının değişimi.....	69

SEMBOLLER VE KISALTMALAR

min	: Minimum
md	: d Doğrusunun Eğimi
α	: d Doğrusunun Düzlemle Yaptığı Açısı
lim	: Limit İfadesi
Δx	: x 'te Değişim Miktarı
\approx	: Yaklaşık Eşit
$'$, ∂	: Türev
λ, γ, η	: Öğrenme Katsayısı
Δt	: t 'de Değişim Miktarı
ϱ	: Momentum Hız Katsayısı
G	: Geçmiş Gradyanların Karesel Toplamı
SGD	: Stokastik Gradient Descent
Gİ	: Gradyan İniş
RBAK	: Referans Biçimlendiren Adaptif Kontrol
MRAK	: Model Referans Adaptif Kontrol
AGİK	: Adaptif Gradyan İniş Kontrol
LTI	: Linear Time Invariant
FIR	: Finite Impulse Response
IIR	: Infinite Impulse Response
TV-FIR	: Time Varying Finite Impulse Response
IC	: Akıllı Kontrol
PID	: Proportional Integral Derivative
PI	: Proportional Integral
τ	: Zaman Sabiti
C_{pid}	: PI Transfer Fonksiyonu
L	: Zaman Gecikmesi
OMH	: Ortalama Mutlak Hata
OKH	: Ortalama Karesel Hata
ORH	: Ortalama Relatif Hata
BMZG	: Birinci Mertebe Zaman Gecikmeli
LSM	: Least Mean Square

ÖZET

Yüksek Lisans Tezi

GRADYAN İNİŞ YÖNTEMİ VE KONTROL SİSTEMLERİNDE UYGULAMASI

NAGİHAN YAĞMUR

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

81+X sayfa

2020

Danışman: Doç. Dr. Barış Baykant ALAGÖZ

Gradyan iniş algoritması makine öğrenmesi ve derin öğrenme yöntemlerinde sıklıkla kullanılan bir optimizasyon tekniğidir. Gradyan iniş yöntemi problemlere kolay uygulanabilir olması ve basit nümerik çözümler sağlamadığı için tercih edilen doğrusal olmayan bir optimizasyon yöntemidir. Birçok alanda yaygın olarak kullanımına rağmen kontrol uygulamaları oldukça sınırlı kalmıştır. Bunun temel nedeni gerçek zamanlı kontrol sistemlerinin hızlı cevaba ve garanti edilmiş nümerik çözüm kararlılığına ihtiyaç duymasıdır. Bu tez çalışmasında gradyan iniş yöntemi ve bu yöntemin kontrol uygulamaları incelenmiştir. Öncelikle sürekli gradyan iniş dinamiğinin Lyapunov kararlılığı incelenmiş ve nümerik gradyan iniş çözümleri ile kıyaslanmıştır. Daha sonra adaptif gradyan iniş kontrol çalışmaları incelenmiştir. Zaman gecikmeli dinamik sistem modellerinin adaptif gradyan iniş kontrol performansını iyileştirmek için zamanla değişen FIR model için gradyan iniş yöntemi çözümleri elde edilmiştir. Bu çözümlerin adaptasyon kabiliyeti simülasyon çalışmalarına göre değerlendirilmiştir. Adaptif gradyan iniş kontrol yönteminin adaptasyon performansının iç-modelin kontrol edilen sistemi temsil edebilme kabiliyetine bağlı olarak iyileştirilebileceği gözlemlenmiştir. Bu nedenle adaptif IIR filtre yapılarının zaman gecikmeli dinamik sistem modellerini temsil performansı incelenmiştir. Düşük zaman gecikmeleri için adaptif IIR filtrelerin dinamik sistem modellemesinde kullanılabilirliği görülmüştür. Bu tez çalışmasında elde edilen bulgular adaptif gradyan iniş kontrol ile ilgili gelecek yapabilecek çalışmalara katkı sağlayabilir.

Anahtar Kelimeler: Gradyan iniş yöntemi, Lyapunov kararlılık, kontrol sistemleri, zamanla değişen FIR filtre, sistem modelleme, adaptif IIR filtre, zaman gecikmeli dinamik sistemler

ABSTRACT

Master Thesis

THESIS TITLE

Nagihan YAGMUR

Inonu University

Graduate School of Nature and Applied Sciences

Department of Computer Engineering

81+X sayfa

2020

Supervisor: Doc. Dr. Baris Baykant ALAGOZ

Gradient descent algorithm is an optimization technique that has been frequently used in machine learning and deep learning methods. Gradient descent method is a nonlinear optimization method that is preferred since it is easily applicable to problems and providing simple numerical solutions. Although gradient descent method is widely used in many fields, its applications in control are quite limited. Main reason of this trend is that the real-time control systems require fast response and guaranteed stability of numerical solutions. In this thesis, gradient descent method and its control applications are investigated. Firstly, Lyapunov stability of continuous gradient descent dynamics is investigated, and it is compared with solutions of numerical gradient descent method. Afterwards, adaptive gradient descent control studies are surveyed in detail. To improve control performance of adaptive gradient control for time-delay dynamical systems, time variable FIR model solutions of gradient descent method are obtained. The adaptation performance of these solutions is evaluated according to simulation studies. It is observed that the adaptation performance of gradient descent control can be enhanced depending on the capability of internal model to represent the controlled system. For this reason, we investigated representation performance of adaptive IIR filters for time delay dynamical systems. We observed that the adaptive IIR filters can be used for modeling of the dynamic systems with the lower time delay. The findings of this thesis can contribute to future works that are related to adaptive gradient descent control.

Keywords: Gradient descent method, Lyapunov stability, control systems, time varying FIR filters, system modeling, adaptive IIR filter, time delay dynamic systems.

1. GİRİŞ

Bu tez çalışmasında gradyan tabanlı doğrusal olmayan bir optimizasyon tekniği olan gradyan iniş yönteminin adaptif kontrol sistemlerinde uygulanması araştırılmıştır. Gradyan iniş yöntemi doğrusal olmayan optimizasyonların çözümünde uygulanabilen nümerik bir optimizasyon tekniğidir [1]. Gradyan iniş yönteminde, amaç fonksiyonun gradyanının azalan yönünde adım adım ilerlenerek iteratif olarak amaç fonksiyonun minimumuna yaklaşılmasını sağlar.

Nümerik çözüm tabanlı problemlere kolayca uygulanabilen bir yöntem olan gradyan iniş yönteminin kullanım alanları gün geçtikçe artmaktadır. Örnek verilecek olursa; robotik çalışmalarında [2], yapay sinir ağlarında [3, 4, 5, 6, 7, 8, 9], bilgisayarla görü çalışmalarında [10, 11], eniyileme algoritmalarında [12], derin öğrenme çalışmalarında [13] sıkça kullanılmaktadır. Fakat, kontrol sistemlerinde gradyan iniş yöntemi yaygın bir kullanım bulamamıştır. 90'ların sonlarına doğru direkt gradyan iniş kontrol adı verilen model tabanlı kontrol çalışmaları yapılmış fakat yaygınlaşmamıştır [14, 15, 16, 17]. Kontrol uygulamalarında gradyan iniş yönteminin doğrudan kullanımının sınırlı kalmasının nedenleri matematiksel olarak modellenen bir dinamik sistem modelinin gerçek zamanlı optimizasyonunun karmaşıklığı, gradyan iniş yöntemi ile dinamik sistemlerin kararlılık ve yerleşme noktası kontrolünün garanti edilmesinin zorluluğudur. Öncelikle gradyan iniş yöntemi nümerik çözüm kararlılığının iyileştirilmesine ihtiyaç vardır. Bu konuda gradyan iniş yöntemin çözüm kararlılığını incelemek için Lyapunov kararlılık koşulları uygulanmıştır [18]. Gerçek zamanlı dinamik sistem kontrol uygulamalarında yaygın kullanım bulmasa bile modelleme uygulamalarında gradyan iniş yönteminin etkinliğini gösteren çalışmalar yapılmıştır [19, 20]. Direkt gradyan iniş kontrolü yakın zamanlı yapılan çalışmalar ile akıllı kontrol başlığı altında yeniden gündeme gelmeye başlamıştır [21, 22, 23, 24]. Bu çalışmalarda özellikle adaptif kontrol uygulamaları için çözümler önerilmiştir.

Bu tez çalışmasında gradyan iniş yönteminin adaptif kontrol sistemlerinde uygulaması incelenmiştir. Bu amaçla iç sistem modellemesinde (internal model) kullanımı araştırılmış ve adaptif kontrol sisteminin kararlılığının iyileştirilebilmesi için gradyan iniş çözümlerinin kararlılığı incelenmiştir. Simülasyon çalışmaları Matlab

Simulink ortamında gerçekleştirilmiştir. Tez çalışmasının bölümleri şu şekilde özetlenebilir:

İkinci bölümde temel bilgiler ve literatür özeti sunulmuştur. Bu bölümde optimizasyon, gradyan iniş yöntemi, türleri ve test fonksiyonları üzerinde performans karşılaştırmaları, kontrol sistemleri temel bilgiler, gradyan iniş yönteminin kontrolde uygulamaları incelenmiş ve literatür özeti sunulmuştur.

Üçüncü bölümde sürekli gradyan iniş dinamiğinin Lyapunov kararlılığı incelenmiş ve gardyan iniş çözümlerinin Lyapunov kararlılığını sağlayan koşullar araştırılmıştır.

Dördüncü bölümde birinci mertebe zaman geçikmeli dinamik sistem modellerinin zamanla değişen sonlu impuls cevabı (FIR) iç modelleri (internal model) yardımı ile adaptif gradyan iniş kontrolü önerilmiş ve performansı incelenmiştir.

Beşinci bölümde adaptif gradyan iniş yöntemin adaptasyon performansının iyileştirilmesi için zamanla değişen sonluz impuls cevabı (IIR) iç modellerinin gradyan iniş çözümleri elde edilmiş ve birinci mertebe zaman gecikmeli sistemleri modelleme performansları araştırılmıştır.

Altıncı bölümde elde edilen bulgular ve sonuçlar sunulmuştur.

2. LİTERATÜR ARAŞTIRMASI

2.1 Optimizasyon ve Temel Bilgiler

Mühendislik problemlerinin çözümünde önemli bir yere sahip olan ve matematikte eniyileme olarak da adlandırılan optimizasyon, sistem kaynakların en iyi biçimde kullanımına imkan sağlayan koşulları ve durumları belirlemede kullanılır. Optimizasyon kısaca ele alınan fonksiyonu minimize ya da maksimize etme süreci ve işlemlerini ifade eder [25]. Optimizasyon yöntemleri gerçek hayatta karşılaşılan problemlerin optimal çözümlerinin elde edilmesinde yararlanılmaktadır [26].

Mühendislik problemlerinde optimizasyon tekniklerinin uygulanması temelde iki aşamaya ihtiyaç duyar. Bunlar modelleme aşaması ve uygulanan optimizasyon yönteminin çözümlenmesi aşamasıdır [27]. Modelleme, ele alınan problemin matematiksel olarak ifade edilmesidir. Optimal çözümlenmesi aşamasında ise modele göre tanımlanmış olan problemin en iyi çözümü aranır. Optimize edilecek istenen bütün sistemler için sistemi temsil eden bir modele ihtiyaç duyulur. Maliyetlerin minimize edilmesi, verimliliğin maksimize edilmesi gibi gereksinimlerden dolayı modelleme ve optimizasyon ekonomi ve üretim planlamasının en temel konuları haline dönüşmüştür. Bu konularda çalışmaların 1930'larda başladığını görmekteyiz. ABD'nin ekonomik yapısı ve dış ticaretinin modellenmesi yapılmıştır [28, 29]. Üretim planlamasında karşılaşılan problemlerin modellenmesi [30], ulaşım sektörünün sorunlarını azaltmaya yönelik modelleme [31] çalışmaları yapılmıştır. Daha sonra ekonomik alanda kapasite kullanımına dair modellemeler yapılmıştır [32, 33, 34, 35, 36].

Optimizasyon problemlerinin çözümüne yönelik önemli çalışmalar 1940'lara doğru görülmeye başlamıştır [37]. Optimizasyon için kullanılan modeller, sistemin optimizasyonda kullanılan parametrelere bağlı davranışını matematiksel olarak anlatan yapılarıdır [38]. Bu matematiksel anlatım oluşturma aşamalarında ilk olarak sistemin özelliklerini gösteren parametreler, en iyi sonuçlara yakınsayacak değişkenler ve sistemin performans özellikleri belirlenerek problem tanımı yapılır [38]. Daha sonra sistemin sınırlarını belirleyen kısıtlar belirlenir [27]. Bir optimizasyon problemi aşağıda temsili örneği verilen iki bileşenden oluşur:

$$\text{Problem Tanımlama :} \quad \min E(x) \quad (2.1)$$

$$\text{Kıstas Tanımlama : s.t.} \quad \begin{aligned} G(x) < 0 \\ F(x) > 0 \end{aligned} \quad (2.2)$$

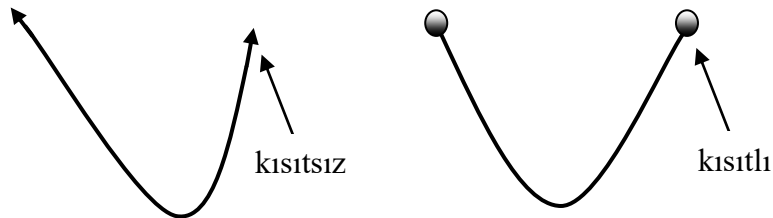
Örnekteki optimizasyon probleminde sistemin amaç fonksiyonu yani minimize veya maksimize etmek istediğimiz sistemin performans ölçütü $E(x)$ ile ifade edilmiştir. $G(x) < 0$ ve $F(x) > 0$ kısıtlarında ise optimizasyondan pratik olarak anlamlı sonuçlar alabilmek için ifade edilen sınırlardır [27, 39]. Amaç fonksiyonunun optimal değeri kısıt fonksiyonları ve koşullarını sağlamak zorundadır. Literatürde yaygın olarak minimize edilmek istenen fonksiyonlar hata fonksiyonları, maliyet fonksiyonu, kayıp fonksiyon olarak adlandırılmaktadır [40].

Optimizasyon problemleri Çizelge 2.1’de görüldüğü üzere içerdikleri değişken sayıları, amaç fonksiyonun türü ve kısıtlamaların özelliklerine göre sınıflanmıştır [41].

Çizelge 2.1: Optimizasyon problem çeşitleri.

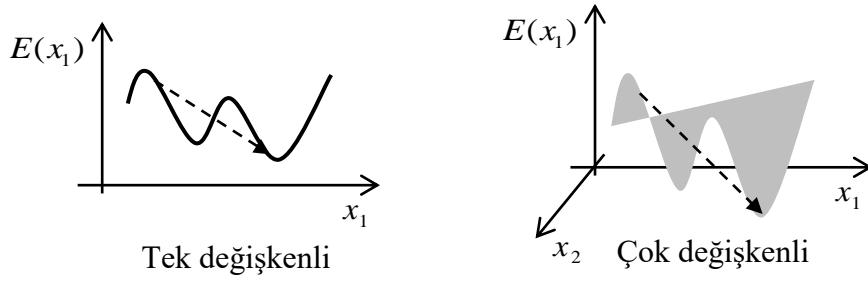
Karakteristiği	Özelliği	Sınıflandırma
Değişken sayısı	Bir	Tek değişkenli
	Birden fazla	Çok değişkenli
Değişken türü	Sürekli	Sürekli
	Tamsayı	Tamsayı veya kesikli (ayrık)
	Hem sürekli hem Tamsayı	Karışık tamsayı
Hedef ve kısıtlayıcı fonksiyonlar	Doğrusal fonksiyon	Doğrusal
	Kuadratik fonksiyon	Kuadratik
	Doğrusal olmayan fonksiyon	Doğrusal olmayan
Problem formülasyonu	Kısıtlama var	Kısıtlı
	Kısıtlama yok	Kısıtlamasız

Çizelgede verilen bazı durumları görsel olarak inceleyelim: Kısıtlar arama uzayını sınırlar. Şekil 2.1’de kısıtsız durum ve kısıtlı durum temsili olarak gösterilmiştir [42].



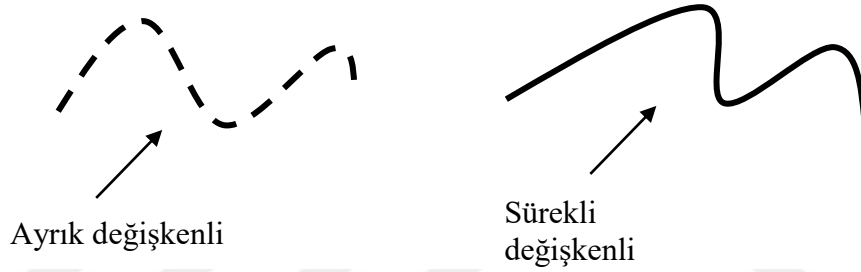
Şekil 2.1 : Kısıtlı – kısıtlamasız problem figürü.

Şekil 2.2’de bir veya daha fazla değişkenli optimizasyon temsili gösterilmiştir [42].



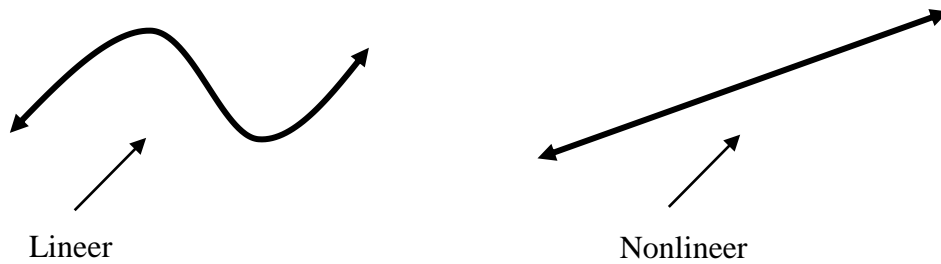
Şekil 2.2 : Tek değişkenli – çok değişkenli problem figürü.

Optimizasyon problemleri ayrık değişkenli (mesela yalnızca tamsayı değerlerine sahip) veya sürekli değişkenli olabilir [42]. Şekil 2.3'te bu durum temsili gösterilmiştir.



Şekil 2.3 : Ayrık değişkenli – sürekli değişkenli optimizasyon problemi.

Optimizasyon problemleri doğrusal veya doğrusal olmayan karaktere sahip olabilir [42]. Amaç fonksiyonları doğrusal ise bu optimizasyon probleminin doğrusal optimizasyon problemi olmasını sağlar.



Şekil 2.4 : Doğrusal ve doğrusal olmayan eğriler.

Örneğin aşağıda tanımlanan optimizasyon problemi kısıtsız, tek değişkenli bir doğrusal optimizasyon problemi ifade eder.

$$\min f(x) = 2x_1 + 1 \quad (2.3)$$

Örneğin aşağıda tanımlanan optimizasyon problemi çok değişkenli, kısıtlı ve lineer olmayan bir optimizasyon problemidir.

$$\begin{aligned} \min f(x) &= 5x_1^2 - 4x_2^3 \\ s.t. & \\ x_1 + x_2 &> 3 \end{aligned} \quad (2.4)$$

Optimizasyon teknikleri matematiksel ve yöntemsel farklılıkları bakımından 4 temel grupta toplanabilir: Analitik yöntemler, nümerik yöntemler, sezgisel yöntemler ve metasezgisel yöntemler. Analitik ve nümerik optimizasyon yöntemleri deterministik karakterdedirler. Sezgisel ve metasezgisel yöntemler stokastik özellikler barındırırlar.

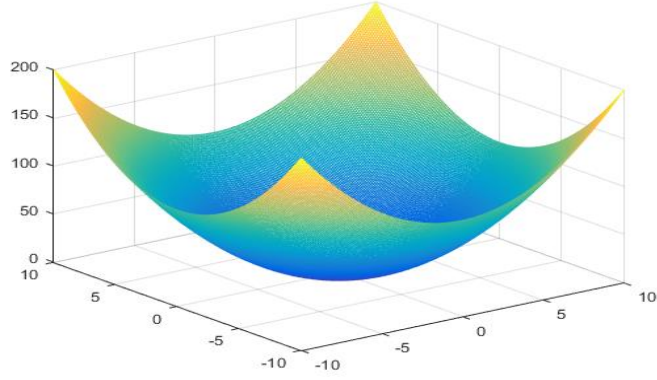
2.1.1. Optimizasyon test fonksiyonları

Literatürde optimizasyon yöntemlerinin performansını değerlendirmek için bir çok test fonksiyonu kullanılmaktadır [43]. Bu tez çalışmasında test fonksiyonlarından bazıları Matlab'ta kodlanmış ve gradyan iniş yöntemlerinin testi için kullanılmıştır. Aşağıda optimizasyon yöntemlerinin performans testinde yaygın kullanılan test fonksiyonları kısaca tanıtılmıştır.

2.1.1.1 De Jong fonksiyonu :

En basit test fonksiyonlarından biridir.

$$\text{Genel Formül : } f(x) = \sum_{i=1}^n x_i^2 \quad (2.5)$$

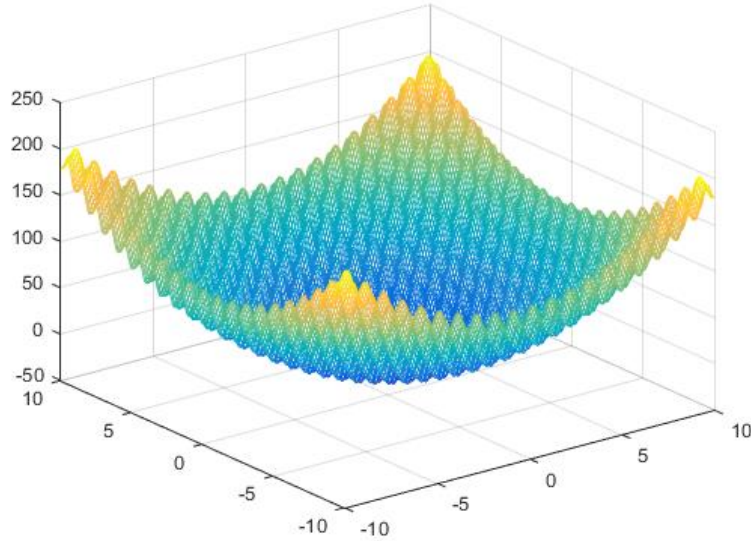


Şekil 2.5 : İki boyutlu De Jong fonksiyonu.

2.1.1.2 Rastrigin fonksiyonu :

Kosinüs fonksiyonu ilave edilmesi ile De Jong fonksiyonuna dayanır.

$$\text{Genel formül : } f(x) = 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] \quad (2.6)$$



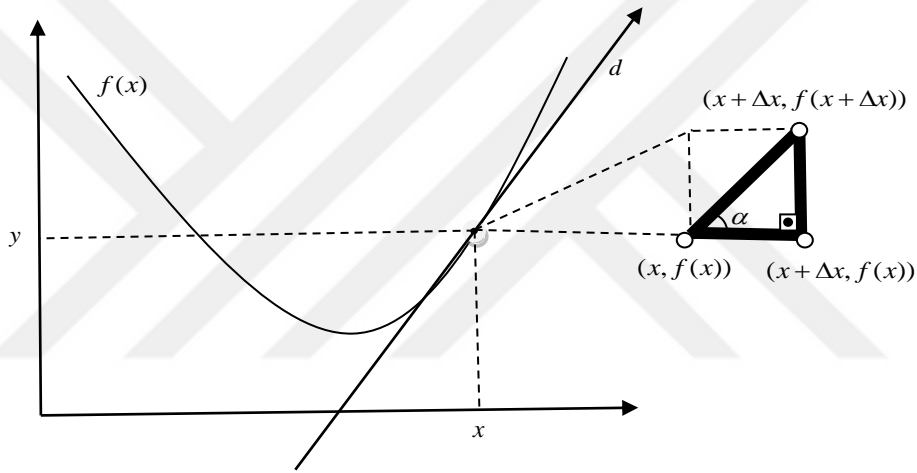
Şekil 2.6 : Rastrigin fonksiyonunun iki boyutlu gösterimi.

2.2 Gradyan İniş Algoritması

Gradyan iniş algoritması 1847’de Fransız matematikçi Louis Augustin Cauchy tarafından icat edilmiştir [44]. Makine öğrenmesi uygulamalarında ve geri yayımlı yapay sinir ağları optimizasyonunda matematiksel formülasyonunun basit olması ve iteratif nümerik çözüm sağlayabilmesi nedeniyle en yaygın kullanılan algoritmalarından biridir. Bu yöntem, iteratif olarak fonksiyonların minimum noktalarına doğru çözümü ilerletir ve

doğrusal olmayan programlamada (nonlinear programming) yaygın kullanılır. Yöntem pratik uygulamalarda öğrenme hatasının azalan yönünde model parametrelerinin güncellenmesi için yaygın kullanım bulur. Gradyan iniş algoritması fonksiyonun türevine bağlıdır. Fonksiyonun türevine (çok değişkenli fonksiyonlar için gradyan işlemidir) göre işlemi gerçekleştirir. Bu nedenle gradyan iniş algoritmasının uygulanacağı fonksiyonların türevlenebilir olması gerekmektedir.

Gradyan iniş yönteminin daha iyi anlaşılması için türev tanımı iyice kavranmalıdır. Türev, tanımlanan fonksiyonun değişkeninde meydana gelen küçük değişimin, yapılan bu değişikliğe göre fonksiyonun değerinde meydana getireceği değişikliğe oranıdır. Bu oran aynı zamanda fonksiyonun türevi alınan noktada eğimini ifade eder.



Şekil 2.7: Türevin tanımının geometrik yorumu.

Şekil 2.7'de d doğrusunun fonksiyona teğet geçtiği nokta yakın civarında x 'e eklenen yeterince küçük bir değer $f(x)$ 'de meydana getirdiği değişime oranı d doğrusunun eğimine yakınsar. Bu oran aynı zamanda $f(x)$ 'in x noktası yakın civarında türevini ifade eder. Diğer bir ifade ile, girişte oluşturulan küçük değişimin çıktıda oluşturabileceği değişim miktarını ifade eder [45].

$$md = \tan \alpha = f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (2.7)$$

Denklem (2.7)'de içler dışlar çarpımı yapılırsa,

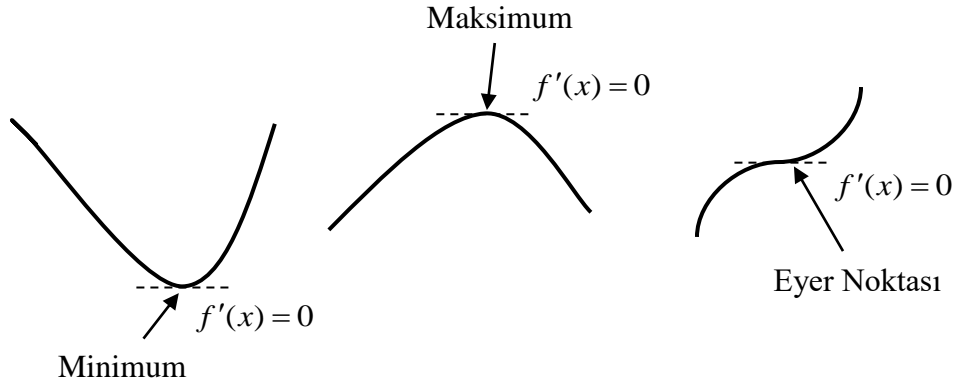
$$f(x + \Delta x) \approx f(x) + \Delta x f'(x) \quad (2.8)$$

$f(x)$ yalnız bırakılırsa,

$$f(x) = f(x + \Delta x) - \Delta x f'(x) \quad (2.9)$$

elde edilir. Optimizasyonda türev işlemi $f(x)$ fonksiyonunda ufak bir iyileştirme elde etmek için x 'i nasıl değiştirmemiz gerektiğini gösterebilmesidir [45].

Eğer fonksiyonun azalan yönünde hareket edilmek istenirse x değişkeni eğimin (türevin) negatif olduğu yönde ilerletilir. Böylece fonksiyonun dip noktasına ulaşılabilir. Eğer x değişkeni eğimin pozitif olduğu yönde ilerletilirse tepe noktasına ulaşılır. Türevin sıfır olduğu nokta ($f'(x)=0$) tepe veya dip noktasıdır ve x değişkenin ilerlemesi durur. Bu nokta yani fonksiyonun eğiminin 0 olduğu nokta ulaşılmak istenen nokta kritik nokta veya ekstremum noktasıdır. Temelde eğimin sıfır olduğu 3 tip kritik nokta olabilmektedir. Bu kritik noktalar minimum nokta, maximum nokta ve eyer (saddle) noktasıdır.



Şekil 2.8 : Kritik nokta türleri.

Yerel maksimum noktası fonksiyonun artandan azalana yön değiştirdiği noktadır. Dolayısı ile yerel maksimumda $f(x)$ değeri tüm komşu noktalarından daha yüksek değerde olur [45]. Yerel minimum noktası fonksiyonun azalandan artana yön değiştirdiği noktadır. Bu noktada fonksiyon değeri komşu noktalarındaki değerlerinden küçüktür [45]. Diğer bir kritik nokta türü olan eyer noktaları ne bir yerel minimum neden bir yerel maksimum noktadır. Fonksiyonun bu nokta yakın civarında değerleri fonksiyonun bu noktadaki değerinden büyük yada küçük olabilir [45].

$f(x)$ fonksiyonunun mutlak en düşük değerini elde eden nokta global minimum noktasıdır. Bir fonksiyonun sadece bir adet global minimum noktası vardır. Diğer minimumlar lokal minimum noktaları olarak adlandırılır [45]. Derin öğrenme uygulamalarında optimal olmayan birçok local minimum noktası olan veya birçok eyer noktası olan problemler optimize edilmeye çalışılmaktadır [45]. Ancak bu noktalar global minimuma erişmeyi amaçlayan optimizasyon yöntemleri için bir zorluk yaratmaktadır.

Fonksiyon değerinin minimum olduğu bir noktaya ulaşmak için $f(x)$ 'in gradyanını (türevini) kullanma işlemi gradyan iniş yöntemi olarak adlandırılmıştır. Denklem 2.9'dan ilham alarak temel nümerik gradyan iniş yöntemi,

$$x_i[n+1] = x_i[n] - \eta_i \frac{\partial F}{\partial x_i} \quad (2.10)$$

formülü ile ifade edilir. Bu denklemdeki $F(x)$ minimizasyonu yapılacak amaç fonksiyonudur. Burada $x_i \in R^p$ 'nin i . değişkendir.

Kontrol uygulamaları için sürekli zaman gradyan iniş dinamiği

$$\frac{dx_i}{dt} = -\gamma_i \frac{\partial F}{\partial x_i} \quad (2.11)$$

diferansiyel denklemi ile ifade edilmektedir [17, 23, 46]. Bu denklemdeki $F(x)$ minimizasyonu yapılacak amaç fonksiyonudur. $x_i \in R$, $x_i \in R^p$ 'nin i . bileşenleridir. Bu denklem nümerik olarak çözümlendiğini zaman denklem (2.10)'daki nümerik gradyan iniş çözümü elde edilir. Bunu gösterelim: Bu denklemi Euler yöntemine göre çözmek için ileri fark denklemi [18, 46],

$$\frac{dx_i}{dt} \cong \frac{x_i[n+1] - x_i[n]}{\Delta t} \quad (2.12)$$

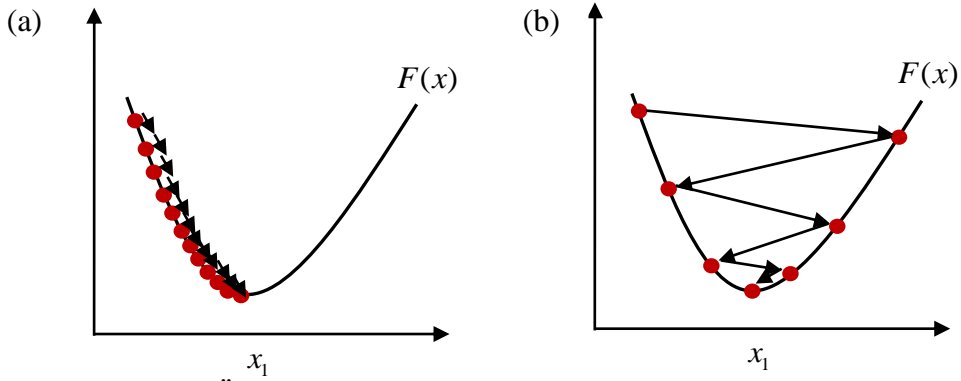
elde edilir. Denklem (2.12) denklem (2.11)'de kullanılır ve düzenlenirse,

$$x_i[n+1] - x_i[n] = -\Delta t \gamma_i \frac{\partial F}{\partial x_i} \quad (2.13)$$

elde edilir. $\Delta t \gamma_i$ katsayısı yerine η_i katsayısı yazılabilir,

$$x_i[n+1] = x_i[n] - \eta_i \frac{\partial F}{\partial x_i} \quad (2.14)$$

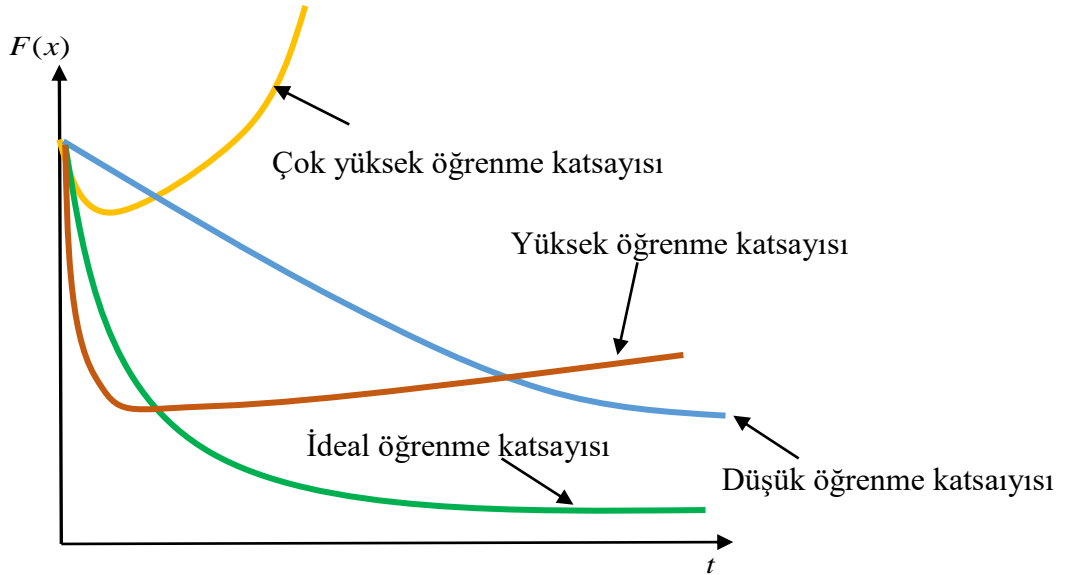
elde edilir. Burada η_i parametresi öğrenme katsayısı olarak adlandırılır. Burada x, η katsayısı ile tanımlanan küçük adımlarla $\frac{\partial F}{\partial x_i}$ 'in işaretine zıt bir yönde hareket ettirilerek daha düşük bir değere taşınır. Öğrenme katsayısı, adımın büyüklüğünü belirleyen skaler bir ifadedir [45]. Şekil 2.9'da öğrenme katsayısının değerine göre optimuma ilerleyiş biçimi 2 farklı grafik üzerinde gösterilmiştir.



Şekil 2.9 : Öğrenme katsayısı değerine göre optimum noktaya ilerleyiş.

η parametresi ne kadar düşük olursa, minimuma yaklaşırken aşağı doğru eğimde o kadar yavaş ilerlenir (Şekil 2.9 (a)). Bu durum herhangi bir yerel minimum noktasının kaçırılmasına fırsat vermezken, aynı zamanda yakınsama için uzun zaman alacağını gösterir. Büyük ayarlanırsa da yerel minimum noktayı aşabilir ve yakınsama başarısız olabilir (Şekil 2.9 (b)).

Genellikle öğrenme katsayıları kullanıcı tarafından rastgele belirlenir. Kullanıcının problem için en iyi katsayının ne olduğuna karar vermesi için geçmiş deneyimlerinden faydalanmak zorunluluğu vardır. Şekil 2.10'da öğrenme katsayısı yapılandırırken kullanıcın karşılaşılabileceği farklı senaryolar gösterilmektedir.



Şekil 2.10 : Çeşitli öğrenme katsayılarının yakınsama üzerindeki etkisi.

Öğrenme katsayısı fonksiyonun minimum noktasına ulaşma hızını etkiler. Bu nedenle doğru katsayısı seçimi önemlidir. Öğrenme katsayısını belirlemek için bir çok çalışma yapılmıştır. Yoshua Bengio "Neural Networks: Tricks of the Trade" kitabında

öğrenme oranı ayarlamasının önemini vurgulamış ve öğrenme katsayısı için uygulanabilir bir aralık belirlemeye çalışmıştır [47]. Breuel tarafından hazırlanan çalışmada da çeşitli hiperparametreler hakkında bilgiler sunulmuştur [48]. Leslie N. Smith modeli ilk başta düşük bir öğrenme katsayısı ile eğitip her iterasyonda bu katsayıyı arttırarak iyi bir öğrenme oranı elde edileceğini öne sürmüştür [49]. Daha sonra adaptif öğrenme katsayıları üzerine çalışmalar yapılmıştır [20, 50, 51, 52].

$F(x)$ hata fonksiyonu genelde makine öğrenmesi uygulamalarında öğrenme hatalarının karesel toplamı olarak ifade edilmektedir.

$$F(x) = \frac{1}{2} \sum_{j=1}^k e_j^2 \quad (2.15)$$

Bu denklemde k eğitim kümesindeki veri sayısıdır. j . veri için anlık öğrenme hatası şu şekildedir :

$$e_j = y_j - y_e(x[n]) \quad [12] \quad (2.16)$$

Gradyan iniş yöntemi ağırlık güncellemesinde kullanılan veri miktarına göre 3 farklı şekilde sınıflandırılmıştır [53].

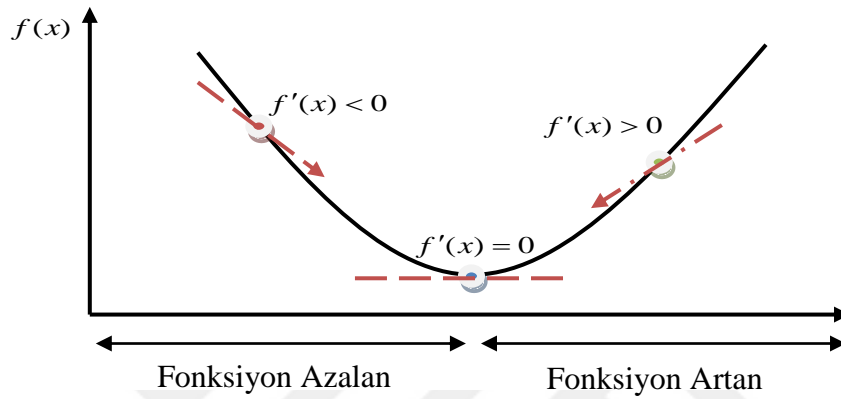
Tek güncellemede eğitim uzayındaki verilerin hepsinin kullanıldığı durum batch (yığın) gradyan iniş yöntemi olarak adlandırılır. Gradyan iniş yönteminin en basit çeşitidir. Tüm eğitim örneklerini değerlendirdikten sonra, model parametrelerini günceller. Batch gradyan iniş yöntemi verilerin hepsini kullandığı için yüksek işlem maliyeti oluşturur. Diğer bir dezavantaj da tüm eğitim setinin bellekte olmasını ve algoritma tarafından kullanılabilir olmasını gerektiren bir yöntemdir. Fakat yine aynı sebeple yani eğitim uzayındaki tüm verileri kullandığı için eğitim setinin öğrenilme kabiliyetini arttırır.

Her parametre güncellemesinde eğitim uzayında sadece bir verinin kullanıldığı durum stokastik gradyan iniş yöntemidir. Tüm eğitim örneklerini değerlendirdikten sonra parametrelerini güncelleyen batch gradyan iniş yönteminden farklıdır. Bu yöntemde işlem maliyeti düşüktür. Fakat batch gradyan iniş yönteminin tersine öğrenme kabiliyeti diğer iki yöntemle göre daha düşüktür. Güncellemelerin sık olması gürültülü inişlere neden olabilir ve hata oranının yavaşça azalması yerine dalgalanmasına sebep olabilir.

Ağırlık güncellemesi tüm eğitim uzayındaki veri kümesinden daha az bir veri kümesiyle gerçekleşiyorsa mini – batch (mini – yığın) gradyan iniş yöntemi denir. Eğitim setini küçük gruplar halinde ayırır ve bu grupların her biri için güncelleme gerçekleştirir.

Bu yöntem batch yönteminden daha az maliyetli ve stokastik yöntemden daha iyi öğrenme kabiliyetine sahiptir. [53]

Gradyan iniş metodunda eğimin + veya – olmasına bakılarak bir sonraki adımda minimizasyon için uygun konumun nereye taşınacağı hesaplanır. Yani bu metotta türev püf noktadır. Şekil 2.11’de de gösterildiği üzere x ’in konumuna göre minimum noktaya doğru taşınmasını sınıflandırırsak 3 grupta incelenebilir :



Şekil 2.11 : Başlangıç noktası seçimine göre minimum noktasına gitme güzergahı.

- 1- $\frac{\partial f(x)}{\partial x_n} > 0$ iken şekilde de görüldüğü üzere fonksiyon artandır ve türev değeri 0'dan büyüktür. Optimizasyonu gerçekleştirebilmek için ve yeni x konumunu güncelleyebilmek için x 'in azalan yönde hareket ettirilmesi gerekmektedir.

$$x_{n+1} = x_n - \gamma \frac{\partial f(x)}{\partial x_n} \quad (2.17)$$

$$x_{n+1} < x_n \quad (2.18)$$

- 2- $\frac{\partial f(x)}{\partial x_n} = 0$ iken şekilde de görüldüğü üzere minimum noktasına ulaşılmıştır. Bu noktada türev 0 'dır. x 'in konumunun değiştirilmesine gerek yoktur.

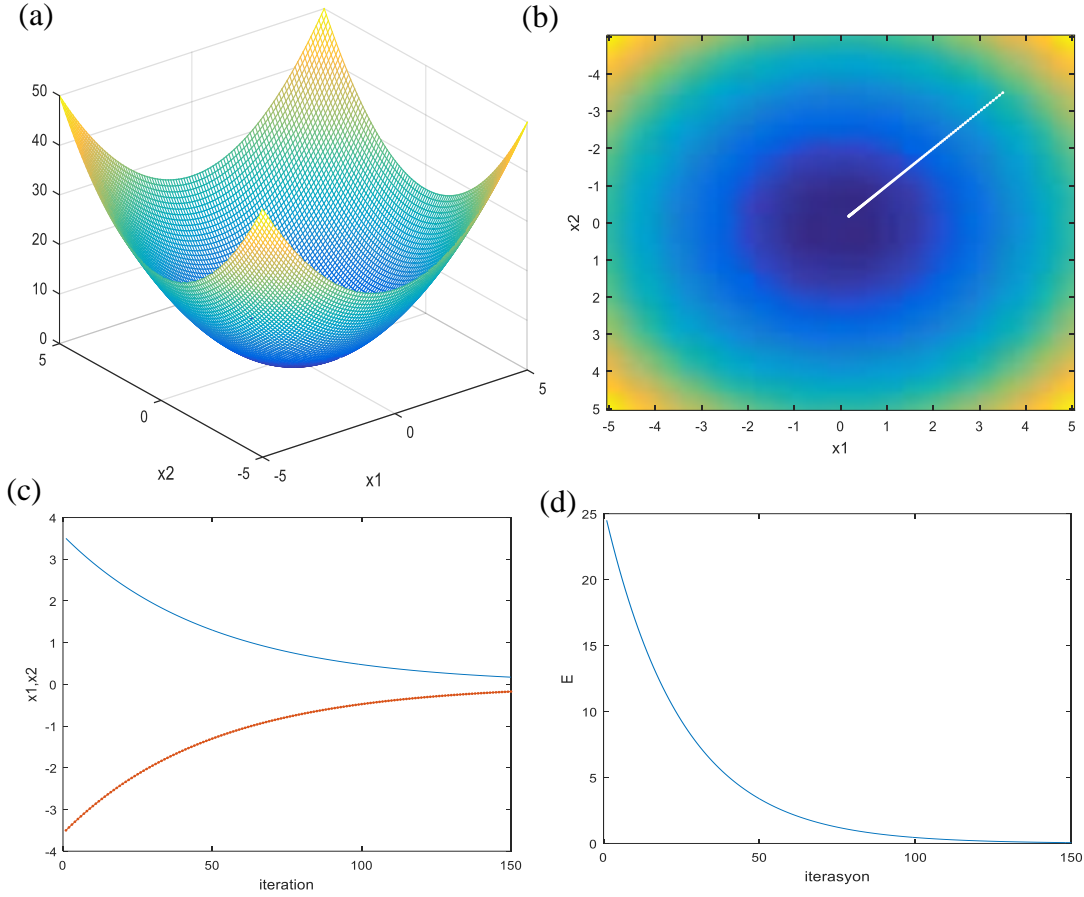
$$x_{n+1} = x_n \quad (2.19)$$

- 3- $\frac{\partial f(x)}{\partial x_n} < 0$ iken şekilde de görüldüğü üzere fonksiyon azalandır ve türev değeri 0 ' dan küçüktür. Optimizasyonu gerçekleştirebilmek için ve yeni x konumunu güncelleyebilmek için x 'in artan yönde hareket ettirilmesi gerekmektedir.

$$x_{n+1} > x_n \quad (2.20)$$

Aşağıda gradyan iniş algoritmasının iki test fonksiyonlarının üzerinde uygulaması incelenmiştir.

1- De Jong fonksiyonun optimizasyonu:

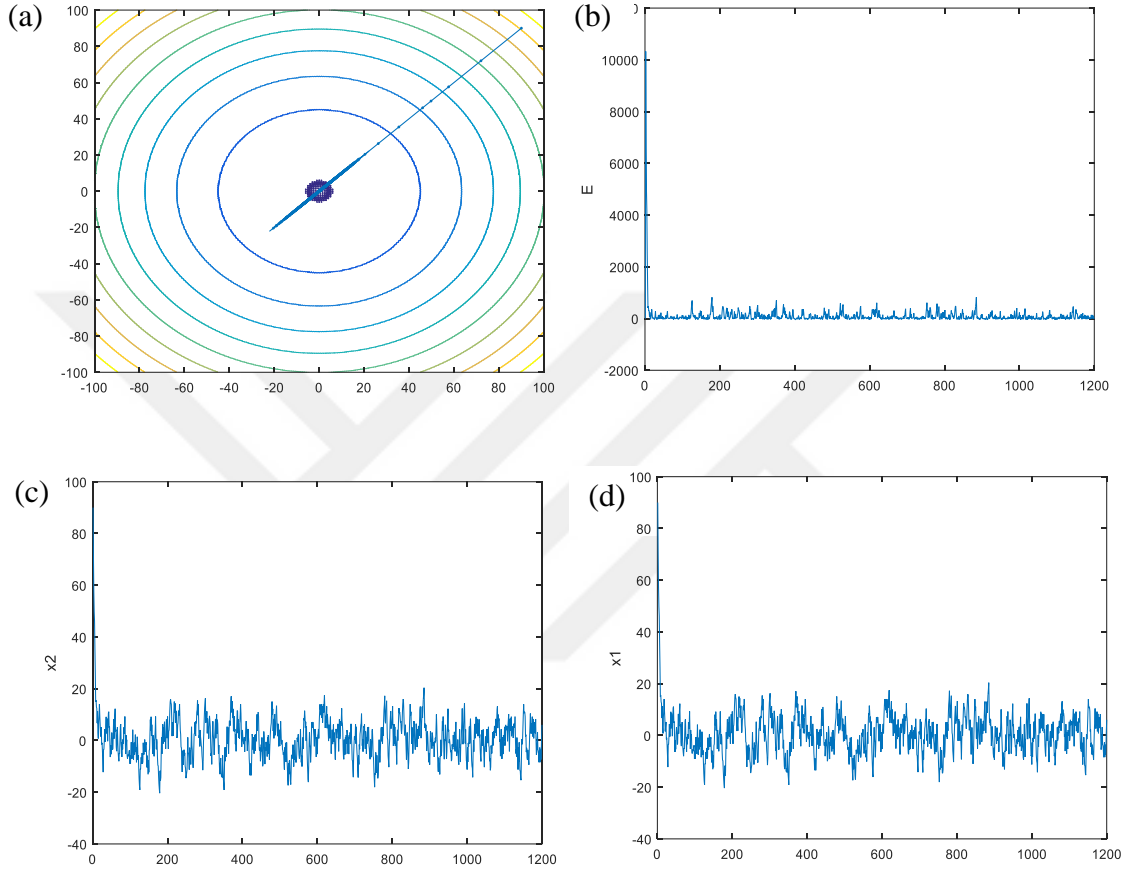


Şekil 2.12 : De Jong fonksiyonunun gradyan iniş yöntemi ile optimizasyonu; (a) 3 boyutlu De Jong fonksiyonu gösterimi, (b) Gradyan iniş yöntemi çözümlerinin De Jong fonksiyonu üzerinde ilerleyişi, (c) Optimizasyon sırasında x_1 ve x_2 değerlerinin değişimi, (d) Optimizasyon sırasında hata fonksiyonu değişimi

Şekil 2.12’de De Jong fonksiyonunun gradyan iniş metodu ile optimizasyonunda elde edilen sonuçlar görülmektedir. Şekil 2.12 (a)’da De Jong fonksiyonunun Matlab ortamında 3 boyutlu görüntüsü sunulmuştur. Bu görüntüden de anlaşıldığı üzere De Jong fonksiyonu unimodal yani tek optimum nokta içeren bir fonksiyondur. Bu yüzden optimizasyonu karmaşık değildir. Şekil 2.12 (b)’de De Jong fonksiyonuna gradyan iniş metodunun uygulanması sonucunda minimum noktaya hareketi beyaz çizgi ile gösterilmiştir. Burada gradya iniş yöntemi için seçilen başlangıç noktaları x_1 için 3.5 ve x_2 için -3.5 ’tir. Gradyan iniş metodu çözümlerinin başlangıç noktalarından fonksiyonun minimum noktasına ilerleyişi görülmektedir. Şekil 2.12 (c)’de belirlenen başlangıç

noktalarından minimum noktaya ilerlerken güncellenen x_1 ve x_2 değerlerindeki değişim görülmektedir. Şekil 2.12 (d)'de de optimize edilen hata fonksiyonu E 'deki optimizasyon boyunca yaşadığı değişimler sunulmuştur. Hatanın optimizasyon boyunca azalışı ve 0'a yaklaşması gözlenmektedir.

2- Rastrigin fonksiyonunun optimizasyonu:



Şekil 2.13 : Rastrigin fonksiyonunu gradyan iniş yöntemi ile optimize etme; (a) Gradyan iniş yöntemi ile optimize edilmiş Rastrigin fonksiyonu, (b) Optimizasyon sırasında hata fonksiyonu değişimi, (c) ve (d) x_1 ve x_2 değişimi.

Şekil 2.13'te Rastrigin fonksiyonunun gradyan iniş metodu ile optimizasyonunda elde edilen sonuçlar görülmektedir. Şekil 2.13 (a)'da Rastrigin fonksiyonunun Matlab ortamında gradyan iniş metodu çözümlerinin ilerleyişi sunulmuştur. Bu görüntüde gradyan iniş yöntemi için seçilen başlangıç noktaları x_1 için 3.5 ve x_2 için -3.5 'tir. Gradyan iniş metodu ile seçilen başlangıç noktalarından fonksiyonun minimum noktasına ilerleyişi görülmektedir. Şekil 2.13 (b)'de optimize edilen hata fonksiyonu E 'nin optimizasyon süresince değişimleri sunulmuştur. Şekil 2.13 (c) ve (d)'de belirlenen başlangıç noktalarından minimum noktaya ilerlerken güncellenen x_1 ve x_2 değerlerindeki değişimi görülmektedir.

Gradyan iniş yönteminin bazı avantajları ve dezavantajları vardır. En önemli avantajı, matematiksel yapısı basit ve kolay uygulanabilir olmasıdır. Şekil 2.12 ve şekil 2.13'te görüldüğü gibi optimum noktaya yakın bir civarından kolaylıkla yakınsayabilmektedir. Bu durum bazı durumlarda bir dezavantaja dönüşebilir. Çok sayıda lokal minimum olduğu durumda başlangıç noktasına yakın bir lokal minimuma takılabilir ve global minimuma ulaşamayabilir. Dolayısı ile performansının başlangıç noktasına bağımlılığı vardır. Bir diğer zayıflığı da öğrenme katsayısının uygun belirlenme sorunudur. Eğer katsayı büyük seçilirse çözüm kararsız olabilir ve yakınsama olmayabilir. Eğer çok küçük seçilirse minimuma yakınsama çok işlem adımı gerektirebilir ve optimizasyon süresi çok uzayabilir.

2.3 Gradyan İniş Algoritması Türleri

Gradyan iniş yöntemi, yöntemin kolaylığının ve zorluğunun ifade edilmesi zor olduğu için genellikle kara kutu optimize edici olarak adlandırılmaktadır [53]. Optimizasyon için en yaygın yollardan biridir ve sinir ağlarını optimize etmek için en popüler yöntemdir.

Gradyan iniş yönteminin en belirgin dezavantajı öğrenme katsayısının belirlenme zorluğudur. Çünkü öğrenme katsayısı küçük seçildiğinde, optimum değere yakınsama işlemi yavaş olacaktır. Öğrenme katsayısı büyük seçilirse optimum değerden uzaklaşma olabilir veya minimum değerler etrafında salınımlara yol açabilir.

Literatür incelendiğinde optimizasyon yöntemleri genelde parametrelerin daha hızlı bir şekilde güncellenmesine odaklanmıştır. Bazı çalışmalar parametrelerin daha hızlı güncellenmesi üzerinde yöntemler sunarken, bazı çalışmalar öğrenme katsayısı ile ilgili bahsedilen problemin çözümüne yönelik yöntemler sunmuştur [54, 55, 56, 52, 57, 58, 59, 60].

Bu bölümde gradyan iniş algoritmasının optimize edilmesine yardımcı olarak gradyan iniş algoritması varyantları sunulacaktır [53]. Sunulan bu varyantların Matlab ortamında kodlanarak sonuçların analizine yer verilmiştir. Bu algoritmalar aynı zamanda derin öğrenme uygulamalarında yaygın olarak kullanılmaktadır.

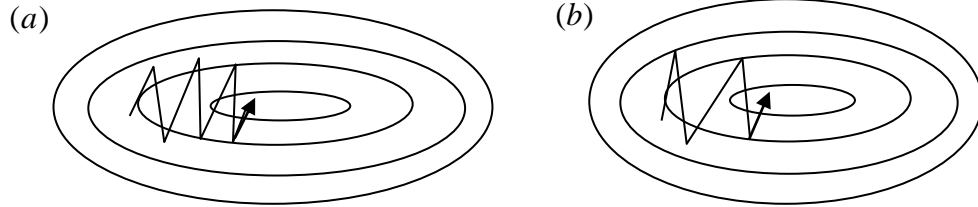
Bu yöntemlerden bazıları şunlardır: momentum, adagrad , rmsprop nesterovacceleratedgradient, adadelat, adam, adamax, nadam.

Bu bölümde bu yöntemlerden uygulamada yaygın kullanım bulabilen ilk üçü detaylı olarak incelenmiştir. Bu amaçla Matlab ortamında yöntemler kodlanmış De Jong fonksiyonu optimizasyonu probleminde performansları incelenmiştir:

a) *Momentum yöntemi:*

Momentum, inişin sürekli olarak aynı yönü gösterdiği boyutlar boyunca ilerlemenin hızlanmasıdır [58, 57, 53]. Bu algorithmda, optimize edilecek fonksiyon eğer vadi şeklinde düşünülürse, gradyan iniş metodu vadi tabanına geldiğinde yani yerel minimum noktasına yaklaştığı zaman yavaş ilerlemeye başlar. Vadi tabanında eğim, vadi duvarlarından daha az olduğu için vadi tabanında gradyan iniş metodunun optimizasyon noktasına ulaşması yavaş olur. Bu da Şekil 2.14 (a)'da gösterildiği gibi optimum noktaya ulaşana kadar birçok salınım olur. Bu salınımlar yakınsamaya ulaşmayı zorlaştırır ve öğrenme etkinliği yavaşlar. Bu nedenle ağırlıklar güncellenirken önceki değişimlerin de eklenmesi sağlanarak momentum etkisi yaratılır. Yani momentum metodu, parametreleri güncellerek sadece o iterasyondaki gradyanı değil geçmiş gradyanları da kullanır. Önceki değişimlerden kastedilen, yerel minimuma ilerlerken atılan adımlardan biriken momentumlar hız kazandırır ve atılacak adımları büyütür. Böylece Şekil 2.14 (b)'deki gibi az salınımlı, daha hızlı optimizasyon yapılmış olur.

Bu yöntem stokastik gradyan iniş yöntemini ilgili yönde hızlandırmaya yardımcı olan bir yöntemdir.



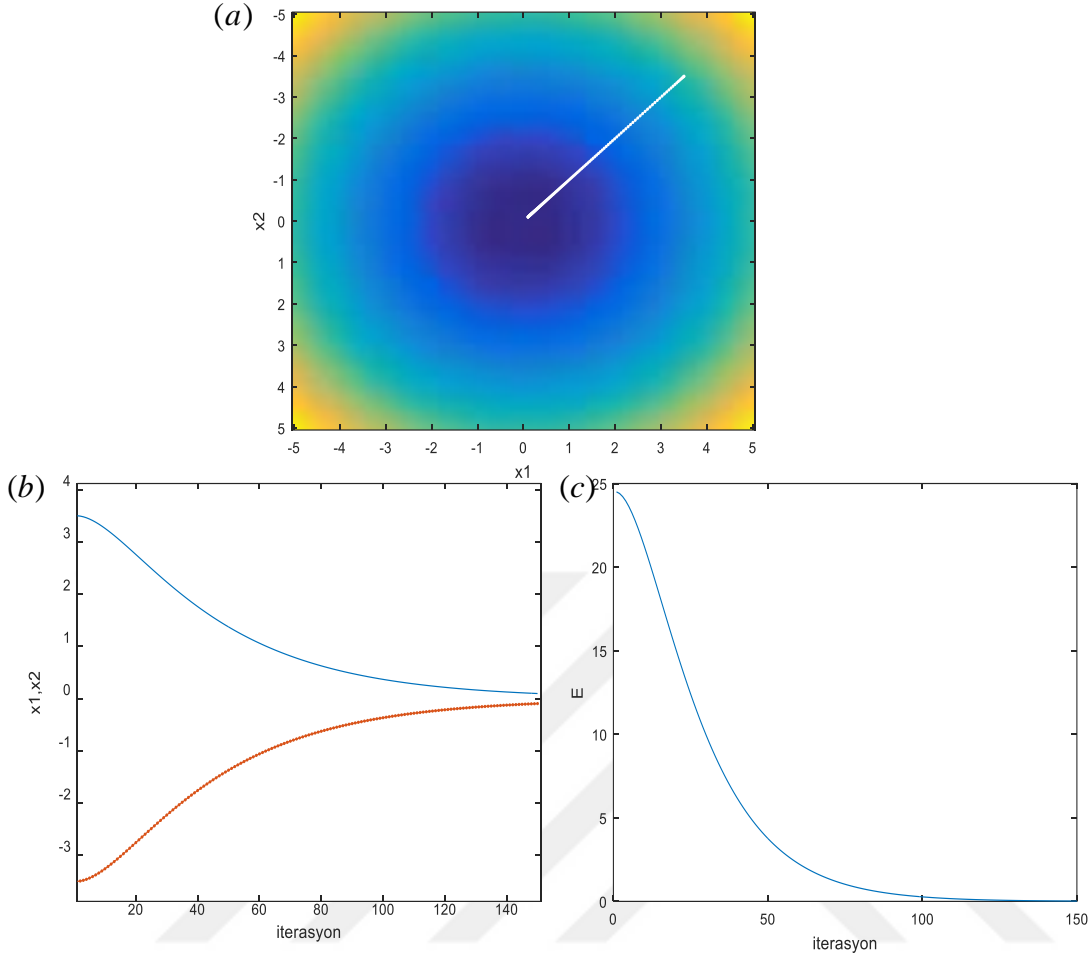
Şekil 2.14 : Gradyan iniş yöntemi yakınsama örneği, (a) momentum uygulanmamış, (b) momentum uygulanmış.

Formülasyon :

$$\mathcal{G}_t = \gamma \mathcal{G}_{t-1} + \eta \frac{\partial f(x)}{x_i} \quad (2.21)$$

$$x_{i+1} = x_i - \mathcal{G}_t \quad (2.22)$$

Denklem (2.21)'de \mathcal{G}_t ile ifade edilen gradyan iniş yönteminin optimum noktaya giderken kazandığı hız yani o konumdaki momentumdur. γ katsayısı momentum terimidir ve genellikle 0.9 veya yakın bir değerle ifade edilir. η 'da sıklıkla kullandığımız öğrenme katsayısıdır.



Şekil 2.15 : De Jong fonksiyonunun momentum yöntemi ile optimizasyonu; (a) Momentum yöntemi çözümlerinin De Jong fonksiyonu üzerinde ilerleyişi, (b) Optimazasyon sırasında x_1 ve x_2 değerlerinin değişimi, (c) Optimizasyon sırasında hata fonksiyonu değişimi.

Şekil 2.15'te De Jong fonksiyonunun momentum yöntemi ile optimizasyonunda elde edilen sonuçlar görülmektedir. Şekil 2.15 (a)'da De Jong fonksiyonunun üzerinde momentum yöntemi çözümlerinin ilerleyişi görülmektedir. Şekil 2.15 (b)'de optimizasyon sırasında x_1 ve x_2 katsayılarının değişimi görülmektedir. Bu görüntüde momentum yöntemi için seçilen başlangıç noktaları x_1 için 3.5 ve x_2 için -3.5 'tir. Momentum yöntemi ile seçilen başlangıç noktalarından fonksiyonun minimum noktasına ilerleyişi görülmektedir. Buna göre optimize edilen parametrelerin global minimum olan 0 sınır noktasına yaklaştığı görülmektedir. Burada öğrenme katsayısı $\eta = 0.001$, maksimum iterasyon sayısı 150, momentum katsayısı $\gamma = 0.9$ olarak alınmıştır. Şekil 2.15 (c)'de optimize edilen hata fonksiyonu E 'deki optimizasyon boyunca değişimi sunulmuştur. E 'deki değişimler gözlemlenirse, iterasyon sonuna doğru hatanın 0'a yaklaştığı açıkça görülmektedir.

b) *Adagrad yöntemi*:

Gradyan iniş metodunda öğrenme katsayısının kullanıcı tarafından belirlenmesi sorununu elimine etmek amacıyla iterasyonlar ilerledikçe öğrenme katsayısını düşürmektedir [61, 53]. Dolayısı ile her adımda öğrenme katsayısı da güncellenmektedir. Öğrenme katsayısının güncellenmesinde geçmiş gradyanlarının karelerinin toplamından yararlanır. Böylece eğitim süreci devam ettiği süre zarfında öğrenme katsayısı aşırı küçülmektedir [62].

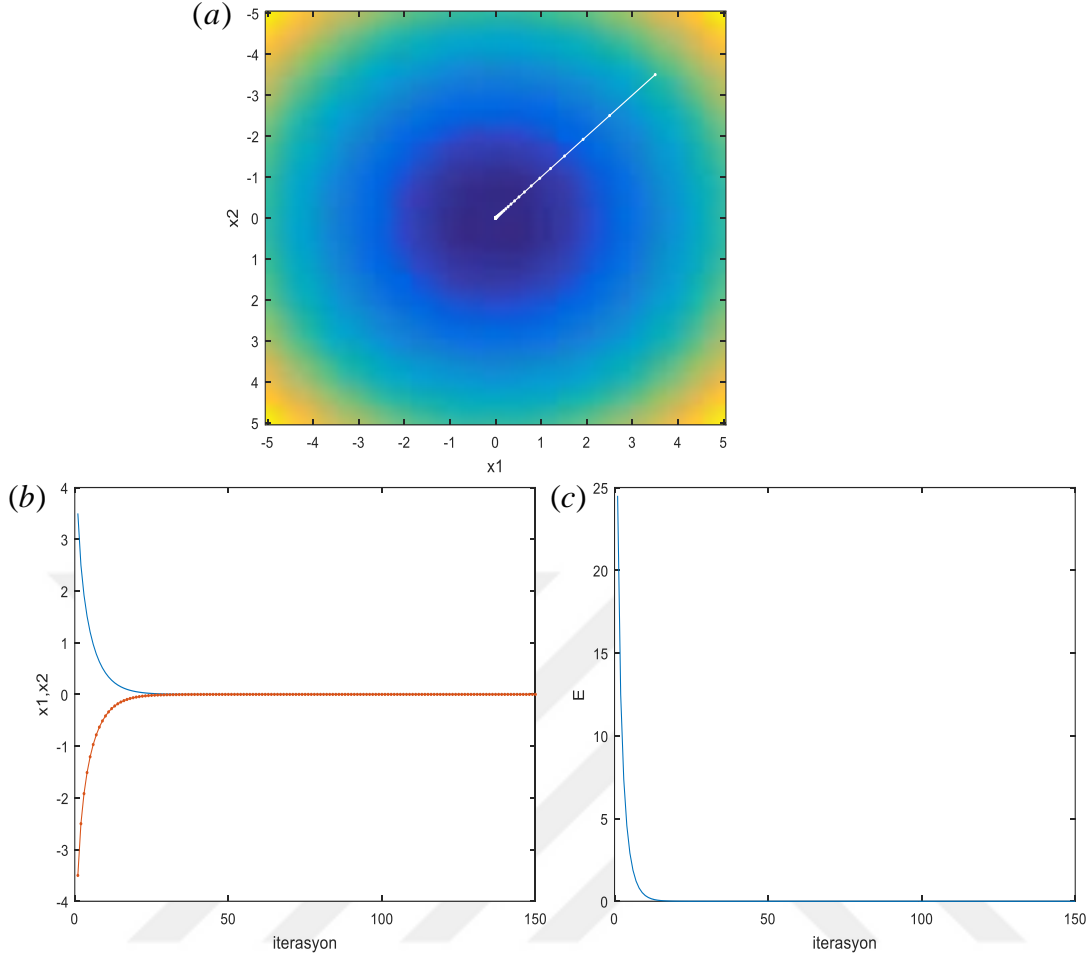
Formülasyon :

$$G_{toplam} = G_{toplam-1} + \left(\frac{\partial f(x)}{\partial x_i}\right)^2 \quad (2.23)$$

$$x_{i+1} = x_i - \frac{\eta}{\sqrt{G_{toplam-1} + \varepsilon}} \left(\frac{\partial f(x)}{\partial x_i}\right) \quad (2.24)$$

Denklem (2.23)'te G_{toplam} geçmiş gradyanların karesel toplamını ifade eder. G_{toplam} denklem (2.24)'te görüldüğü üzere öğrenme katsayısının (η) iyileştirilmesinde etkili olan parametrelerin önceki gradyanlarının karelerinin toplamıdır. η 'da sıklıkla kullandığımız öğrenme katsayısıdır. Paydada yer alan ε terimi önemlidir. Bu terim paydanın sıfır olmasını engellemek için kullanılır. Ve genelde 10^{-8} olarak ayarlanmaktadır.

Şekil 2.16'da De Jong fonksiyonunun adagrad yöntemi ile optimizasyonunda elde edilen sonuçlar görülmektedir. Şekil 2.16 (a)'da De Jong fonksiyonu üzerinde adagrad yöntemi çözümlerinin ilerleyişi sunulmuştur. Başlangıç noktasından minimum noktasına nasıl ulaştığı görülmektedir. Şekil 2.16 (b)'de optimizasyon sırasında x_1 ve x_2 katsayıların güncellenmeleri gösterilmiştir. Bu görüntüde adagrad yöntemi için seçilen başlangıç noktaları x_1 için 3.5 ve x_2 için -3.5 'tir. Optimizasyon sonucunda yani 150 iterasyon tamamlanınca adagrad yöntemi ile seçilen başlangıç noktalarından fonksiyonun minimum noktasına ilerleyişi görülmektedir. Uygulamada öğrenme katsayısı $\eta = 1$, maksimum iterasyon sayısı 150, $\varepsilon = 0.001$ olarak alınmıştır. Şekil 2.16 (c)'de optimize edilen hata fonksiyonu E 'nin iterasyon sürecinde azaldığı ve 0'a yaklaştığı görülmektedir.



Şekil 2.16 : De Jong fonksiyonunun adagrad yöntemi ile optimizasyonu; (a) Adagrad yöntemi çözümlerinin De Jong fonksiyonu üzerinde ilerleyişi, (b) Optimizasyon sırasında x_1 ve x_2 değerlerinin değişimi, (c) Optimizasyon sırasında hata fonksiyonu değişimi.

c) *Rmsprop yöntemi*:

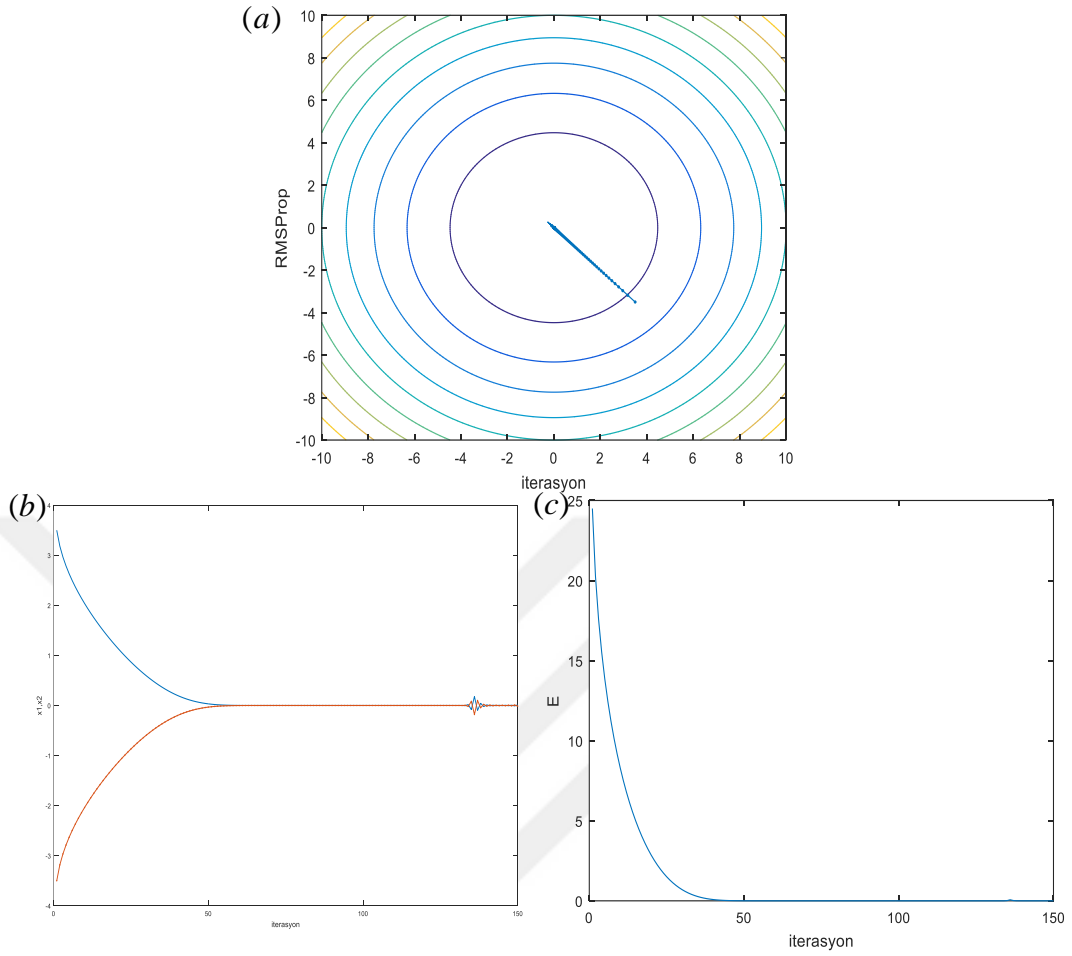
Geoff Hinton tarafından önerilen bir yöntemdir. Adagrad yöntemi gibi gradyanların karesel toplamını direk kullanmaz. Bunun yerine gradyanların karesel toplamını belli bir oranda kullanmaktadır [53, 63]. Adagrad algoritmasındaki öğrenme katsayısının belli bir iterasyondan sonra çok fazla küçülmesi sorununa karşı geliştirilmiştir. Belli oranlarda çerçevenmiş gradyanların karesel toplamı,

$$G_{toplama} = 0.9G_{toplama-1} + 0.1\left(\frac{\partial f(x)}{\partial x_i}\right)^2 \quad (2.25)$$

Ağırlıkların güncellenmesi ise ;

$$x_{i+1} = x_i - \frac{\eta}{\sqrt{G_{toplama-1} + \epsilon}} \left(\frac{\partial f(x)}{\partial x_i}\right) \quad (2.26)$$

Formülü ile ifade edilmektedir.

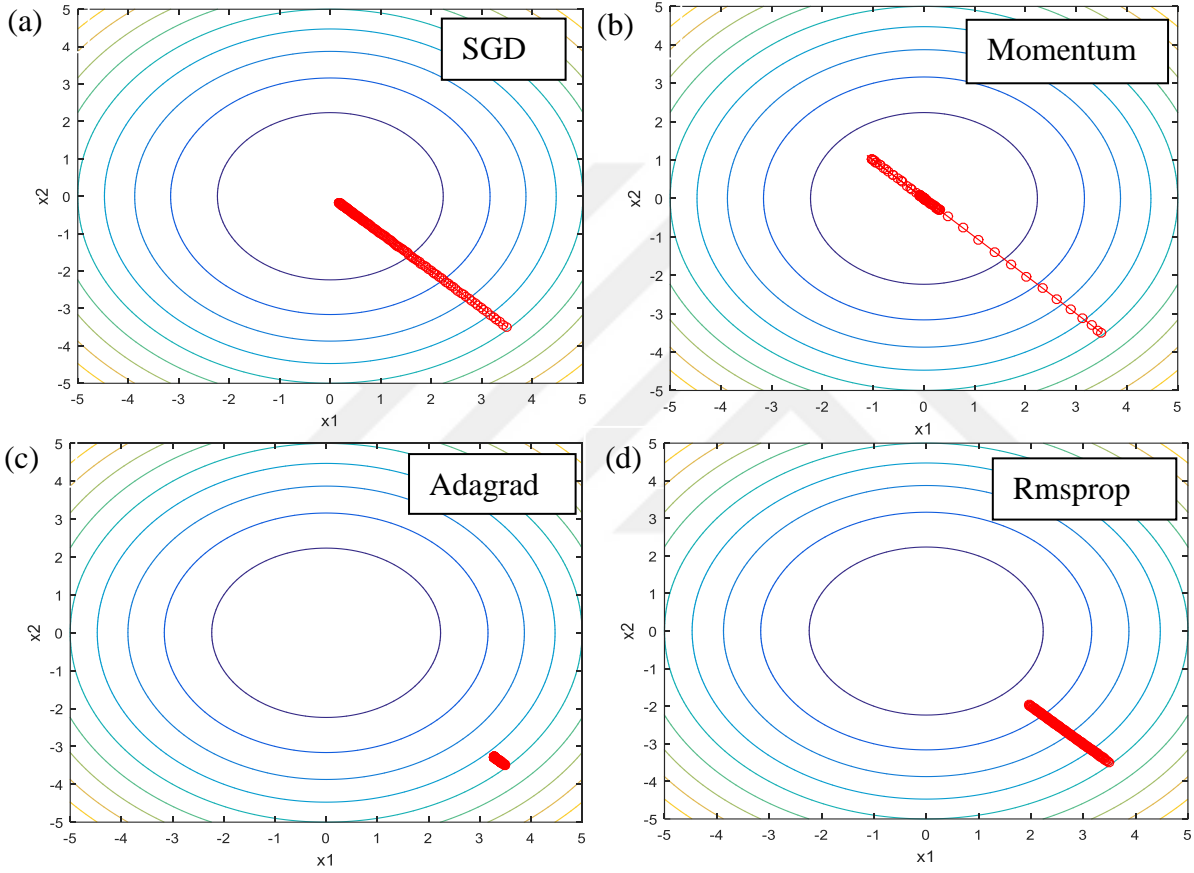


Şekil 2.17 : De Jong fonksiyonunun rmsprop yöntemi ile optimizasyonu; (a) Rmsprop yöntemi ile optimize edilmiş De Jong fonksiyonu üzerinde çözüm noktalarının ilerleyişi, (b) Optimizasyon sırasında x_1 ve x_2 değerlerinin değişimi, (c) Optimizasyon sırasında hata fonksiyonu değişimi.

Şekil 2.17’de De Jong fonksiyonunun rmsprop yöntemi ile optimizasyonunda elde edilen sonuçlar görülmektedir. Şekil 2.17 (a)’da De Jong fonksiyonunun rmsprop yöntemi uygulanması sonucu çözüm noktalarının ilerleyişi görülmektedir. Başlangıç noktasından minimum noktasına nasıl ulaştığı gösterilmiştir. Şekil 2.17 (b)’de optimizasyon sırasında çözüm noktaları x_1 ve x_2 katsayıların güncellenen değerleri görülmektedir. Bu görüntüde rmsprop yöntemi için seçilen başlangıç noktaları x_1 için 3.5 ve x_2 için -3.5’dir. Uygulamada öğrenme katsayısı $\eta = 0.1$, maksimum iterasyon sayısı 150, $\varepsilon = 0.001$ olarak alınmıştır. Şekil 2.17 (c)’de optimize edilen hata fonksiyonu E ’nin iterasyon sürecinde azaldığı ve 0’a yaklaştığı görülmektedir.

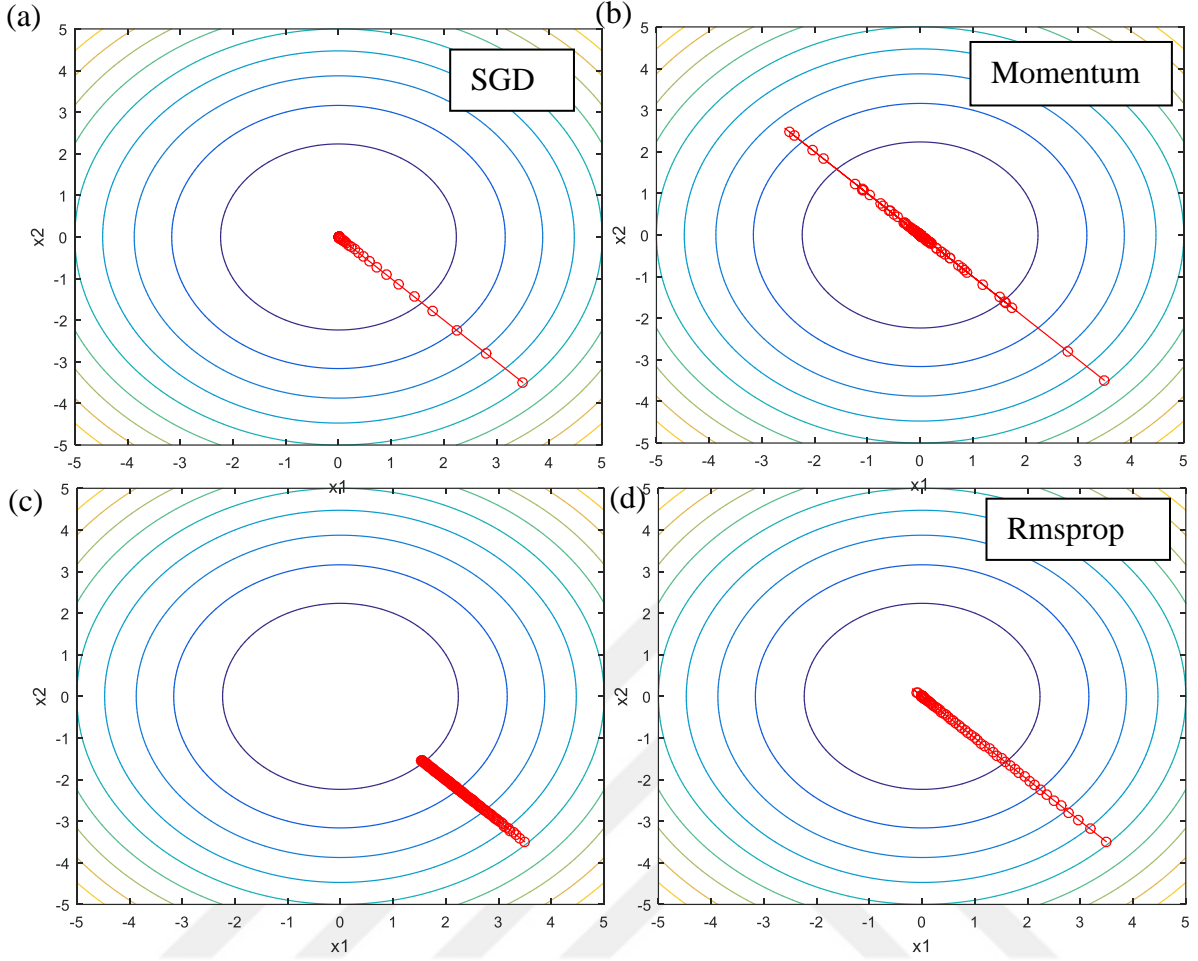
2.3.1 Algoritmaların kıyaslanması

Bu bölümde, bir önceki bölümde incelenen stokastik gradyan iniş yöntemi, momentum yöntemi, adagrad yöntemi, rmsprop yöntemleri karşılaştırılmıştır. Karşılaştırma için yöntemler Matlab ortamında kodlanmış ve De Jong fonksiyonunu optimizasyonunda test edilmiştir. Bütün yöntemlerde aynı öğrenme katsayısı kullanılmıştır. Öğrenme katsayısının 0.01 (Şekil 2.18), 0.1 (Şekil 2.19), 0.2 (Şekil 2.20), 0.5 (Şekil 2.21) ve 1 (Şekil 2.22) değerleri için testler yapılmıştır.



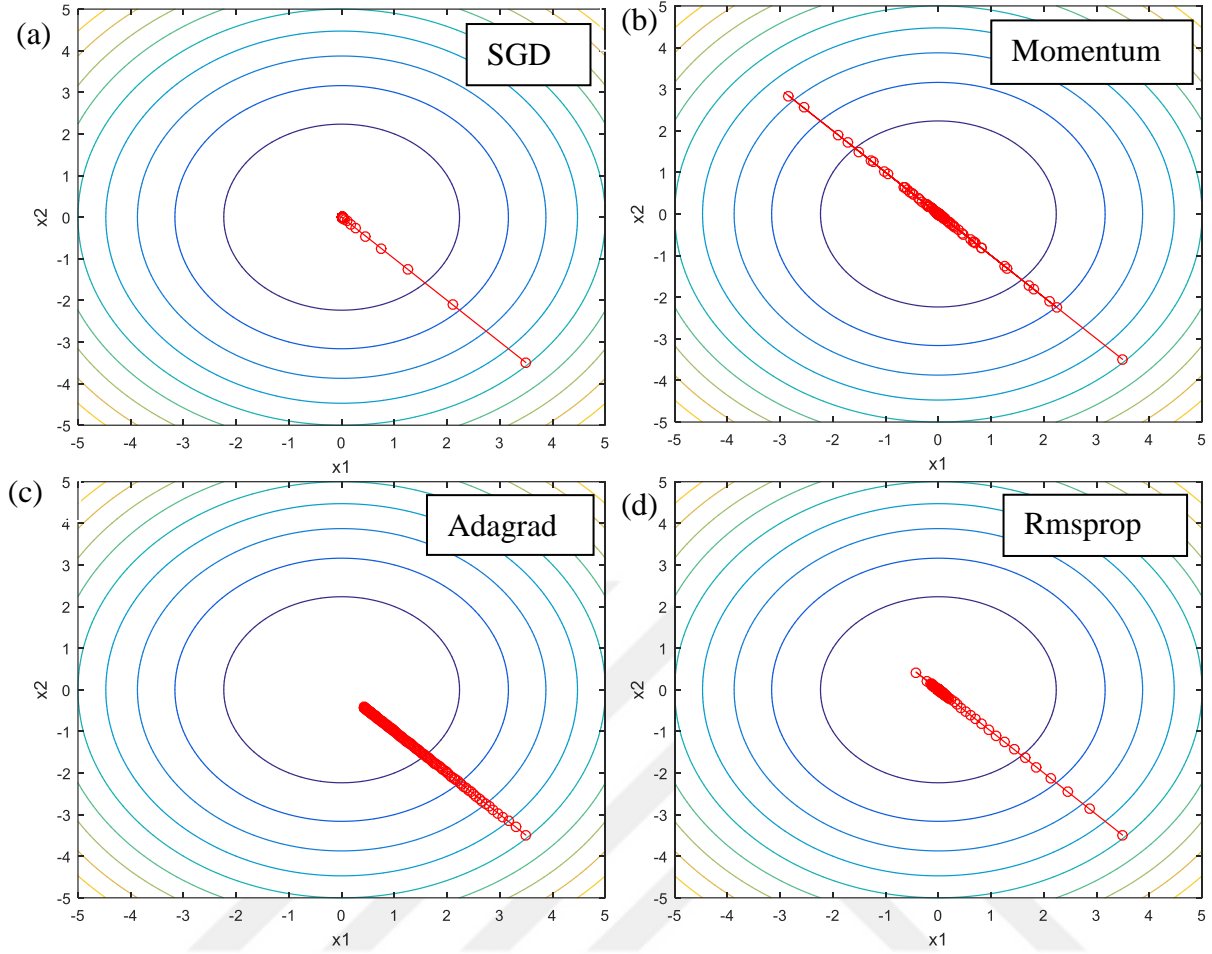
Şekil 2.18 : Öğrenme katsayısı $\eta = 0.01$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.

Şekil 2.18'de öğrenme katsayısı $\eta = 0.01$ iken stokastik gradyan iniş (SGD) yöntemi ve momentum yönteminin minimum noktaya ulaştığını fakat diğer yöntemlerin ulaşamadığı görülmektedir. Momentum yöntemi SGD yöntemine göre daha büyük adımlar atarak minimum noktaya daha hızlı yaklaşmıştır. Fakat öğrenme katsayısı büyüklüğü nedeniyle minimum noktası etrafında ileri geri sıçramalar olmuştur. Ancak adagrad ve rmsprop yöntemleri için verilen öğrenme katsayısı minimum noktaya ulaşmak için yeterli olmamıştır.



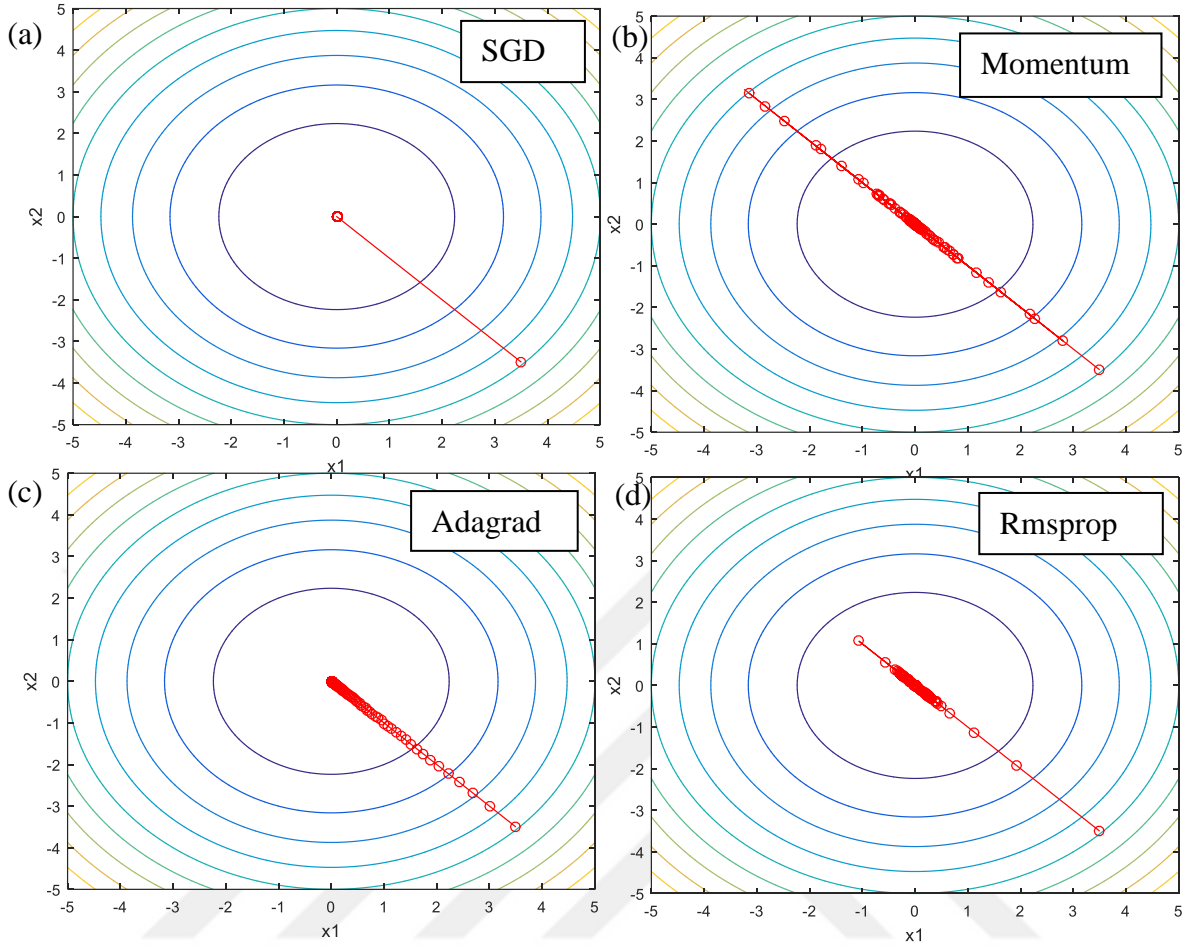
Şekil 2.19 : Öğrenme katsayısı $\eta = 0.1$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.

Şekil 2.19’te öğrenme katsayısı artırılarak $\eta = 0.1$ iken tekrar simülasyonlar gerçekleştirilmiştir. Simülasyonlara göre Stokastik gradyan iniş yöntemi öğrenme katsayısı arttığı için minimum noktaya daha büyük adımlar atarak daha hızlı ulaşmıştır. Momentum yöntemi de yine minimum noktaya daha hızlı yaklaşmıştır ancak minimum etrafında salınımları artmıştır. Adagrad yöntemi için verilen öğrenme katsayısı minimum noktaya ulaşmak için yeterli olmamıştır. Rmsprop için de $\eta = 0.1$ öğrenme katsayısı minimum noktaya ulaşabilmek için yeterlidir fakat iterasyon adımları küçük olduğu için yakınsama yavaştır.



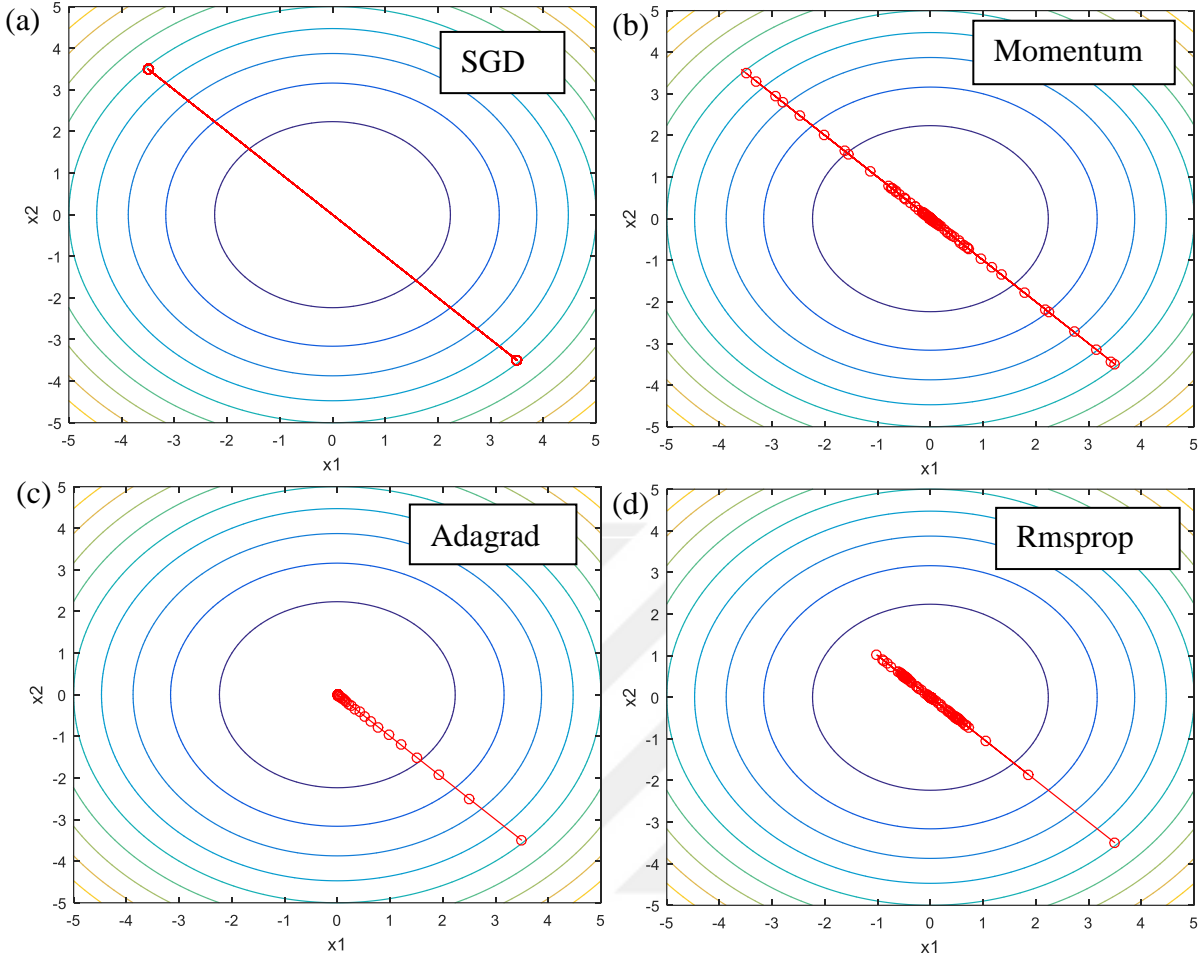
Şekil 2.20 : Öğrenme katsayısı $\eta = 0.2$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.

Şekil 2.20’de öğrenme katsayısı artırılarak $\eta = 0.2$ iken tekrar simülasyonlar gerçekleştirilmiştir. Simülasyonlara göre stokastik gradyan iniş yöntemi minimum noktaya daha büyük adımlarla daha hızlı ulaşmıştır. Momentum yöntemi de yine aynı şekilde Şekil 2.19 (b)’ye göre minimum noktaya daha hızlı yaklaştığı ve öğrenme katsayısı büyüklüğü sebebiyle minimum noktası yakınlarında ileri geri hareket ettiği görülmektedir. $\eta = 0.2$ için adagrad yöntemi minimum noktaya ulaşmıştır fakat öğrenme katsayısı adagrad yönteminin iyi performans sağlaması için yeterli değildir. Rmsprop yöntemi Şekil 2.19 (d)’ye göre daha iyi performans göstermiş, minimum noktaya daha hızlı yaklaşmıştır.



Şekil 2.21: Öğrenme katsayısı $\eta = 0.5$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.

Şekil 2.21’de öğrenme katsayısı artırılarak $\eta = 0.5$ iken tekrar simülasyonlar gerçekleştirilmiştir. Simülasyonlara göre SGD yöntemi minimum noktaya tek adımda ulaşmıştır. Yani $\eta = 0.5$ öğrenme katsayısı SGD yöntemi için ideal öğrenme oranıdır. Momentum yöntemi için de şekilde Şekil 2.20 (b)’ye göre minimum noktaya daha hızlı ulaştığı görülmektedir. Adagrad yönteminin Şekil 2.20 (b)’ye göre minimum noktaya ulaşma performansı daha iyidir. Rmsprop yöntemi Şekil 2.20 (d)’ye göre çok iyi bir performans göstermiş, minimum noktaya daha hızlı ve daha büyük adımla yaklaşmıştır. Öğrenme katsayısının yüksek olması nedeni ile minimum noktası civarlarında salınarak yaklaşmıştır.



Şekil 2.22 : Öğrenme katsayısı $\eta = 1$ iken (a) Stokastik gradyan iniş yöntemi, (b) Momentum yöntemi, (c) Adagrad yöntemi, (d) Rmsprop yöntemi sonuçları.

Şekil 2.22’de öğrenme katsayısı $\eta = 1$ iken tekrar simülasyonlar gerçekleştirilmiştir. Simülasyonlara göre SGD yöntemi için öğrenme katsayısı minimum noktaya yakınsayamadan salınmaya başlamış ve SGD çözümü kararsız hale gelmiştir. SGD minimuma daha fazla yaklaşamamıştır. Bu öğrenme katsayısı SGD için çok yüksektir ve bu nedenle SGD çözümü kararsızdır. Momentum yöntemi için de Şekil 2.21 (b)’ye göre minimum noktaya daha fazla salınımda ulaşabildiği görülmektedir. Adagrad yönteminin $\eta = 1$ için diğer değerlere göre minimum noktaya ulaşma performansının daha iyileştiği görülmüştür. Minimum noktaya daha hızlı ulaşılmıştır. Rmsprop yöntemi için katsayı büyüklüğünden dolayı minimum noktası etrafında salınım hareketleri artmıştır. Şekil 2.21 (d)’ye göre daha büyük adımlar ile yakınsama sağlanmıştır. Bu incelemede öğrenme katsayısının uygun seçiminin incelenen bu dört gradyan iniş yöntemi türü için önemli olduğu ve çözümün kararlılığının ve yakınsamsının bu parametreye bağlı olduğu görülmüştür.

2.4 Kontrol Sistemlerinin Temelleri

Bilim ilerledikçe insan gücüne olan ihtiyaç azalmaktadır. Fabrikalar örnek olarak verilecek olursa kas gücünün yerini makineler almaktadır. Kontrol ve otomasyon mühendisliği, bu sistemlerin insana ihtiyaç duymadan kontrol edilmesi üzerinde çalışır. Böylece üretim hızı, verimlilik, ürün kalitesi anlamında avantajlar salanır.

Sistem belirli bir iş için bir araya getirilmiş birbirleriyle etkileşim halinde olan birimler topluluğudur. Kontrol sistemi herhangi bir işi yapan eleman topluluğunun denetlenmesi için oluşturulmuş sistemlerdir [64]. Bir sistemin çıkış durumunu istenen değere getirme işini yapar. Günlük hayatta yapılan işlemlerin çoğunun sonucu kontrol sistemlerine dayanmaktadır [64]. Bir kontrol sisteminin temel öğeleri şu maddelerle açıklanmaktadır [65] :

- a) Kontrol girişi
- b) Kontrol sistemi öğeleri
- c) Sonuç ya da çıkışlar.

Bu öğelerin ilişkisi aşağıda gösterilmektedir.



Şekil 2.23 : Kontrol sistemi temel öğeleri.

Giriş, sisteme girdi olarak alınan ve işlenen birimlerdir.

Çıkış, sistemden çıkan, işlenmiş büyüklüklerdir.

Örneğin, bir kontrol sisteminin girişi olan işaret elektrik enerjisiyse, çıkış işareti olarak mekanik enerjisi üretilebilir. Diğer bazı kontrol sistemi uygulamaları için otomobillerde hız kontrolü, sıcaklık kontrolü, endüstriyel dikiş makinası kontrolü, güneş enerji sistemlerinde güneş izleme kontrolü uygulamaları kontrol sistem uygulamalarına örnek olarak verilebilir [65].

Kontrol sisteminin genel amacı girişleri değerlendirerek çıkış işaretlerini kontrol etmektir [65]. Yani kontrol sistemlerinin ana felsefesi gerçek sistem çıkışının ($Y_s(t)$),

istenen sistem çıkışına ($Y_d(t)$) değerine yakınsamasını sağlamaktır. Bu yakınsamayı ölçmek için bir hata işareti ($e(t)$) tanımlanır. Bu işarete kontrol hatası adı verilir ve

$$e(t) = Y_d(t) - Y_s(t) \quad (2.27)$$

ile ifade edilir. Kontrol sistemlerinde amaç [66]

$$\begin{aligned} e(t) &\cong 0 \\ Y_s(t) &\cong Y_d(t) \end{aligned} \quad (2.28)$$

ile ifade edilebilir. Bir kontrol sisteminde kontrol işleminin başarılı olması en temel anlamda [66]

$$\lim_{t \rightarrow \infty} |e(t)| = 0 \quad (2.29)$$

ile formülize edilebilir. Bu denkleme göre kontrol hatası zamanla sifıra yakınsamalıdır.

Pratik kontrol sistemlerinin şu özelliklere sahip olması beklenir [66]:

- a) Değişen çevre koşulları ve tahmin edilemeyen bozucu etkiler kontrol sistemlerinin performansını etkiler. Sistem bozucu etkilere karşı dayanıklı olmalı ve kontrol hatasını yeterince küçük tutabilmelidir.
- b) Gerçek hayat sistemlerinin dinamikleri bire bir modellenemez. Burada modelleme ve hesaplama yöntemlerinin limitleri önemli bir rol oynar. Dolayısı ile matematiksel modellere göre geliştirilen kontrol sisteminin pratik performansı düşer.

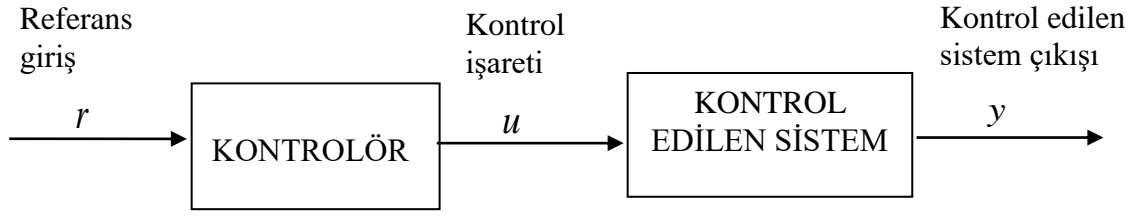
2.4.1 Kontrol sistemi çeşitleri

Kontrol sistemleri sistemlerin çalışma tipine göre 2 'ye ayrılır :

2.4.1.1 Açık çevrim kontrol sistemi (Geri beslemesiz sistemler)

Sistemi kontrol eden düzenek sistemin çıkışından etkilenmez. Günlük hayatta yaygın olarak kullanılır. Anahtar kullanarak lambayı yakmak en basit örneğidir. Açık çevrim kontrol sistemi bozucu etkileri azaltamaz. Çıkışın sadece girişin bir fonksiyonu olduğu bu sistemlerde sistem çıkışının kontrol işleminde hiçbir etkisi yoktur.

Açık çevrim kontrol sistemi Şekil 2.24'te görüldüğü gibi 2 kısıma ayrılır [65].

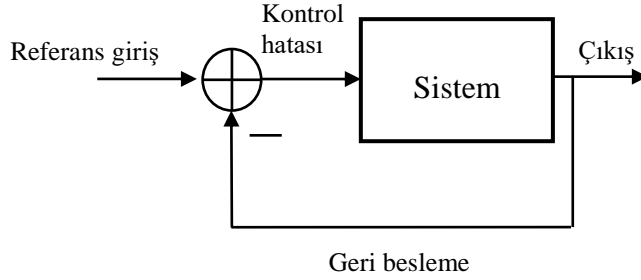


Şekil 2.24: Açık çevrim kontrol sistemlerinin öğeleri.

Burada u çıkışlı kontrolöre r giriş işareti uygulanmış ve bu u işareti, sistem çıkışındaki kontrol edilen y işaretini, çıkışta istenen şekilde davranmasını sağlayacak biçimde etkiler. Basit ve ekonomik oldukları için bu tarz açık çevrimli kontrol sistemleri, karmaşık olmayan çalışmalarda kullanılır.

2.4.1.2 Kapalı çevrim kontrol sistemi (Geri beslemeli sistemler)

Kontrol edilen sistem çıkışından geri besleme alınan ve bu geribeslemeye göre kontrol sağlayan sistemlerdir. Negatif geribeslemeli kapalı çevrim sistemler kontrol sistemi çıkışının referans girişe yakınsayabilmesi için kontrol hatasını kontrol işleminde kullanırlar. Böylece kontrol hatasını sifıra götürebilen bir kapalı çevrim kontrolör tasarlanabilir [66].



Şekil 2.25 : Negatif geribeslemeli kapalı çevrim kontrol sistemi.

Şekil 2.25'te çıkış, giriş ve geri besleme işaretleri gösterilmiştir. Burada sistem kontrolör ve kontrol edilen sistemi temsil etmektedir.

2.4.2 Sistem modellemesi ve kontrol

Bir sistemin kontrol edilebilmesi için öncelikle sistemin davranışını temsil edebilen matematiksel modelle gereksinim vardır. Doğadaki dinamik sistemler diferansiyel denklemler ile modellenirler. Diferansiyel denklemler ile modellenen dinamik sistemler için kontrolör tasarımı temelde iki bölgede yapılır:

a) Frekans domeni tasarımı: Diferansiyel denklemler ile ifade edilen matematiksel modeller Laplace dönüşümü ile frekans domeninde ifade edilir. Klasik kontrol yaklaşımları genelde frekans bölgesi tasarımlarını konu alır.

b) Zaman domeni (Modern kontrol yaklaşımları) : Diferansiyel denklemler durum-uzayı modelleri yardımı ile daha basit bir formda temsil edilir. Modern kontrol yaklaşımları genelde frekans bölgesi tasarımlarını konu alır.

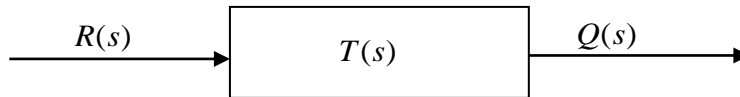
Diferansiyel denklemler,

$$F(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \frac{d^3y}{dx^3}, \dots, \frac{d^n y}{dx^n}) = 0 \quad (2.30)$$

genel denklemi ile ifade edilir. Diferansiyel denklemlerinin çözümleri diferansiyel denklem türlerine göre farklılık gösterir. Bu nedenle durum-uzayı modelleri veya frekans bölgesi modelleri ile kontrol sistemi tasarımı yapılır. Ayrıca kontrol sistemleri doğrusal (linear) veya doğrusal olmayan (nonlinear) olarak iki ayrı gruba ayrılır. Bu durum kontrol sistemi tasarım yöntemlerinde farklılaşmasına yol açar. Kontrol sistemi tasarımında hedeflenen mümkün olduğunca az model bağımlı ve adaptif kontrol sistemleri tasarımıdır. Adaptif sistemler bulunduğu koşullara kendi kendini adapte olma (uyarlanma) kabiliyetine sahip olan sistemleri ifade eder ve pratik kontrol uygulamaları açısından adaptif kontrol öneme sahiptir.

2.4.3 Transfer fonksiyonları

Transfer fonksiyonu, lineer sistemde sistem girişi ile çıkışı arasındaki matematiksel ilişkinin Laplace dönüşümüdür. Bu model frekans domeninde lineer sistemleri temsil etmek kullanılan bir model türüdür. Açık çevrim ve kapalı çevrim sistemlerin transfer fonksiyonlarını elde edelim.

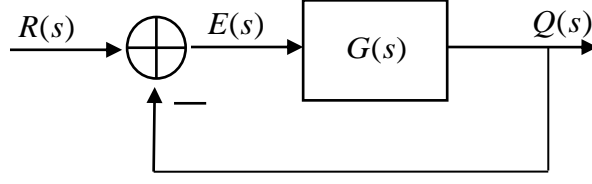


Şekil 2.26 : Açık çevrim kontrol sistemi blok diyagram gösterimi

Şekil 2.26'da görülen açık çevrim sistem için transfer fonksiyonu

$$T(s) = \frac{Q(s)}{R(s)} \quad (2.31)$$

denklemleri ile ifade edilir.



Şekil 2.27 : Kapalı çevrim kontrol sistemi blok diyagramı gösterimi

Şekil 2.21'de görülen kapalı çevrim sistem için transfer fonksiyonu

$$\begin{aligned} Q(s) &= G(s)E(s) \\ E(s) &= R(s) - Q(s) \\ C(s) &= G(s)[R(s) - Q(s)] \end{aligned} \quad (2.32)$$

blok diyagramına göre yazılabilir. Gerekli düzenlemeler yapılırsa,

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)} \quad (2.33)$$

elde edilir.

Transfer fonksiyonları özellikle sistem kararlılık analizlerinde kullanım bulur. Transfer fonksiyonun paydası sistemin karakteristik denklemdir ve köklerinin yerleşimi ile kararlılık analizleri frekans domeninde (s bölgesinde) kolaylıkla yapılabilmektedir.

2.4.4 Kararlılık nedir?

Kararlılık, bir sistem bozucu etkilere karşı, geçici hal davranışı kazandıktan sonra denge durumuna geri dönüyorsa o sistem kararlıdır. Eğer sistemin bozucu etkilere cevabı sürekli artıyorsa yani büyükten genlikli bir sinyal şeklindeyse o sistem kararsızdır.

Lineer kontrol sistemlerinin kararlılığı, kapalı çevrim transfer fonksiyonlarının kutuplarından diğer bir ifade ile karakteristik denklem köklerinden bulunmaktadır. Kararlı bir sistem için karakteristik denklem köklerinin hepsi s düzleminin sol yarısında olmalıdır. s düzleminin sağ tarafında kalan tek bir kök, sistemi kararsız hale getirir. Sanal eksen üzerinde eşlenik kökler sabit genlikli titreşim cevap verir. Bu durumda sistem kararlılık sınırındadır. Özetlemek gerekirse karakteristik köklerin yerleşimi lineer sistemin asimptotik kararlılık konusunda fikir verir.

2.5 Lyapunov Kararlılık

Diğer bir sistem kararlılık durumu sınırlı - giriş ve sınırlı - çıkış kararlılığı durumudur. Bir sistem girişine sınırlı bir giriş değeri uygulandığında çıkışı sınırlı kalabiliyorsa (sonlu bir değerde) bu sistem sınırlı - giriş sınırlı çıkış kararlıdır. Zaman domeninde kararlılık analizi nispeten zorluklar içerir. Dinamik sistemi ifade eden diferansiyel denklemlerin çözümüne ihtiyaç duyulur. Bu noktada, 1980 yılında Rus matematikçi Lyapunov, “The General Problem of Motion Stability” adlı kitabında doğrusal olmayan sistemlerin kararlılığının incelenmesi konusundaki yaklaşımlarını yayınladı. Lyapunov’un diferansiyel denklem çözmeden yani karmaşık matematiksel formüllerle uğraşmadan kararlılık analizi yapan bu çözüm, doğrusal olmayan sistemler için en genel metottur [23, 67, 68, 69, 70]. Lyapunov kararlılığı, gradyan iniş dinamiğinin kararlılığında [18, 68], adaptif kontrol sistemleri kararlılığında [23, 71, 72] , modelleme [73, 74] gibi alanlarda kullanılmaktadır.

Lyapunov’un bu yöntemi ikiye ayrılmaktadır. İlki, sistemin doğrusallaştırılmasını gerektiren dolaylı yöntemdir. İkincisi ise, doğrudan yöntem olarak adlandırılır ve Lyapunov fonksiyonlarıyla, doğrusal olmayan sistemlerin kararlılık özelliklerinin belirlenmesi yöntemidir. Lyapunov kararlılık kriteri, enerjisi azalan sistemlerin zamanla kararlı hale geldiği prensibine dayanır. Enerji fonksiyonun pozitif ve sürekli azalan bir fonksiyon olmasını şart koşar. Lyapunov kararlılık analizinin kararlılık uygulamalarında tercih edilmesinin nedenleri önceki bölümde anlatıldığı gibi s domeninde yapılan kararlılık analizleri doğrusal sistemler için tercih edilmektedir. Doğrusal olmayan sistemler için uygun değildir.

Lyapunov kararlılık analizi şöyle yapılabilir. Sistemi temsil edilen matematiksel model üzerinde sürekli ve türevlenebilir bir $F(x)$ fonksiyonu tanımlanmalı ve bu fonksiyon pozitif definit ($F > 0$) veya yarı-pozitif definit ($F \geq 0$) olmalıdır. Bir fonksiyona veya sisteme Lyapunov kararlılık kriterinin uygulanabilmesi için bu şartları taşıması önemlidir. Bu şartları taşıyarak tanımlanan her sisteme Lyapunov kararlılık kriteri uygulanabilir [74].

$F > 0$ veya $F \geq 0$ için Lyapunov enerji fonksiyonu ,

$$V = F(x) \quad (2.34)$$

verilmiş olsun. $x \in R$ parametresi için Lyapunov enerji fonksiyonun Lyapunov kararlı olabilmesi için yeter koşul,

$$\frac{dV}{dx} \leq 0 \quad (2.35)$$

ile ifade edilir. Bu denklemden enerji fonksiyonunun zamanla azalması veya sabit kalması anlamına gelir ve sistemin kararlılık durumuna işaret eder.

Kontrol uygulamaları için Lyapunov kararlılık yönteminin adımları [66];

1) Sistemi ifade eden bir Lyapunov fonksiyonu oluşturulur. Bu fonksiyonun pozitif definit veya semi pozitif definit olması koşulu unutulmamalıdır. kapalı çevrim kontrol sistemlerinden örnek verecek olursak kontrolde genelde kontrol hatası kararlılığı gösteren önemli bir sistem parametresidir. Kontrol hatasının zamanla azalması sistem kararlılığının bir göstergesidir. Kontrol hatasının karesi alınarak semi pozitif definit Lyapunov fonksiyonu tanımlanabilir.

$$V = e(t)^2 \quad (2.36)$$

2) $\frac{dV}{dt} < 0$ olduğu gösterilirse sistem kararlıdır. Bu şart denklemin (2.36)'da uygulanırsa,

$$\frac{dV}{dt} = 2e(t) \frac{de(t)}{dt} < 0 \quad (2.37)$$

koşulu elde edilir. Bu koşulun doğruluğu sistem için matematiksel olarak gösterilirse, sistem kararlıdır denir.

Örnek : $\frac{dy}{dt} = -y$ sisteminin Lyapunov kararlılığını inceleyiniz.

İlk olarak sistemin durumunu ifade eden pozitif definit veya semi pozitif definit Lyapunov fonksiyonunun oluşturulması gerekmektedir. Lyapunov fonksiyonu olarak y 'nin enerjisinin alınacağı düşünülürse,

1) $V = y^2$ pozitif definit bir fonksiyondur.

2) $\frac{dV}{dt} < 0$ olduğu kanıtlanmalıdır.

$$\frac{dV}{dt} = 2yy' = 2y(-y) = -2y^2 \quad (2.38)$$

$$-2y^2 < 0 \quad (2.39)$$

olduğu için sistem kararlıdır.

2.6 Direk Gradyan İniş Yöntemi

Klasik kontrol sistemleri model bağımlıdır. Direk gradyan iniş kontrol yöntemi, modeli belirli olan doğrusal olmayan klasik sistemlerin doğrudan gradyan iniş çözümünü kullanarak kontrolüne dayanmaktadır. Kontrol hatasının karesi ile tanımlanan hata fonksiyonun gradyan iniş yöntemi ile minimizasyonuna dayanmaktadır. Shimizu ve arkadaşları tarafından 1993'te öne sürülmüştür [76]. Bu çalışmada kontrol işareti doğrudan gradyan iniş çözümü ile üretilebilmiştir. Direk gradyan iniş kontrol sistemlerinin kararlılığı Lyapunov kriterine göre incelenmiştir [14, 77]. Zubov kararlılık teoremi [77] ve Sontag formülasyonu [78] kullanılarak direk gradyan iniş kontrolünde kararlılık bölgesi tespit çalışmaları yapılmıştır [77]. Direk Gradyan iniş kontrolünün performansını arttırmak için çalışmalar yapılmış. Yöntemin yakınsama hızı ve kararlılığının iyileştirilmesi için önerilerde bulunulmuş böylece hem performans fonksiyonu hem de zaman türevi aynı anda azaltılmıştır [79, 80, 81]. Bu yöntem çevrimiçi bir uygulamadır ve basit bir şekilde uygulanabilir. Daha sonra doğrusal kontrol sisteminin kararlılığı için dinamik feedback kontrolü üzerinde çalışılmıştır [82].

Klasik kontrol sistemlerinde rastlanan model bağımlılığı, kontrol sisteminin statik olması ve adaptasyon kabiliyetinin azalmasına yol açmaktadır. Adaptif kontrol bu nedenle önemlidir ve mümkün olduğunca modelleme bağımsız ve akıllı kontrol sistemlerine giden yolda önemli bir adımdır. Günümüzde gradyan iniş yöntemi akıllı kontrolün bir dalı olarak adaptif kontrol uygulamaları için geliştirilmeye çalışılmaktadır [23, 24]. Akıllı kontrol sistemleri kontrol edilen sistemlerin matematiksel modellemeye olan bağımlılığını azaltmayı, sistemin adaptasyon kabiliyetini arttırmayı amaç edinir. Böylece gerçek dünya şartlarında daha verimli, adaptif ve dayanıklı kontrol performansı sağlayan kontrol sistemlerinin oluşumuna imkan sağlamayı amaçlar.

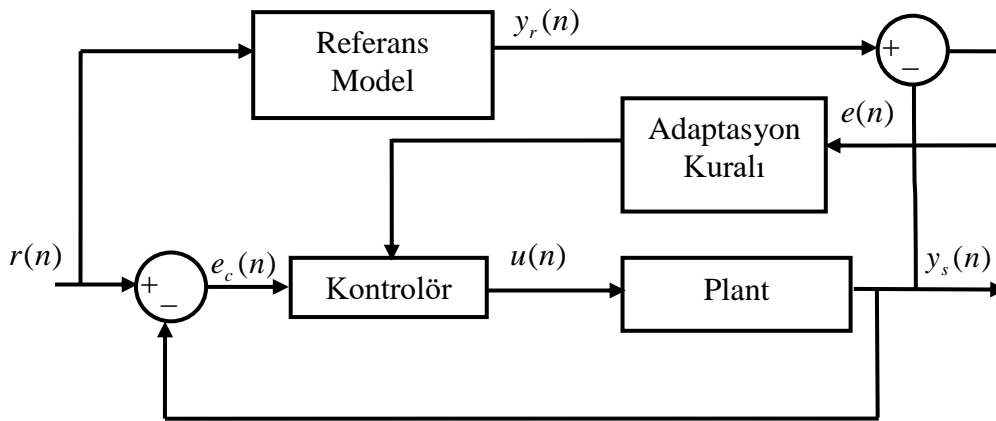
2.7 Adaptif Gradyan İniş Kontrol

Kontrol sisteminin değişen gerçek dünya dinamiklerine uyum sağlama yeteneği kontrol performansı için son derece önemlidir. Kontrol sistemleri sistem parametrelerinin değişmesine yol açan etkenler ve bozucu etkiler karşısında dayanıklı kontrol performansı sergileyebilmelidir. Önceki bölümde matematiksel sınırlamalardan dolayı gerçek dünyanın veya gerçek sistemlerin mükemmel olarak modellenemediğinden bahsedilmiştir. Değişen bu koşullara uyum sağlayamayan kontrol sistemleri simülasyon çalışmalarında yeterli gibi görünse de, gerçek dünya koşullarında aynı performansı gösteremezler. Bu nedenle kontrol

sistemleri konusundaki çalışmalar adaptif (uyarlanabilir) ve akıllı sistemlere doğru yönelmektedir. Sistemlerin adaptasyon becerisi ve öğrenme becerisi modern kontrol sistemlerinin güvenilir ve dayanıklı kontrol performansı için önemli özellikler olacaktır [23].

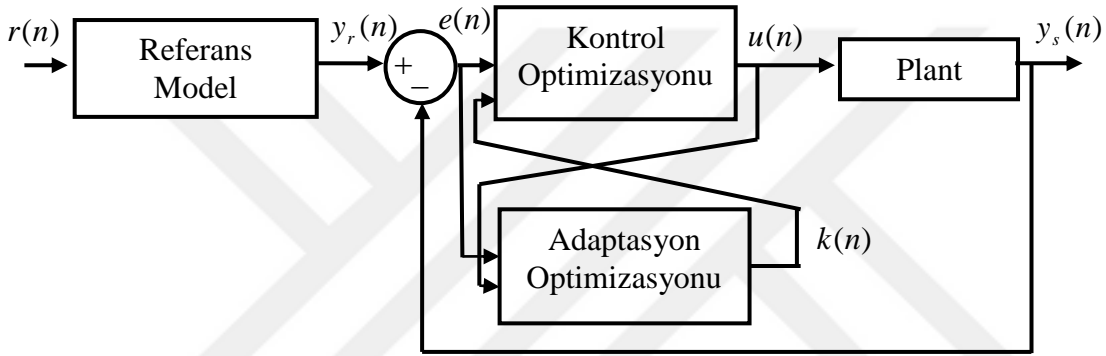
Adaptif kontrol, kontrol sistemlerinin değişen koşullara uyum sağlaması ve kontrol performansını sürdürülebilmesi durumunu ifade eder. Değişen koşullara bağlı olarak kontrol performansı kötüleşen sistemler için, adaptif kontrol işlemleri kontrol performansını geri elde etmek için adaptasyon adı verilen süreci gerçekleştirir [23]. İstenen kontrol cevabının referans bir model ile tanımlandığı kontrol sistemleri Model Referans Adaptif Kontrol (MRAK) sistemleri olarak adlandırılır. MRAK çalışmaları 70'lerden itibaren gündeme gelmeye başlar [83, 84, 85, 86, 87]. Daha sonra MRAK, adaptif kontrol çalışmalarının temel bir konusuna dönüşmüştür [86, 87, 88, 17, 89, 90].

Genel olarak şekil 2.28'de görüldüğü üzere MRAK sistemleri 2 çevrim içerir [91, 46] : İç döngü, ayarlanabilir katsayılarla sahip bir kontrolör ve kontrol edilecek bir planttan oluşan bir kapalı çevrim kontrol yapısı içerir. Bu döngü sistem kararlılığını ve kontrolünü sağlar. Referans model ve adaptasyon bloğundan oluşan bir dış döngü kontrolör parametrelerini değişen koşullara adapte etmek için ayarlar [23, 46, 92]. Tek çevrimli uygulama olarak MIT kuralı model referans adaptive kontrol için önerilmiştir [91]. Ayrıca MIT kuralının iki çevrimli uygulamaları PID ve kesir dereceli PID kontrol sistemlerine adaptiflik özelliği kazandırmak için uygulanmıştır [93, 94, 95].



Şekil 2.28 : Genel MRAK yapısı [23]

Literatürde MIT kuralı dışında kontrol ve adaptasyon işlemlerinin sadece model hatasına göre gerçekleştirildiği tek çevrimli yaklaşımlar sunulmuştur. Dikkat çeken bir diğer çalışma Referans Biçimlendiren Adaptif Kontrol (RBAK) olarak isimlendirilen bir model referans adaptif kontrol yaklaşımıdır [23]. Şekil 2.29’da görüldüğü üzere bu yaklaşıma göre kontrol sisteminin referans girişi doğrudan kontrol sistemine uygulanmaz. Referans giriş önce bir referans model tarafından şekillendirilir, sonra kapalı çevrim kontrol sistemine uygulanır. Önerilen adaptif kontrolör sadece referans modelinin çıkışını takip eder. Kontrol optimizasyonu, model hatasını azaltmayı hedefleyen gradyan iniş optimizasyonu uygular ve adaptasyon optimizasyonu kontrol edilen sistemin anlık giriş-çıkış ilişkisini temsil eden bir iç modelin katsayılarını optimize eder.



Şekil 2.29 : Önerilen RBAK sistem blok diyagramı [23]

RBAK yaklaşımında referans model ile kontrol sistemi arasındaki model uyumsuzluğu, referans sistem çıkışı $y_r(n)$ ile plant çıkışı $y_s(n)$ arasındaki fark olarak anlık hata sinyali $e(n)$ ile ifade edilmiştir.

$$e(n) = y_r(n) - y_s(n) \quad (2.40)$$

Hata fonksiyonu, çıktıların ($y_r(n), y_s(n)$) karesel farkı olarak ifade edilmiştir,

$$E(n) = \frac{1}{2} e(n)^2 = \frac{1}{2} (y_r(n) - y_s(n))^2. \quad (2.41)$$

Bu maliyet fonksiyonunu minimize etmek için kontrol sinyalini üreten kontrol optimizasyonu için MIT kuralının temeli olan şu formül yazılmıştır,

$$\frac{\partial u}{\partial t} = -\eta_c \frac{\partial E}{\partial u} \quad (2.42)$$

Bu denklemdeki η_c parametresi daha önce de değinilen öğrenme katsayısı adlandırılan öğrenme parametresidir. Referans modeli kontrol sinyalinden bağımsız olduğu için $\frac{\partial y_r}{\partial u} = 0$ ' a neden olur. Böylece,

$$\frac{\partial E}{\partial U} = -\frac{\partial y_s}{\partial u} e \quad (2.43)$$

şeklinde yazılabilir. Plantın anlık giriş-çıkış ilişkisi [23],

$$y_s(t) = k(t)u(t) \quad (2.44)$$

şeklinde ifade edilmiştir. Bu denklem bize herhangi bir zamanda sistem çıkışının giriş değerinin zamanla değişen bir faktörü olarak yazılabileceğini gösterir. Denklem (2.43)'e göre $\frac{\partial y_s}{\partial u} = k$ yazılabilir ve kontrol sinyali için ,

$$\frac{\partial u}{\partial t} = \eta_c k(t)e(t) \quad (2.45)$$

elde edilir. Bu formülasyon ayrık zamanda ifade edilirse [23],

$$u(n+1) = u(n) + \eta_c k(n)e(n) \quad (2.46)$$

denklemini elde edilir. Daha sonra $k(n)$ 'i tahmin eden adaptasyon optimizasyonu için gradyan iniş formülü [23],

$$\frac{\partial k}{\partial t} = -\eta_a \frac{\partial E}{\partial k} \quad (2.47)$$

şeklindedir. Burada $\frac{\partial E}{\partial k}$ işleminin sonucu,

$$\frac{\partial E}{\partial k} = -\frac{\partial y_s}{\partial k} e \quad (2.48)$$

dir. Denklem 2.44'ya göre $\frac{\partial y_s}{\partial k} = u$ yazılırsa yeni denklem,

$$\frac{\partial k}{\partial t} = \eta_a u(t)e(t) \quad (2.49)$$

şeklinde yazılır. Bu denklemin ayrık zamanda gösterimi [23],

$$k(n+1) = k(n) + \eta_a u(n)e(n) \quad (2.50)$$

Şeklinde elde edilir.

Sonuç olarak makalede önerilen RBAK yapısı, iki gradyan iniş optimizasyonu işlemi gerçekleştirir [23]. Birincisi, kontrol sinyali üretimi için ikincisi ise adaptasyon işlemi için anlık giriş-çıkış ilişkisini ifade eden iç modelin (internal model) iyileştirilmesi için kullanılır. RBAK yapısı oldukça basittir bir yapıdadır ve öğrenme kuralları basit denklemlere dayanır. Bu bir iç-modele göre gradyan iniş nümerik çözümlerine dayanan bu yöntem adaptif gradyan iniş kontrol (AGİK) adı verilmiştir. Önemli bir sınırlaması kararlı plant modellerinde uygulanabilir olmasıdır. Kontrol edilen sistem kararsız olduğu zaman AGİK'in yetersiz kalmakta ve kontrol sistemi kararlı sonuçlar verememektedir. Diğer bir sınırlama gradyan iniş çözümlerinin kararlılığının garanti edilememesidir. Kararlı plant durumlarında öğrenme katsayısı yeterince küçük değilse gradyan iniş kontrol çözümleri kararsız olmaktadır. Bu tez çalışmada ilerleyen bölümlerde gradyan iniş dinamiğinin kararlılık koşulları Lyapunov kararlılık kriterine göre incelenmiş ve Lyapunov kararlılık kriterine elde edilmiştir.

Daha sonraları yukarıda önerilen AGİK yöntemi çözümleri doğrusal olmayan sistemlerin adaptif kontrolünü sağlamak için geliştirilmiştir [96]. Doğrusal olmayan dinamik plant fonksiyonlarının dinamiğini daha iyi takip edebilmek için anlık giriş-çıkış ilişkisini ifade eden iç model polinom olarak tanımlanmıştır. Polinomdaki lineer olmayan terimlerin katsayıları adaptasyon optimizasyonu ile optimize edilerek doğrusal olmayan sistem cevabına iç-modelin daha kolay adapte olması sağlanmıştır [96]. Çalışmada lineer olmayan anlık giriş – çıkış ilişkisi,

$$y_s(t) = k_1(t)u(t) + k_2(t)u(t)^2 + k_3(t)u(t)^3 + k_4(t)u(t)^4 \quad (2.51)$$

ile ifade edilmiştir. Buradaki $k_1(t)$, $k_2(t)$, $k_3(t)$, $k_4(t)$ ve $k_5(t)$ dinamik adaptasyon katsayılarıdır.

Denklem (2.41) ile ifade edilen karesel hata fonksiyonuna göre $E(t)$ 'yi minimize eden kontrol sinyali,

$$\frac{\partial u}{\partial t} = -\eta_c \frac{\partial E}{\partial u} \quad (2.52)$$

ile bulunur. $E(t)$ 'yi minimize edecek zamanla değişen adaptasyon katsayıları,

$$\frac{\partial k_1}{\partial t} = -\eta_{a1} \frac{\partial E}{\partial k_1}, \quad (2.53)$$

$$\frac{\partial k_2}{\partial t} = -\eta_{a2} \frac{\partial E}{\partial k_2}, \quad (2.54)$$

$$\frac{\partial k_3}{\partial t} = -\eta_{a3} \frac{\partial E}{\partial k_3}, \quad (2.55)$$

$$\frac{\partial k_4}{\partial t} = -\eta_{a4} \frac{\partial E}{\partial k_4} \quad (2.56)$$

ile elde edilir. Denklem (2.51) kullanılarak,

$$\frac{\partial y_s}{\partial u} = \sum_{i=1}^4 i.k_i(t).u(t)^{i-1}, \quad (2.57)$$

$$\frac{\partial y_s}{\partial k_i} = u(t)^i \quad (2.58)$$

Formülleri elde edilir. $E = \frac{1}{2} e(t)^2 = \frac{1}{2} (r(t) - y_s(t))^2$ denklemi de göz önüne alınarak sürekli zamanda,

$$u(t) = \eta_c \int (k_1(t) + 2k_2(t)u(t) + 3k_3(t)u(t)^2 + 4k_4(t)u(t)^3) e(t) dt, \quad (2.59)$$

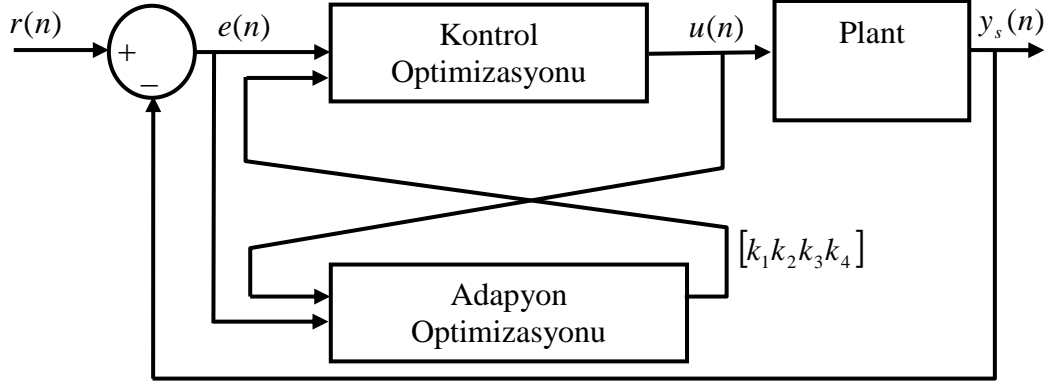
$$k_1(t) = \eta_{a1} \int u(t).e(t) dt, \quad (2.60)$$

$$k_2(t) = \eta_{a2} \int u(t)^2 .e(t) dt, \quad (2.61)$$

$$k_3(t) = \eta_{a3} \int u(t)^3 .e(t) dt, \quad (2.62)$$

$$k_4(t) = \eta_{a4} \int u(t)^4 .e(t) dt \quad (2.63)$$

Gardyan iniş çözümleri elde edilir. Şekil 2.30'da bu çözümlerin kontrol sisteminde kullanımı gösterilmiştir. Denklem (2.59) kontrol optimizasyonu bloğunu oluşturmuştur. Denklem (2.60), (2.61), (2.62) ve (2.63) adaptasyon optimizasyonu bloğunu oluşturmuştur.



Şekil 2.30 : Dördüncü dereceden anlık giriş – çıkış ilişkisi için gradyan iniş kontrolün blok diyagramı [96]

Sonuç olarak, adaptasyon kabiliyeti ve plant modeli bağımsızlığı akıllı kontrol yöntemlerinin temel hedefi olmalıdır. Bu çalışmada kontroller tasarımının plant modellemesine olan bağımlılığının azaltılması, plant fonksiyonunun anlık giriş - çıkış ilişki modeli ile sağlanmıştır.

Bu tez çalışmasında adaptif gradyan iniş kontrol yapısına katkı yapılmış ve lineer dinamik plant fonksiyonları için daha iyi sonuçlar elde edebilmek için değişken katsayılı sonlu impuls cevabı (FIR) anlık giriş-çıkış ilişki için gradyan iniş çözümleri elde edilmiş ve simülasyon zaman gecikmeli birinci derece plant fonksiyonları için performans incelemesi yapılmıştır [71]. Bu konu dördüncü bölümde ele alınmıştır.

3. SÜREKLİ ZAMAN GRADYAN İNİŞ DİNAMİĞİ LYAPUNOV KARARLILIĞI VE NÜMERİK GRADYAN İNİŞ YÖNTEMİ ÇÖZÜMLERİ İLE KIYASLAMA

Önceki bölümlerde gradyan iniş algoritmasından detaylı bir şekilde bahsedilmiştir. Gradyan iniş algoritmasının günümüzde yaygın kullanımı olmuştur. Buna rağmen yöntemin kontrol uygulamalarında kullanımı diğer alanlardaki kullanımına göre sınırlı kalmıştır. Bunun temel nedeni matematiksel olarak dinamik sistem modelleri ile ifade edilen sistemlerin gerçek zamanlı optimizasyonunda zorluklar yaşanmasıdır. Bunlardan birincisi kontrol sistemleri dinamik cevabı takip edebilmesi ve kontrol edebilmesi için yeterince hızlı cevaba ihtiyaç duyar. Optimizasyon yöntemlerinin işlem maliyetleri çoğunlukla yüksek olduğu için gerçek zamanlı uygulamalar için hızlı donanıma ihtiyaç duyar. Diğer bir önemli nokta optimizasyon yönteminin çözümünün kararlıdır. Kontrol sistemlerinin kararlı kalabilmesi için optimizasyon yöntemi her zaman kararlı çözüme yakınsayabilmelidir. Gradyan iniş yönteminin yakınsama hızının artırılması ve gradyan iniş çözümlerinin dayanıklı kararlı olabilmesi kontrol uygulamaları için zorunludur. Bu nedenle bu yönde yapılacak çalışmalara ve gelişmelere ihtiyaç duyulmaktadır.

Bu amaçla bu bölümde sürekli zaman gradyan iniş dinamiğinin Lyapunov kararlılığı incelenmiş ve sürekli gradyan iniş dinamiğinin kararlılığı için gerekli Lyapunov kararlılık koşulu elde edilmiştir. Nümerik gradyan iniş çözümünün aslında sürekli gradyan iniş çözümlerinin Euler nümerik çözümü olduğu gösterilmiş ve nümerik gradyan iniş yöntemi sonuçları ve sürekli gradyan iniş dinamiğinin sonuçları De Jong fonksiyonu üzerinde karşılaştırılmıştır. Bu çalışma tezden çıkan yayın olarak uluslararası konferansta bildiri olarak sunulmuştur [18].

3.1. Sürekli Zaman Gradyan İniş Dinamiği ve Nümerik Gradyan İniş Yöntemi

Sürekli zaman gradyan iniş dinamiği,

$$\frac{dx_i}{dt} = -\gamma_i \frac{\partial F}{\partial x_i} \quad (3.1)$$

denklemleri ile ifade edilmektedir [97, 98, 99]. Adaptif kontrol uygulamalarında gradyan iniş işlemi MIT kuralı olarak adlandırılmıştır [23, 17, 46]. Bu denklemi Euler yöntemine göre çözmek için ileri fark denklemi,

$$\frac{dx_i}{dt} \cong \frac{x_i[n+1] - x_i[n]}{\Delta t} \quad (3.2)$$

kullanılırsa denklem (3.1)'deki temel gradyan iniş yöntemi elde edilmiş olur. Bu formülasyondaki Δt parametresi örnekleme periyodu olarak adlandırılır. Nümerik gradyan iniş dinamiği

$$x_i[n+1] = x_i[n] - \gamma_i \Delta t \frac{\partial F}{\partial x_i} \quad (3.3)$$

elde edilir. Burada $\eta_i = \gamma_i \Delta t$ nümerik çözüm öğrenme katsayısıdır. Böylece nümerik gradyan iniş yöntemi temel formülü,

$$x_i[n+1] = x_i[n] - \eta_i \frac{\partial F}{\partial x_i} \quad (3.4)$$

denklemleri ile ifade edilir[53]. Bu denklemdeki $F(x)$ optimizasyonu gerçekleştirilecek yani minimizasyonu yapılacak amaç fonksiyonudur. $x \in R^p$ vektörünün bileşenleri $x_i \in R$, denklemin parametreleridir.

3.2 Gradyan İniş Dinamiği Kararlılık Analizi

Denklem (3.1) ile ifade edilen sürekli zaman gradyan iniş yönteminin kararlılık şartları Lyapunov kararlılık kriteri yardımı ile elde edilmektedir.

Teorem(Lyapunov Kararlılık Koşulu): Sürekli ve türevlenebilir bir $F(x): R^p \rightarrow R$ pozitif definit ($F > 0$) veya yarı-positif definit ($F \geq 0$) reel değerli fonksiyonu için denklem (3.1) ile ifade edilen sürekli zaman gradyan iniş dinamiği $\forall \gamma_i > 0$ için her zaman Lyapunov kararlıdır.

İspat: Bir $F > 0$ veya $F \geq 0$ için Lyapunov enerji fonksiyonu,

$$V = F(x) \quad (3.5)$$

verilsin. $x_i \in R$ parametresi için Lyapunov enerji fonksiyonunun Lyapunov kararlı olabilmesi için yeter koşul,

$$\frac{dV_i}{dt} \leq 0 \quad (3.6)$$

ile ifade edilir [23, 67, 68]. Bu denklem enerji fonksiyonunun zamanla azalması veya sabit kalması özelliğini ifade eder. Bu şart yerine getirilirse,

$$\frac{dV_i}{dt} = \frac{dF}{dt} = \frac{\partial F}{\partial x_i} \frac{dx_i}{dt} \leq 0 \quad (3.7)$$

yazılabilir. Burada $\frac{dx_i}{dt}$ yerine denklem (3.1) kullanılırsa,

$$\frac{dV_i}{dt} = \frac{\partial F}{\partial x_i} (-\gamma_i \frac{\partial F}{\partial x_i}) \leq 0 \quad (3.8)$$

Gradyan iniş dinamiğinin Lyapunov kararlı olabilmesi için yeter koşul,

$$-\gamma_i \left(\frac{\partial F}{\partial x_i} \right)^2 \leq 0 \quad (3.9)$$

elde edilir. Bu eşitsizliğin her zaman ve her x_i için geçerli olabilmesi için yeterli şartın $\forall \gamma_i > 0$ olması gerektiği kolaylıkla görülmektedir.

Lyapunov kararlılık koşulu bize $\forall \gamma_i > 0$ koşulu sağlandığı takdirde denklem (3.1) ile ifade edilen sürekli zaman gradyan iniş dinamiğinin dayanıklı kararlı olacağını ve $x_i(t)$ işaretinin zamanla $F(x)$ 'in azaltan yönde değişeceğini ifade eder. Bu da yöntemin kontrol sistemi tasarımlarında kararlılık sorunlarının çözümünde uygulanabilmene imkan sağlar.

3.3 Örnek Nümerik Analiz

Bu bölümde De Jong fonksiyonu [100] türünde iki değişkenli,

$$F(x_1, x_2) = \sum_{i=1}^2 c_i x_i^2 \quad (3.10)$$

fonksiyonun sürekli zaman gradyan iniş dinamiğinin Euler yöntemi ile nümerik analizi yapılacaktır. c_1 ve c_2 katsayıları pozitif reel sayılardır. Sürekli zaman gradyan iniş dinamiği bu fonksiyona göre ifade edilirse,

$$\frac{\partial F(x_1, x_2)}{\partial x_1} = 2c_1 x_1 \quad (3.11)$$

$$\frac{\partial F(x_1, x_2)}{\partial x_2} = 2c_2 x_2 \quad (3.12)$$

elde edilir. Bu ifadeler denklem (3.4)'te yerine koyulursa,

$$\frac{dx_1}{dt} = -\gamma_1 2c_1 x_1 \quad (3.13)$$

$$\frac{dx_2}{dt} = -\gamma_2 2c_2 x_2 \quad (3.14)$$

diferansiyel denklemlerin analitik çözümü,

$$x_1(t) = x_1(0)e^{-2c_1 \gamma_1 t} \quad (3.15)$$

$$x_2(t) = x_2(0)e^{-2c_2 \gamma_2 t} \quad (3.16)$$

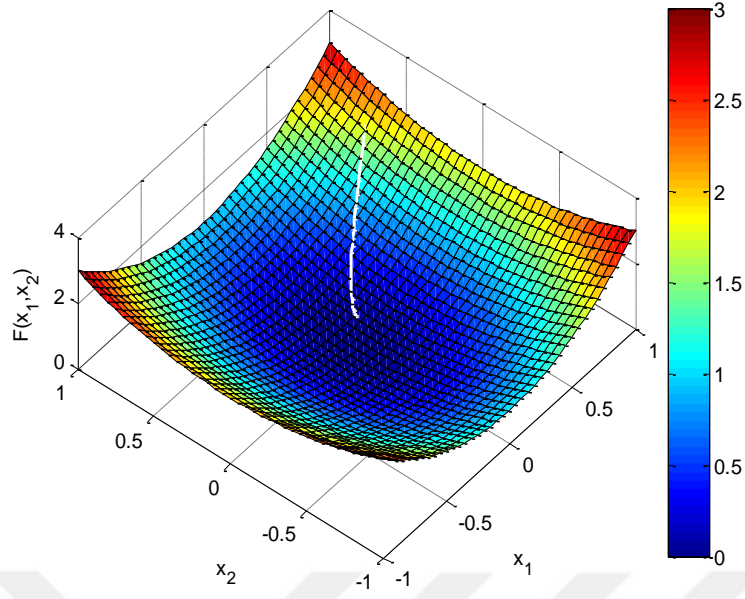
Bu analitik çözümler, nümerik çözümlerin doğruluğunu kıyaslayabilmek için referans çözüm olarak kullanılacaktır. Denklem (3.2) yardımı ile denklem (3.13) ve denklem (3.14) Euler yöntemi ileri fark nümerik çözümü,

$$x_1[n+1] \cong x_1[n] - \Delta t \gamma_1 2c_1 x_1[n] \quad (3.17)$$

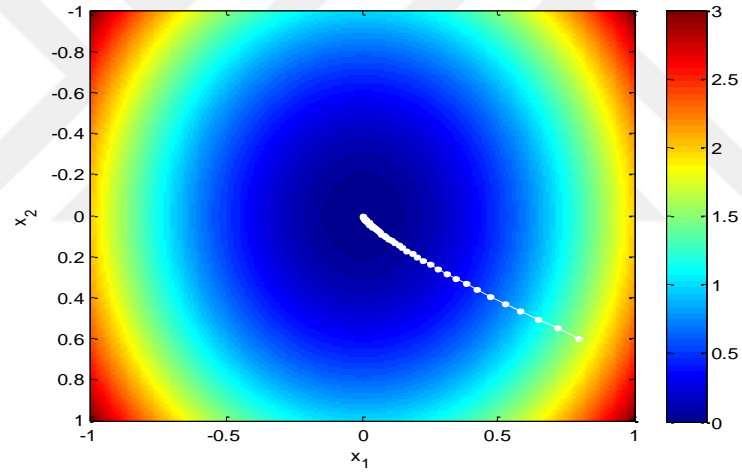
$$x_2[n+1] \cong x_2[n] - \Delta t \gamma_2 2c_2 x_2[n] \quad (3.18)$$

elde edilir. $\eta_1 = \Delta t \gamma_1$ ve $\eta_2 = \Delta t \gamma_2$ katsayıları makine öğrenmesi uygulamalarında da yaygın olarak öğrenme katsayısı olarak adlandırılmıştır. Şekil 3.1 ve 3.2'de $c_1 = 1, c_2 = 2, \gamma_1 = 0.5, \gamma_2 = 0.2$ ve $\Delta t = 0.1$ için gradyan iniş dinamiğinin nümerik çözümü verilmiştir. Buradaki $x_1[n]$ ve $x_2[n]$ çözümlerinin başlangıç noktası $x_1[0] = 0.8$ ve $x_2[0] = -0.6$ noktasından minimuma giderken izlediği yol verilmiştir.

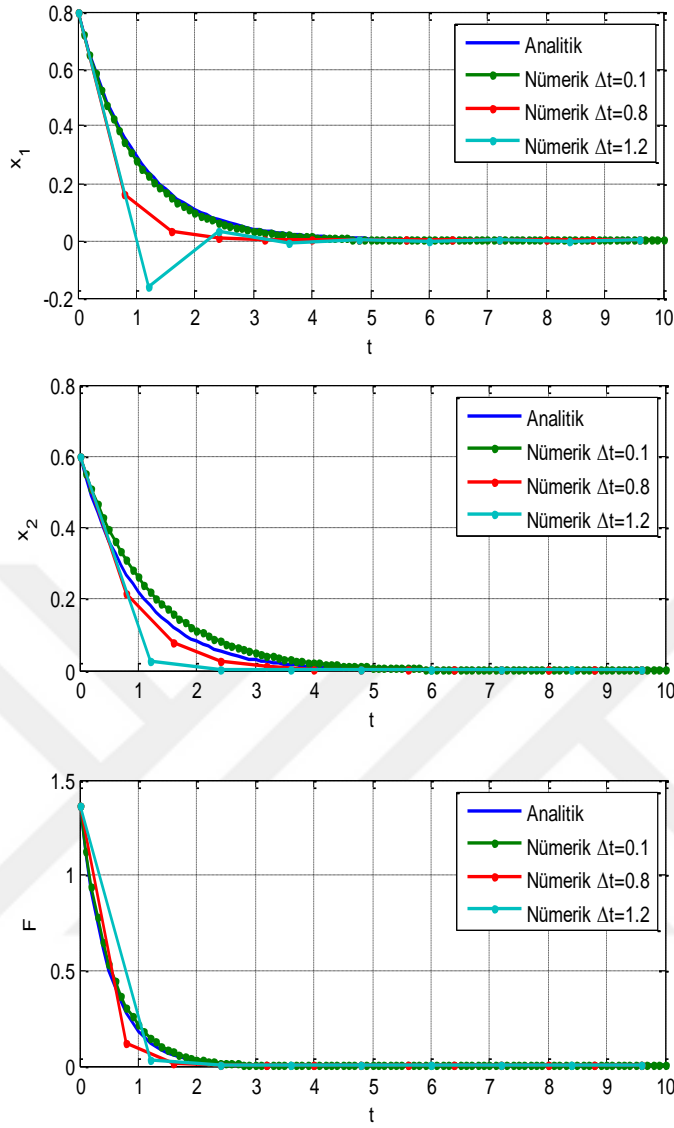
Şekil 3.3'te farklı Δt değerleri için gradyan iniş çözümleri verilmiştir. Δt azaldıkça nümerik çözümün analitik çözüme nasıl yakınsadığı görülmektedir.



Şekil 3.1 : Nümerik gradyan iniş çözümünün üç boyutlu yörüngesi



Şekil 3.2: Nümerik gradyan iniş çözümünün iki boyutlu yörüngesi.



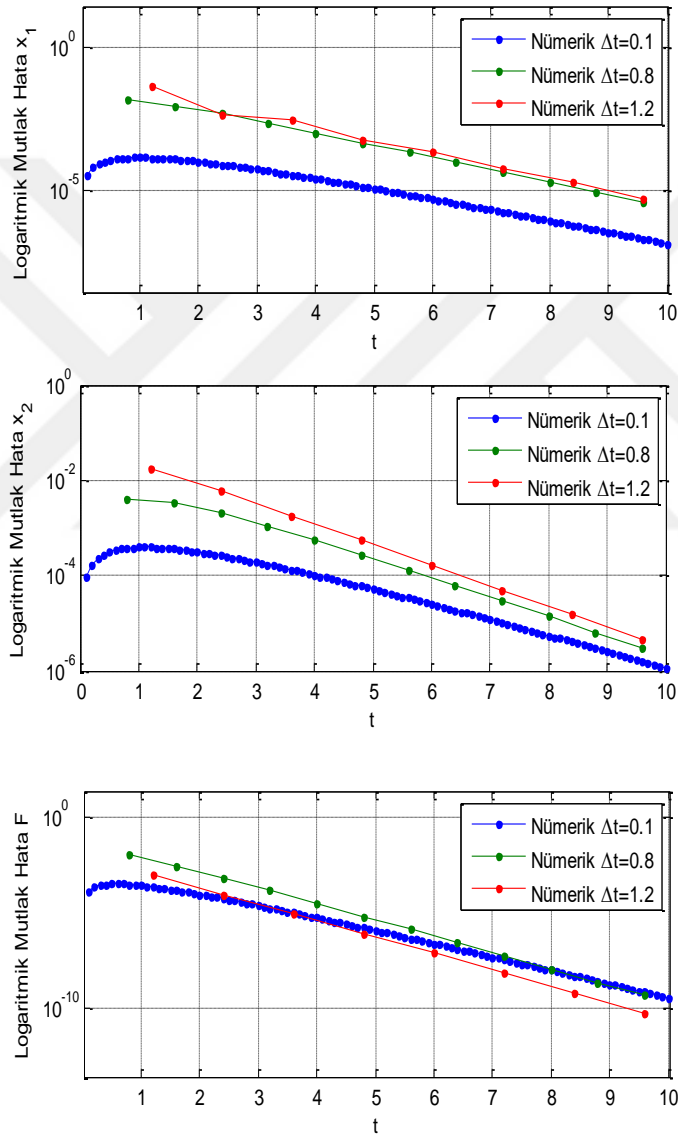
Şekil 3.3: Farklı örnekleme periyotları için nümerik çözüm ile analitik çözüm karşılaştırılması.

Çizelge 3.1’de analitik çözüme göre nümerik çözümlerin ortalama mutlak hataları verilmiştir. İkinci satırda mutlak hata artışı hızlı yakınsamanın sağladığı düşüğe karşın yakınsama hatalarının neden olabildiği artıştan kaynaklanmıştır.

Çizelge 3.1: Nümerik çözümün ortalama mutlak hata değerleri.

Ortalama Mutlak Hata	F	x_1	x_2
$\Delta t = 0.1$	0.0055	0.0040	0.0118
$\Delta t = 0.8$	0.0162	0.0348	0.0117
$\Delta t = 1.2$	0.0120	0.0531	0.0260

Şekil 3.4'te, Çizelge 3.1'deki ortalama mutlak hataların anlık dağılımı görülmektedir. Ortalama mutlak hata analitik çözüme göre uygulanmıştır. Düşük Δt değerleri için nümerik çözüm başlangıçta analitik çözüme göre daha iyi yakınsama sağlayabilmesine rağmen ilerleyen zamanlarda yani $F(x_1, x_2)$ 'in sifira yaklaştığı zamanda büyük Δt yakınsaması iyileşebilmektedir. Bu etkiler Çizelge 3.1'de görünen üçüncü satırdaki ortalama mutlak hata verilerine neden olmaktadır.



Şekil 3.4 : Anlık mutlak hata dağılımı.

4. KARARLI, BİRİNCİ DERECE, ZAMAN GECİKMELİ DİNAMİK SİSTEM MODELLERİNİN ZAMANLA DEĞİŞEN FIR FİLTRE MODELİNE GÖRE ADAPTİF GRADYAN İNİŞ KONTROLÜ

Bu bölümde birinci dereceden kararlı, zaman gecikmeli, LTI sistem modellerinin adaptif gradyan iniş yöntemi ile adaptif kontrolü incelenmiştir. Bu çalışma uluslararası konferansta bildiri olarak sunulmuştur. Adaptif gradyan iniş kontrol yöntemi Alagoz ve ekibi tarafından önerilmiştir [23, 96]. Bu konudaki detaylı bilgiler ikinci bölümde sunulmuştur. Bu tez çalışmasında anlık giriş-çıkış ilişkisini ifade eden iç-modelin zamanla değişen sonlu impulse cevabı (FIR) filtre olduğu varsayılmış ve bu varsayım altında adaptif gradyan iniş çözümleri elde edilmiştir. Burada sağlanan katkı birinci derece zaman gecikmeli sistemlerin gecikmeli dinamiğinin iç-modelde daha iyi temsil ederek adaptif kontrol performansının artırılmasıdır. Burada FIR filtre zaman gecikmesine maruz olan sistem dinamiklerinin FIR filtre yapısında girişin önceki değerlerini ($u(t-i)$) kullanarak temsil edebilmesi prensibine dayanır.

4.1 Teorik Arka Plan

Adaptif gradyan iniş kontrol yönteminde iç-model anlık giriş-çıkış ilişkisini zamanla değişen formunda varsayılmıştır [23].

$$y_s(t) = k(t)u(t). \quad (4.1)$$

Bu varsayım, zamanla değişen ve keyfi belirlenebilen $k(t)$ katsayısının sağladığı serbestlik derecesi sayesinde sistemlerin içinde bulunulan anlık giriş-çıkış ilişkisini temsil edebilir.

Daha sonraları giriş-çıkış ilişkisindeki doğrusal olmayan ilişkilere duyarlı olan ve doğrusal olmayan ilişkileri parametrik olarak temsil edebilecek olan anlık giriş-çıkış ilişkisi yüksek derecede polinomsal varsayılmıştır [96].

$$y_s(t) = k_1(t)u(t) + k_2(t)u(t)^2 + k_3(t)u(t)^3 + k_4(t)u(t)^4 \quad (4.2)$$

Bu yaklaşımlarda zaman gecikmesinden kaynaklanan çıkışta etkin olan dinamiğin gecikmesi durumu temsil edilememektedir. Örnekleme periyodu büyük seçilerek veya öğren katsayısı küçük seçilerek zaman gecikmesinden kaynaklanan temsil sorunları kompanse edilmeye çalışılmıştır. Ancak yüksek zaman gecikmeli durumda bu kompanzasyon yeterli olmamaktadır. Gecikme dinamiğinin giriş-çıkış ilişkisinde temsili için zaman gecikmeli değişkenlere ihtiyaç vardır. Bu noktada FIR filtre modelinin zaman gecikmeli terimleri

faydalı olabilecekti. Bu motivasyon ile zamanla deęişen FIR filtre formunda iç-model varsayımı yapılarak bu tez çalışmasında adaptif gradyan iniş yöntemine katkı yapılmıştır.

4.2 Zaman Deęişen FIR Filtre Modeli Varsayımı ile Adaptif Gradyan İniş Kontrolü

Anlık giriş-çıkış ilişkisini ifade eden iç-model zamanla deęişen FIR filtresi formunda,

$$y_s(t) = \sum_{i=0}^h k_i(t)u(t-i) = k_0(t)u(t) + k_1(t)u(t-1) + k_2(t)u(t-2) + \dots + k_h(t)u(t-h). \quad (4.3)$$

İfade edilmiştir. Buradaki k_1 k_2 k_3 ve k_4 parametrelerine, zaman gecikmeli etkilerin içinde bulunulan ana katkısını temsil eden adaptasyon katsayılarıdır. Böylece bu katsayılar zaman gecikmeli plant modelinin geciken dinamiklerini temsil ederek plant çıkışının kestirimini daha iyi sağlayacaktır. Böylece gradyan iniş kontrolün adaptasyon yeteneğinin zaman gecikmeli durumlar için iyileşebileceği öngörülmüştür.

Kararlı bir plant fonksiyonunun FIR filtre ile modellenmesi durumu için hata fonksiyonu

$$E(t) = \frac{1}{2} e(t)^2 = \frac{1}{2} (r(t) - y_s(t))^2 = \frac{1}{2} (r(t) - \sum_{i=0}^h k_i(t)u(t-i))^2. \quad (4.4)$$

Tüm zaman gecikmelerini göz önünde bulunduran kontrol optimizasyonu için gradyan iniş çözümleri,

$$\frac{\partial u}{\partial t} = -\eta_c \sum_{i=0}^h \frac{\partial E}{\partial u(t-iT_s)}, \quad (4.5)$$

Filtre katsayılarını belirleyen gradyan iniş çözümleri,

$$\begin{bmatrix} \frac{dk_0}{dt} \\ \frac{dk_1}{dt} \\ \frac{dk_2}{dt} \\ \vdots \\ \frac{dk_h}{dt} \end{bmatrix} = \begin{bmatrix} -\eta_{a0} & 0 & 0 & 0 & 0 \\ 0 & -\eta_{a1} & 0 & 0 & 0 \\ 0 & 0 & -\eta_{a2} & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & -\eta_{ah} \end{bmatrix} \begin{bmatrix} \frac{dE}{dk_0} \\ \frac{dE}{dk_1} \\ \frac{dE}{dk_2} \\ \vdots \\ \frac{dE}{dk_h} \end{bmatrix} \quad (4.6)$$

Buradaki η_{ai} katsayıları FIR filtre katsayılarının gradyan iniş çözümleri için öğrenme katsayılarıdır. Denklem (4.5) için türev terimlerin toplamı bulunur.

$$\sum_{i=0}^h \frac{\partial E(t)}{\partial u(t-iT_s)} = -\sum_{i=1}^h k_i(t)e(t) \quad (4.7)$$

Denklem (4.6) için türev vektörlerinin bileşenleri,

$$\frac{dE(t)}{dk_i(t)} = -u(t-iT_s)e(t) \quad (4.8)$$

Ayrık zamanda sonlu fark Euler çözümünün örnekleme periyodu $t = nT_s$, alınırsa kontrol optimizasyonu için nümerik gradyan iniş çözümü

$$u[n+1] = u[n] + \eta_c \sum_{i=1}^h k_i(t)e(t) \quad (4.9)$$

elde edilir. Filtre katsayıları için nümerik gradyan iniş çözümleri

$$k_i[n+1] = k_i[n] + \eta_c u(t-iT_s)e(t) \quad (4.10)$$

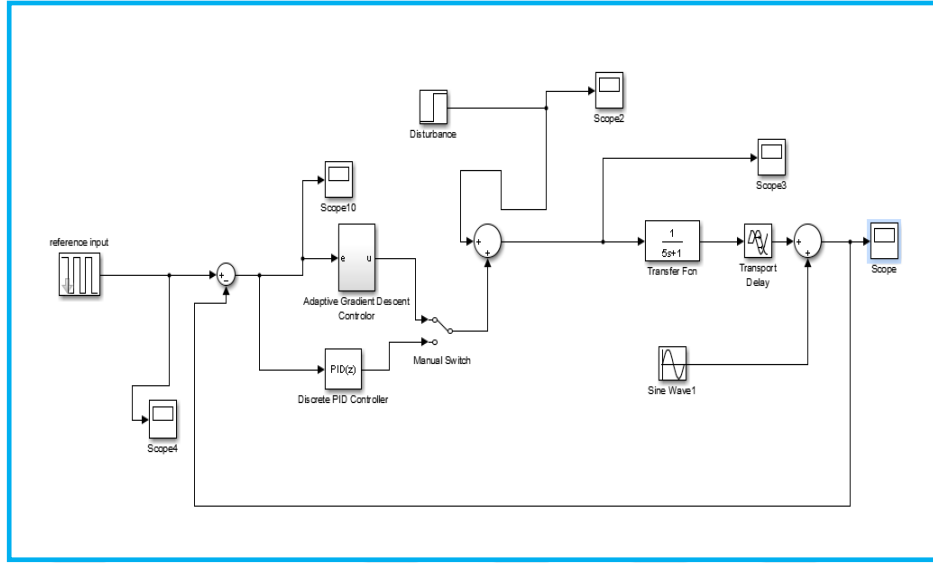
ile ifade edilir. Önerilen adaptif gradyan iniş kontrolünü test etmek için Matlab Simulink simülasyon çalışması gelecek bölümde sunulmuştur.

4.3. Simülasyon Çalışması

Bu bölümde $G(s) = \frac{K_{DC}}{\tau s + 1} e^{-Ls}$ formunda ifade edilen birinci mertebe zaman gecikmeli kararlı plant fonksiyonunu kontrol çalışması yapılmıştır. Burada DC kazanımı (K_{DC}), zaman sabiti (τ) ve zaman gecikmesi (L) parametrelerini ile temsil edilmiştir ve bu parametrelerin değişimine karşı kontrol sistemlerinin adaptasyon kabiliyetleri değerlendirilmiştir. Bu çalışmada tasarlanan adaptif gardyan iniş kontrollör ile klasik PI kontrollörün performansları karşılaştırmalı incelenmiştir. Bu amaçla Şekil 4.1'de görülen Matlab simulink simülasyon ortamı geliştirilmiştir. Simülasyonlarda, FIR filtresi için $h = 5$ alınmıştır. Simülasyon örnekleme süresi 0.1 saniye olarak ayarlanmıştır. Tüm öğrenme katsayıları η_c, η_{ai} 10^{-3} olarak ayarlanmıştır.

Matlab PID optimizer yardımı ile PI kontrollörleri şöyle tasarlanmıştır.

$$C_{pid}(s) = 0.026119080 + 0.052238161s^{-1}. \quad (4.11)$$



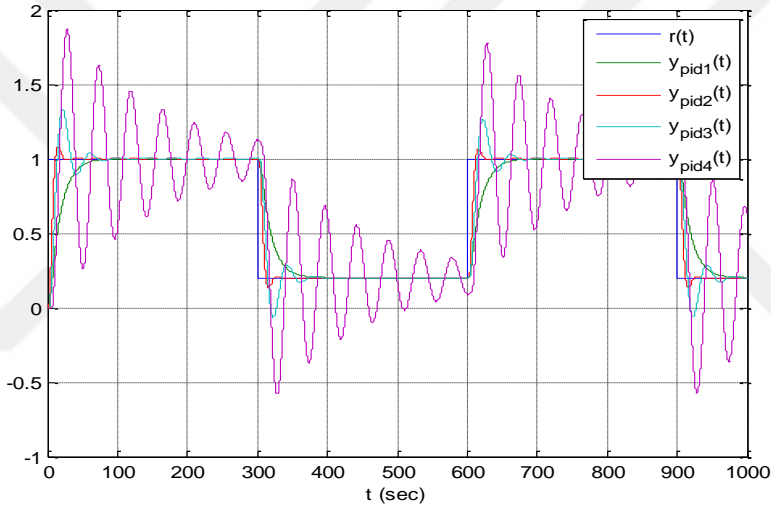
Şekil 4.1 : Matlab Simulink kontrol sistemi simülasyon modeli

Çizelge 4.1’de adaptasyon performansını değerlendirmek için kullanılan plant modelleri sunulmuştur. Kontroller tasarımları $G_1(s)$ plantına göre yapıldı ve $G_1(s)$ modeli orijinal plant olarak kabul edildi. Daha sonra $G_1(s)$ plant parametreleri değiştirilerek parametrik değişime uğramış varyantları olan $G_2(s)$, $G_2(s)$, $G_3(s)$ ve $G_4(s)$ modelleri elde edilmiştir. Böylece $G_1(s)$ plant fonksiyondaki parametrik değişimlerin derecesini göstermek için bir referans model olmuştur.

Çizelge 4.1 : Kontrol performans testleri için parametrik olarak değişen plant modelleri ve bu modellerin $G_1(s)$ modeline göre değişim oranları.

Plant Perturbation	Change rate of DC gain K_{DC}	Change rate of time constant τ	Change rate of time delay L
$G_1(s) = \frac{1}{s+1} e^{-s}$	Original	Original	Original
$G_2(s) = \frac{3}{s+1} e^{-s}$	300%	-	-
$G_3(s) = \frac{3}{5s+1} e^{-s}$	300%	500%	-
$G_4(s) = \frac{3}{5s+1} e^{-5s}$	300%	500%	500%

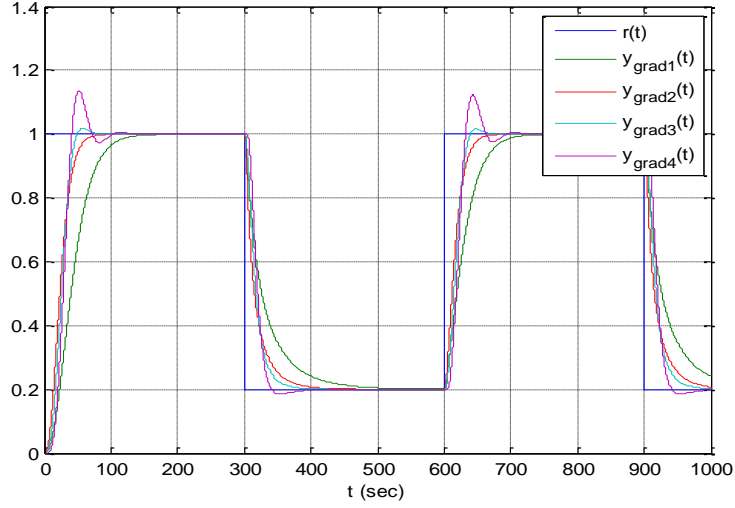
Şekil 4.2’de orjinal plant fonksiyonu $G_1(s)$ ve paramterik olarak değiştirilmiş $G_2(s)$, $G_3(s)$ ve $G_4(s)$ fonksiyonları için PI kontrol sisteminin yanıtlarını gösteren kontrol simülasyon sonuçlarını göstermektedir. $G_1(s)$ için kontrolörün tepkisi $y_{pid1}(t)$ olarak gösterilmiştir. Daha sonra aynı PI kontrolörü için plant fonksiyonları değiştirildi ve plant değişimlerinin kontrol performansı üzerindeki etkilerini gözlemlemek için simülasyonlar yapıldı. Parametrik olarak değiştirilmiş plant fonksiyonları $G_i(s)$ için PI kontrol sisteminin cevabı $y_{pidi}(t)$ olarak görülmektedir. Çizelge 4.1’de açıklanan değişimlerin etkisi kontrol cevabının bozulması olarak görülmektedir. En fazla değişimi ifade eden $G_4(s)$ fonksiyonu için sistem cevabı tamamen salınımlı hale gelmiş ve yerleşme gerçekleşmemiştir.



Şekil 4.2: Orjinal plant fonksiyonu $G_1(s)$ ve parametrik değişmiş varyantları $G_2(s)$, $G_3(s)$ ve $G_4(s)$ fonksiyonları için PI kontrol sisteminin yanıtlarını gösteren kontrol simülasyon sonuçları

Şekil 4.2’de görüldüğü üzere plant modelindeki parametrik değişimler, klasik kontrol sistemlerinin performansını çok etkileyebilir. Bunun temel nedeni değişimlere adaptasyon kabiliyetinin yetersiz olmasıdır.

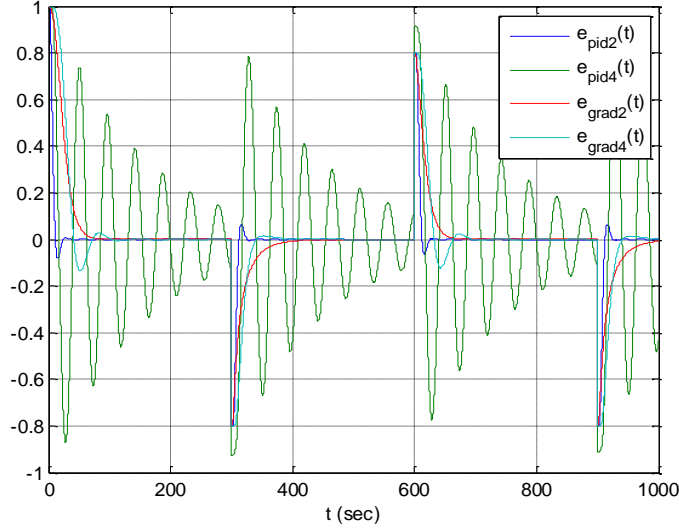
Şekil 4.3’te optimal PI kontrolünün simülasyonunda kullanılan aynı plant fonksiyonları için önerilen adaptif gradyan iniş kontrolünün cevapları sunulmuştur.



Şekil 4.3 : Orjinal plant fonksiyonu ve parametrik değişmiş varyantları için önerilen adaptif gradyan iniş kontrol sisteminin simülasyon sonuçları

Şekil 4.3'te $G_1(s)$ için önerilen adaptif kontrol yanıtı $y_{grad1}(t)$ ile gösterilmiştir. $G_1(s)$ için adaptif gradyan iniş kontrolünün zaman cevabı, Şekil 4.2'deki $G_1(s)$ için optimal PI kontrolün cevabına benzer. Daha sonra aynı gradyan iniş kontrol konfigürasyonları kullanılarak değiştirilmiş plant fonksiyonları üzerindeki kontrol performansını gözlemlemek için simülasyonlar yapılmıştır. Simülasyon sonuçlarında $G_i(s)$ için adaptif gradyan iniş kontrol sisteminin tepkileri $y_{gradi}(t)$ ile gösterilmektedir. PI kontrol ve adaptif gradyan iniş kontrol kıyaslandığında, adaptif gradyan iniş kontrolün değişen plant fonksiyonları karşısında cevabının daha az etkilendiği görülmektedir. Bu adaptif gradyan iniş kontrol yönteminin değişen plant durumlarına daha iyi adaptasyon gösterebildiğine işaret etmektedir. Şekil 4.2'de $G_4(s)$ plantı için PI kontrol salınımlı nedeni ile yerleşemezken, Şekil 4.3'te $G_4(s)$ plantı için adaptif gradyan iniş kontrol biraz aşım ile yerleşebilmiştir.

Şekil 4.4'te hata işaretlerinin karşılaştırılması için $G_2(s)$ ve $G_4(s)$ plantları için her iki kontrolörün kontrol hata işaretleri karşılaştırmaktadır. PI kontrolüne kıyasla adaptif gradyan iniş kontrolünün adaptasyon kabiliyetinin daha iyi olduğu kontrol hatasındaki daha az değişimler ile açıkça görülmektedir.



Şekil 4.4 : $G_2(s)$ ve $G_4(s)$ için her iki kontrolördeki kontrol hataları

Adaptasyon kabiliyetini nümerik olarak değerlendirebilmek için Çizelge 4.2’de bu iki kontrol sisteminin cevaplarındaki Ortalama Mutlak Hata (OMH), Mean Absolute Error (MAE) değerleri listelenmiş ve bu hatalardaki değişim miktarları hesaplanmıştır. Plant değiştiği durumda ortalama mutlak hatanın çok değişmemesi adaptasyon kabiliyetinin yüksekliğini göstermektedir. PI kontrolü için MAE’ler $OMH_{pid} = \frac{1}{T} \int_0^T |y_{pidi} - y_{pid1}| dt$,

gradyan iniş kontrolü için $OMH_{pid} = \frac{1}{T} \int_0^T |y_{pidi} - y_{pid1}| dt$ formülasyonları kullanılarak

hesaplanmıştır. Plant değişikliklerinde adaptif gradyan iniş kontrolün kontrol hatasındaki değişkenliğin PI kontrole göre daha az oluşu görülmektedir. Bu adaptif gradyan iniş kontrolünün adaptasyon kabiliyeti nedeni ile daha dayanıklı kontrol performansı sunabileceğini göstermektedir.

Çizelge 4.2 : Plant fonksiyonlarının ortalama mutlak hataları (OMH).

Plant Perturbations	OMH_{pid}	OMH_{grad}
$G_2(s)$	0.0434	0.0484
$G_3(s)$	0.0452	0.0489
$G_4(s)$	0.2635	0.0555
$G_2(s)$ 'den $G_4(s)$ 'de OMH performansında değişim	507%	13%

5. BİRİNCİ MERTEBE ZAMAN GECİKMELİ SİSTEM DİNAMİĞİNİN GRADYAN İNİŞ YÖNTEMİNE DAYALI ADAPTİF IIR FİLTRELER İLE MODELLENMESİ

Optimizasyon metotları, mühendislik sorunlarının çözümü için önem teşkil eder [101]. Problemlere çözüm bulmak için üzerinde çalışılan sistemi modellemek, optimizasyon metotlarının kullanıldığı temel bir alandır. Sistem modellemesinde, gerçek sistemi en az hata ile yansıtan model parametrelerinin belirlenmesi önemli bir optimizasyon sorunudur [102]. Matematiksel modelleme sistemin tasarımını ve sistem analizini için önemlidir.

Günümüzde matematiksel sistemlerin modellerinin ihtiyaç duyduğu alanlardan bazıları veri analizi [103, 104, 105], makine öğrenmesi [106, 52], dinamik sistem modellemesi [107, 108, 22], kontrol sistemi tasarımı [109, 110] sayılabilir. İteratif bir şekilde nümerik çözümler sağlayan gradyan iniş yöntemini mühendislik problemlerinin çözümünde kullanılmıştır [53]. Örneğin, modellemede [19], robotik çalışmalarında [2], yapay sinir ağlarında [6, 3, 4, 7, 8, 9], eniyileme algoritmalarında [12], adaptif filtrelerde [20, 50], bilgisayarla görü çalışmalarında [10, 11], derin öğrenme çalışmalarında [13] gibi pek çok alanda uygulamaları yapılmıştır.

Gradyan iniş yöntemiyle literatürde FIR ve IIR filtre tasarımları ile ilgili çok sayıda çalışma yapılmıştır [20, 50, 111, 112, 113]. Bu bölümde kontrol sistemlerinin tasarımında sıklıkla kullanılan temel bir model olan Birinci Mertebe Zaman Gecikmeli (BMZG) sistem dinamiğinin gradyan iniş optimizasyonu ile Sonuz Impulse Cevabı (IIR) filtre formunda online giriş-çıkış verisi yardımı ile modellenmesi üzerine çalışılmıştır. Böylece, sistemden alınacak giriş-çıkış verileri yardımıyla gerçek zamanlı olarak bir BMZG sisteminin IIR filtre formunda modelinin oluşturulması hedeflenmiştir. Geliştirilen bu adaptif IIR model gelecek çalışmalarda adaptif kontrollör için bir iç-model olarak kullanılması mümkün olacaktır. Bir önceki bölümde dinamik sistem modellemesi için zamanla değişen FIR filtre modeli kullanılmıştı. Bu bölümde BMZG sistem dinamiğini daha iyi temsil edebilme potansiyeline sahip zamanla değişen IIR filtre modelinin tasarımı incelenmiştir. Gradyan iniş yöntemi ile tasarlanan adaptif IIR filtrenin BMZG sistem dinamiğini ne ölçüde temsil edebileceği araştırılmış ve simülasyon sonuçları sunulmuştur. Burada temel zorluk yüksek zaman gecikmesinin zamanla değişen IIR filtre ile ne ölçüde temsil edilebileceği sorusunun cevabının araştırılmasıdır. Bu konuyu araştırmak için de farklı zaman gecikmeli BMZG sistemlerinin simülasyonları gerçekleştirilmiştir. Simülasyonlar için Matlab

Simülank ortamı kullanılmıştır. Simülasyonlarda BMZG sistemlerden gelen veriler ile adaptif IIR filtre modeli online optimize edilmiş ve adaptif IIR filtre cevabının BMZG sistem cevabına yakınsama performansı incelenmiştir.

5.1 Teorik Arka Plan

Temel nümerik gradyan iniş yöntemi [53],

$$x_i[n+1] = x_i[n] - \eta_i \frac{\partial E}{\partial x_i} \quad (5.1)$$

ile ifade edilmiştir. E hata fonksiyonu genelde hataların karesel toplamı olarak ele alınır [19, 20],

$$E = \frac{1}{2} e^2 \quad (5.2)$$

Denklem (5.2)'deki e anlık hata değeridir ve şöyle ifade edilir.

$$e = d - y \quad (5.3)$$

Denklem (5.3)'te d istenen değer ve y sistemin çıkış değeridir.

IIR filtreleri çıkış değerlerini özyinelemeli olarak girişte kullanan ayrık zamanlı bir sistemi modeller. Çıkıştan girişe doğru geri besleme yapımlarından ötürü geribeslemeli filtreler olarak adlandırılırlar. İşte bu nitelikleri sebebiyle dinamik sistemlerin modellenmesinde kullanılmaktadırlar [114].

IIR filtreler ayrık zaman bölgesinde ayrık fark denklemleri ile,

$$y[n] = -\sum_{i=1}^L a_i y[n-i] + \sum_{j=0}^M b_j x[n-j] \quad (5.4)$$

formunda ifade edilirler [20].

Adaptif IIR filtrelerinin katsayıları filtrenin istenen bir cevaba yakınsaması için optimal bir şekilde belirlenir. Bu noktada gradyan iniş yöntemi, optimal IIR katsayılarının belirlenmesi için sıklıkla kullanılmıştır[20]. Optimal IIR katsayılarının belirlenebilmesi en küçük ortalama kareler (Least Mean Square (LSM)) yöntemi önerilmiştir. Bu yöntem şöyle özetlenebilir [20]. Filtre katsayı vektörleri θ ve veri vektörü $X(n)$,

$$\theta = [a_1 \ a_2 \ \dots \ a_L \ b_0 \ b_1 \ \dots \ b_M] \quad (5.5)$$

$$X(n) = [y(n-1) \ y(n-2) \ \dots \ y(n-L) \ x(n-1) \ x(n-2) \ \dots \ x(n-M)] \quad (5.6)$$

ile tanımlanmış ve n . andaki filtre çıkışı,

$$y(n) = \theta(n)^T X(n) \quad (5.7)$$

olarak ifade edilmiştir. Hata fonksiyonu $E = \frac{1}{2} e^2 = \frac{1}{2} (d - y)^2$ için, Gİ yöntemine göre filtre katsayı güncellemeleri,

$$\theta(n+1) = \theta(n) + \gamma(d - y)\nabla_{\theta} y(n) \quad (5.8)$$

ifade edilmiştir. Burada gradyan $\nabla_{\theta} y(n)$ işlemi,

$$\nabla_{\theta} y(n) = X(n) + \sum_{i=1}^L a_i \nabla_{\theta} y(n) \quad (5.9)$$

ile ifade edilmiştir. Aşağıda bu yöntem açık formda çözümlenmekte ve kontrol sistemi modellenmesi için MATLAB/Simulink ortamında gerçekleştirilmektedir.

5.2 Adaptif IIR Filtre Yapısının Gradyan İniş Yöntemi Çözümlemesi

Bu bölümde ayrık zaman bölgesinde BMZG sistemlerinin cevabına yakınsamak için kullanılan adaptif IIR filtre modelini oluşturmak için bir önceki bölümde denklem (5.1) ile ifade edilen nümerik Gİ çözümü uygulanmaktadır. Bir BMZG dinamik sistem modelinin girişi $u[n]$ ve çıkışı $d[n]$ ile gösterildiği varsayalım. Denklem (5.4)'teki IIR filtre fonksiyonunun bu sistemi tam temsil edebilmesi için iki sistem çıkışının farkının zamanla sifıra yakınsaması beklenir. Dolayısı ile n ayrık zaman indeksi olmak üzere aşağıdaki karesel hata fonksiyonunun minimize edilmesi gerekmektedir.

$$E_m = \frac{1}{2} (d[n] - y[n])^2 = \frac{1}{2} (d[n] + \sum_{i=1}^L a_i y[n-i] - \sum_{j=0}^M b_j u[n-j])^2 \quad (5.10)$$

Her zaman pozitif değerler alabilen bu hata fonksiyonu sifıra yaklaşırsa adaptif IIR filtre çıkışı $y[n]$ aynı $u[n]$ girişi durumunda dinamik sistem çıkışı $d[n]$ 'e yakınsayabilir. Bunun için denklem (5.10) ile ifade edilen hata fonksiyonunu minimize eden a_i ve b_j filtre katsayılarının gradyan iniş yöntemi ile belirlenmesi gerekmektedir. Burada gradyan iniş çözümünü a_i katsayıları için yazalım,

$$a_i[n+1] = a_i[n] - \eta_a \frac{\partial E_m}{\partial a_i[n]} \quad (5.11)$$

Türev ifadesi denklem (5.10) dikkate alınarak

$$\frac{\partial E_m}{\partial a_i[n]} = y[n-i](d[n] + \sum_{i=1}^L a_i[n]y[n-i] - \sum_{j=0}^M b_j[n]u[n-j]) \quad (5.12)$$

elde edilir. Nümerik gradyan iniş dinamiğini b_j katsayısı için uygulanır ise

$$a_i[n+1] = a_i[n] - \eta_b \frac{\partial E_m}{\partial b_j[n]} \quad (5.13)$$

Türev ifadesi denklem (5.10) dikkate alınarak yazılırsa,

$$\frac{\partial E_m}{\partial b_j[n]} = -u[n-j](d[n] + \sum_{i=1}^L a_i[n]y[n-i] - \sum_{j=0}^M b_j[n]u[n-j]) \quad (5.14)$$

elde edilir. Denklemler düzenlenirse ayrık zamanda $a_i[n]$ ve $b_j[n]$ elde edilir.

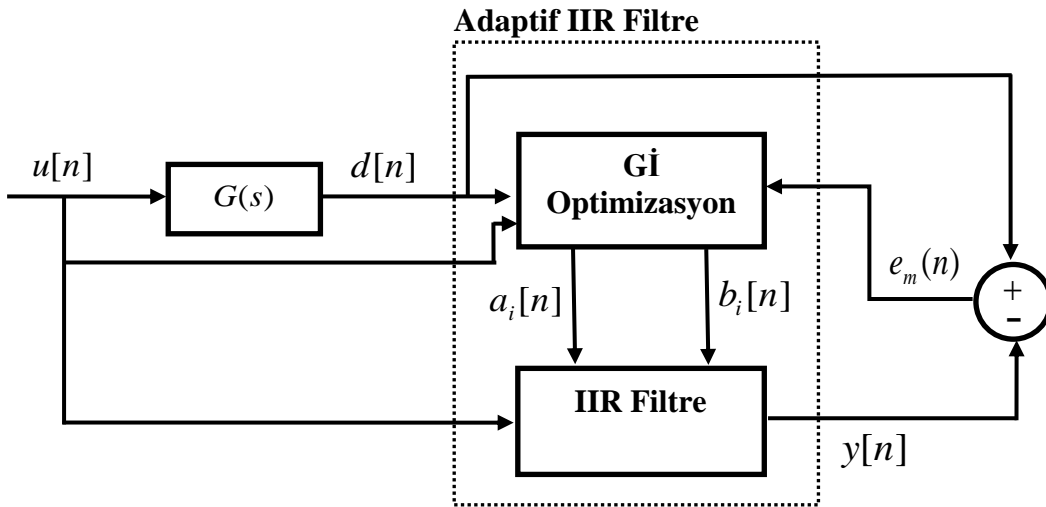
$$a_i[n+1] = a_i[n] - \eta_a y[n-i](d[n] + \sum_{i=1}^L a_i[n]y[n-i] - \sum_{j=0}^M b_j[n]u[n-j]) \quad (5.15)$$

$$b_j[n+1] = b_j[n] + \eta_b u[n-j](d[n] + \sum_{i=1}^L a_i[n]y[n-i] - \sum_{j=0}^M b_j[n]u[n-j]) \quad (5.16)$$

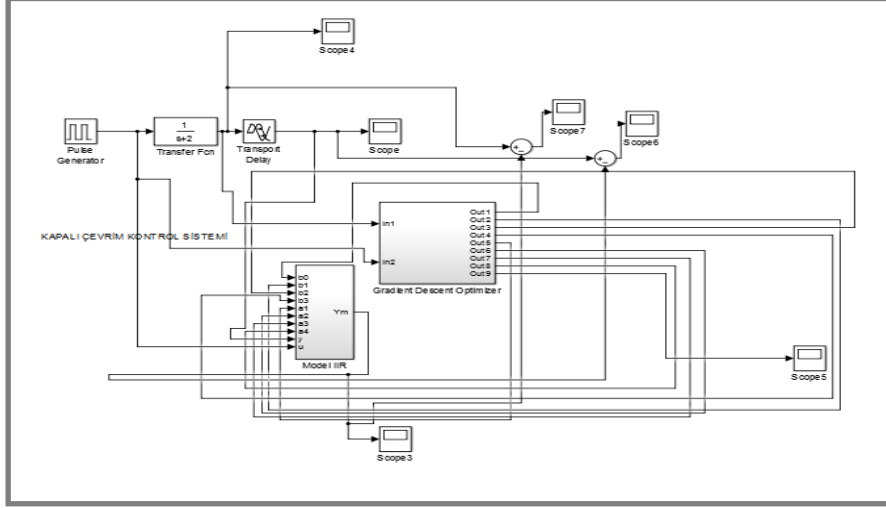
ile ifade edilir.

5.3 Nümerik Çalışma

Bu bölümde bir önceki bölümde elde edilen adaptif IIR filtre Gradyan iniş çözümlerinin $G(s) = \frac{K}{\tau s + 1} e^{-Ls}$ ile ifade edilen BMZG sistem dinamiğine yakınsama performansını incelemektedir. Bu amaçla Şekil 5.1'de blok diyagramı verilen sistem MATLAB/Simulink ortamında tasarlanmıştır. Geliştirilen simulink simülasyon modeli Şekil 5.2'de sunulmuştur.



Şekil 5.1 : Dinamik sistem modellemesi için geliştirilen sistemin blok diyagramı

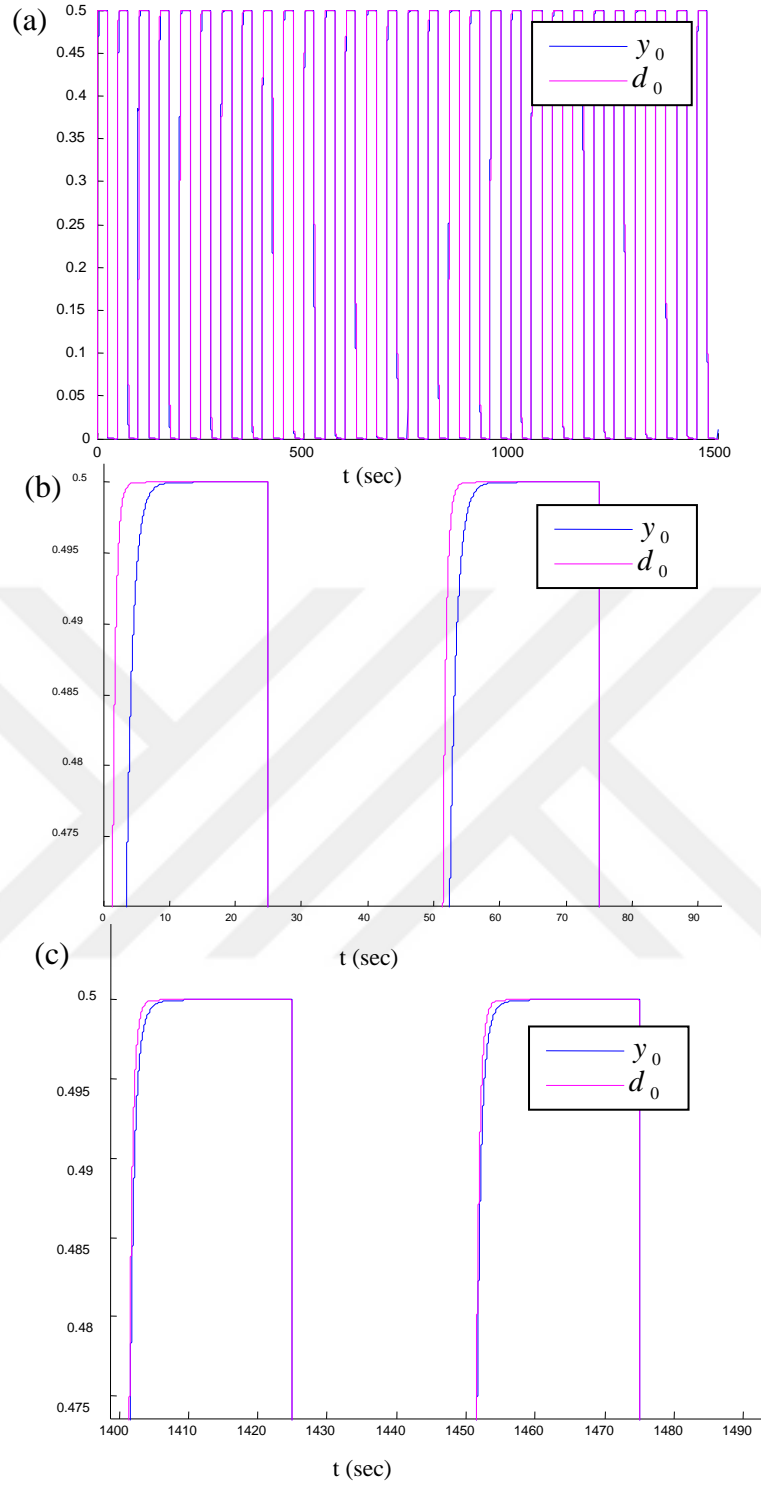


Şekil 5.2 : BMZG sistem modellemesi için geliştirilen simulink simülasyon ortamı

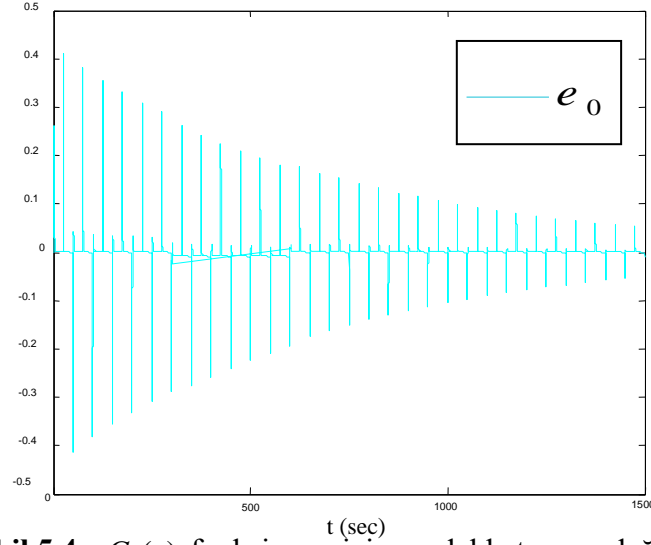
BMZG sistem modellerindeki zaman gecikmesi katsayısı e^{-Ls} sistem cevabının zamanda ötelenmesine neden olur ve dinamik sistemlerin ayrık zaman filtreler ile modelleme performansını olumsuz etkiler. Bu çalışmada IIR filtre ile modelleme performansının zaman gecikmesinin artışıyla nasıl etkilendiği incelenmiştir. Bu etkiyi incelemek için Çizelge 5.1’de görülen BMZG sistem fonksiyonları için simülasyonlar yapılmış ve sonuçlar değerlendirilmiştir. Bu amaçla $G_0(s)$, $G_1(s)$, $G_5(s)$, $G_{10}(s)$ ve $G_{20}(s)$ modellerinin kare dalga girişi için ayrı ayrı simülasyonları yapılmış ve bu modellere IIR filtre çözümlerinin yakınsaması sağlanmıştır. Elde edilen filtre çıkışları y_0 , y_1 , y_5 , y_{10} ve y_{20} , BMZG sistem çıkışları d_0 , d_1 , d_5 , d_{10} ve d_{20} ile karşılaştırılmıştır. Burada indisler zaman gecikmesi sürelerini temsil etmektedir.

Çizelge 5.1 : Farklı zaman gecikmesine sahip BMZG sistem modelleri.

BMZG Sistem Fonksiyonları	Zaman Gecikmesi
$G_0(s) = \frac{1}{s+2} e^0$	0 sn (Gecikmesiz)
$G_1(s) = \frac{1}{s+2} e^{-1}$	1 sn
$G_5(s) = \frac{1}{s+2} e^{-5}$	5 sn
$G_{10}(s) = \frac{1}{s+2} e^{-10}$	10 sn
$G_{20}(s) = \frac{1}{s+2} e^{-20}$	20 sn

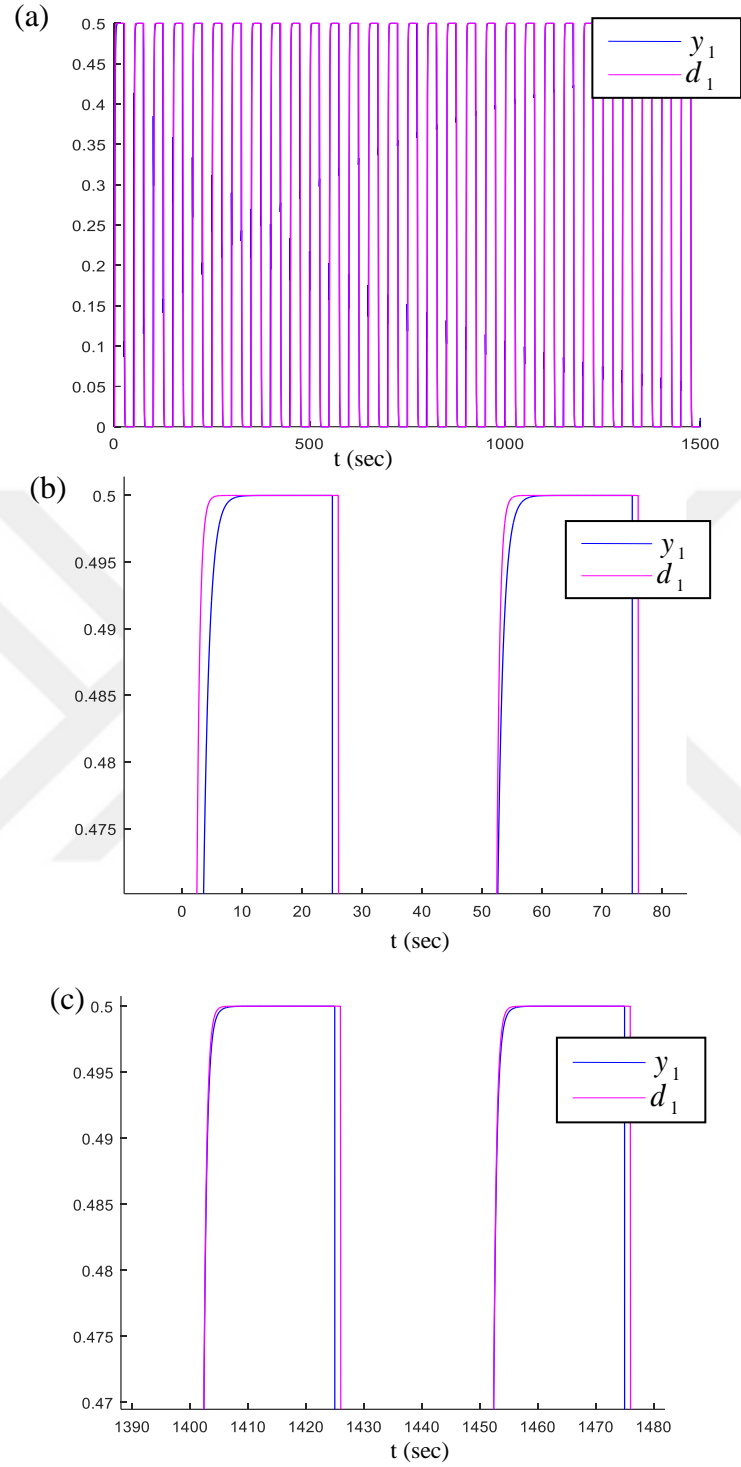


Şekil 5.3 : Kare dalga giriş işareti için $G_0(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki başlangıç anı cevapları, (c) 1400 - 1500 sn aralığındaki cevaplar

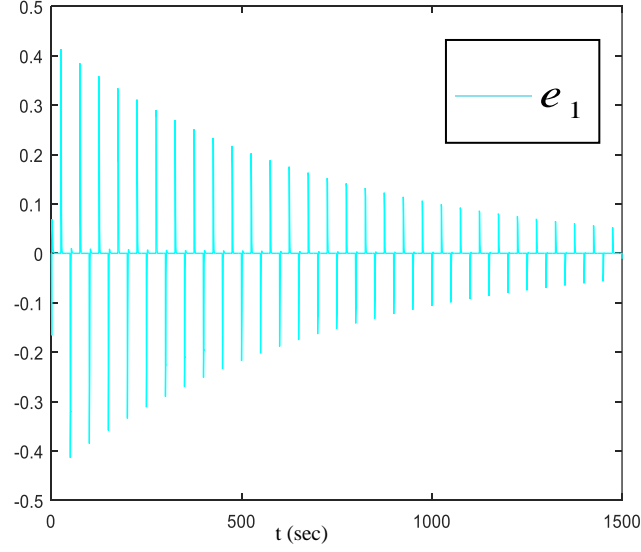


Şekil 5.4 : $G_0(s)$ fonksiyonu için model hatasının değişimi

Şekil 5.3'te kare dalga giriş için $G_0(s)$ ile modellenen BMZG dinamik sistem modeli cevabının adaptif IIR filtre ile online modellenmesi simülasyon sonuçları görülmektedir. Şekilde, adaptif filtre çıkışı y_0 'ın BMZG sistem çıkışı d_0 'a yakınsaması gösterilmiştir. Şekil 5.3 (b)'de simülasyonun başında 0-90 saniyesinde sistem cevapları görülmektedir. İkinci yükselen kenarda filtre çıkışının dinamik sistem çıkışına yakınsamaya başladığı görülmektedir. Şekil 5.3 (c)'de simülasyonun sonlarına doğru 1400-1500 saniye arası için sonuçlar verilmiştir. Burada adaptif filtrenin $G_0(s)$ sistem cevabı arasındaki farkın başlangıç anlarına göre oldukça azaldığı ve filtre cevabının $G_0(s)$ cevabına yakınsadığı açıkça görülmektedir. Şekil 5.4'te e_0 modelleme hatasının zamanla azaldığını ve modellenenin gerçekleştirildiğini göstermektedir.

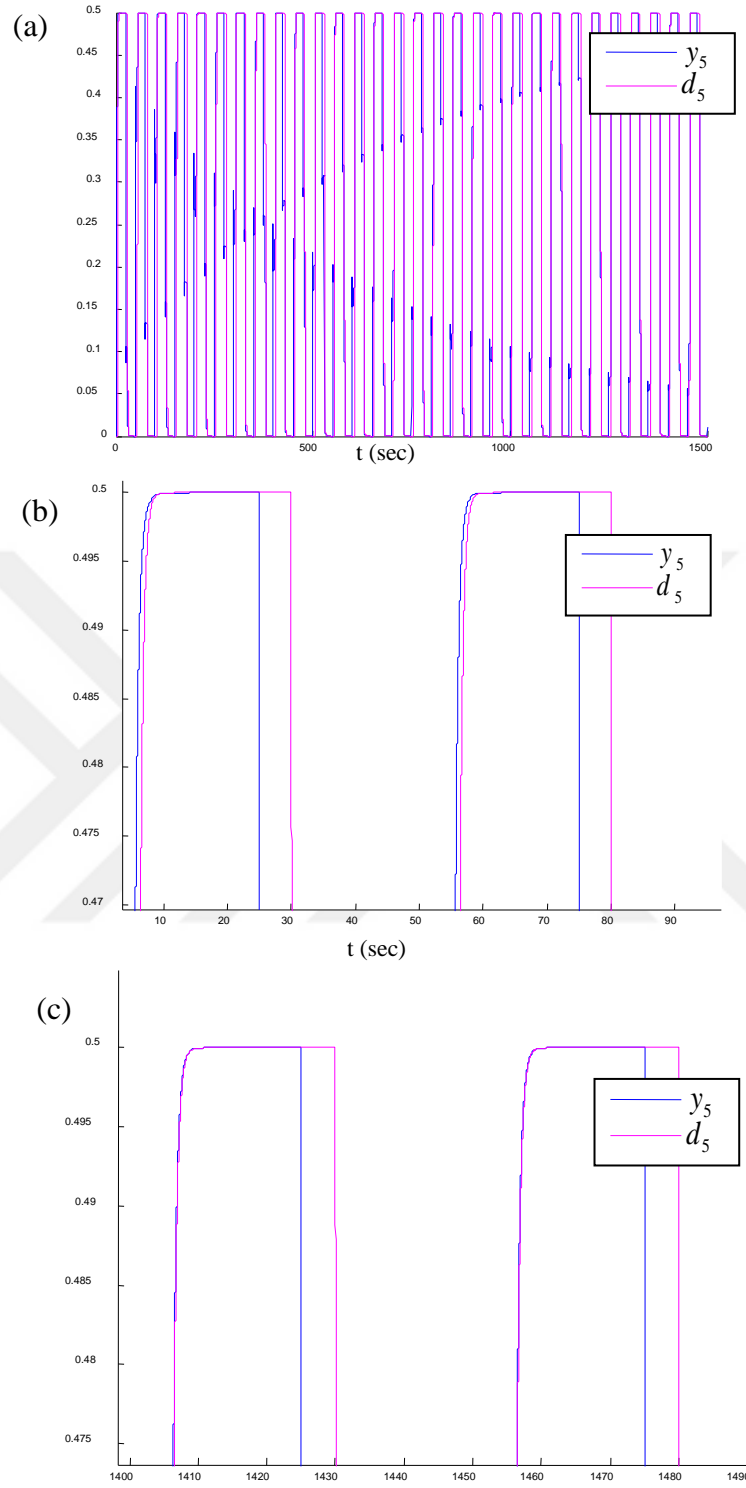


Şekil 5.5 : Kare dalga giriş işareti için $G_1(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki yakın görüntüsü, (c) 1400 - 1500 sn aralığındaki yakın görüntüsü

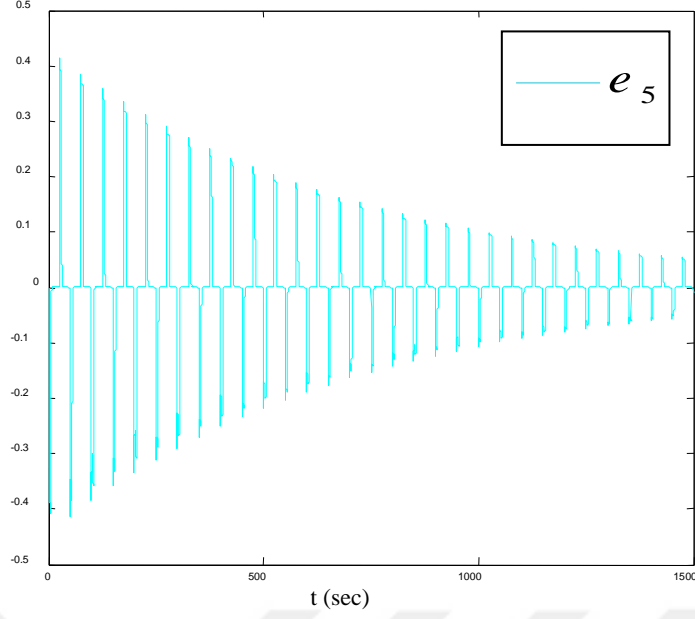


Şekil 5.6 : $G_1(s)$ fonksiyonu için model hatasının değişimi

Şekil 5.5'te kare dalga giriş için $G_1(s)$ BMZG dinamik sistem modeli cevabının adaptif IIR filtre ile online modelleme simülasyon sonuçları görülmektedir. Burada $G_1(s)$ dinamik sistem modeli 1 sn zaman gecikmesine sahiptir. Şekil 5.5 (b) ve (c) kıyaslandığı zaman IIR filtrenin adapte olduğu ve modelleme performansını iyileştirdiği görülmektedir. Şekil 5.6'daki model hatasının değişimi bu sonucu desteklemektedir.

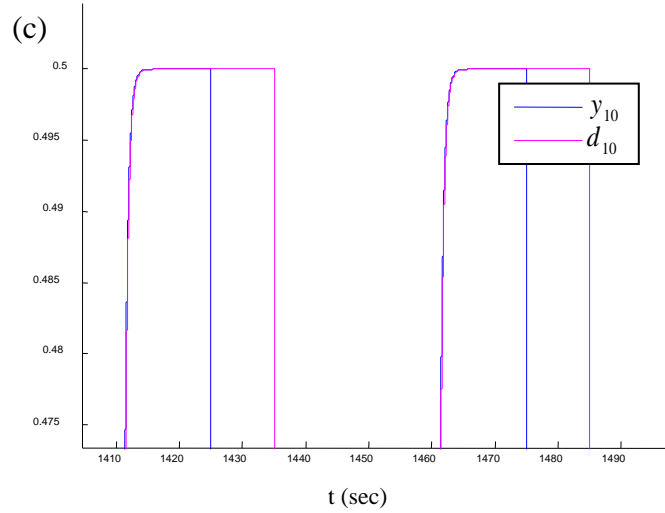
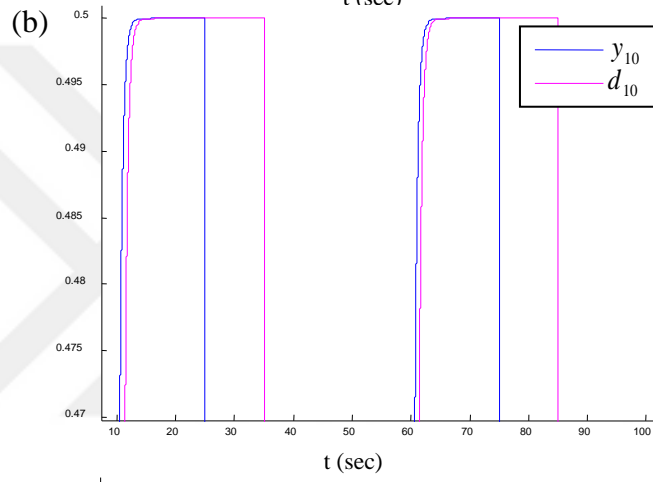
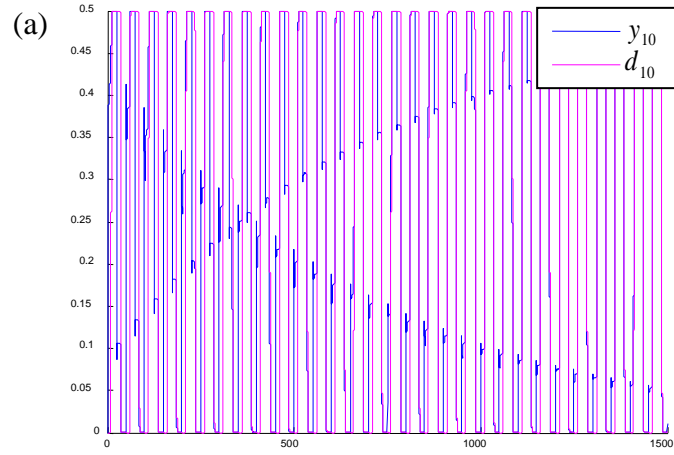


Şekil 5.7 : Kare dalga giriş işareti için $G_5(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki yakın görüntüsü, (c) 1400 - 1500 sn aralığındaki yakın görüntüsü

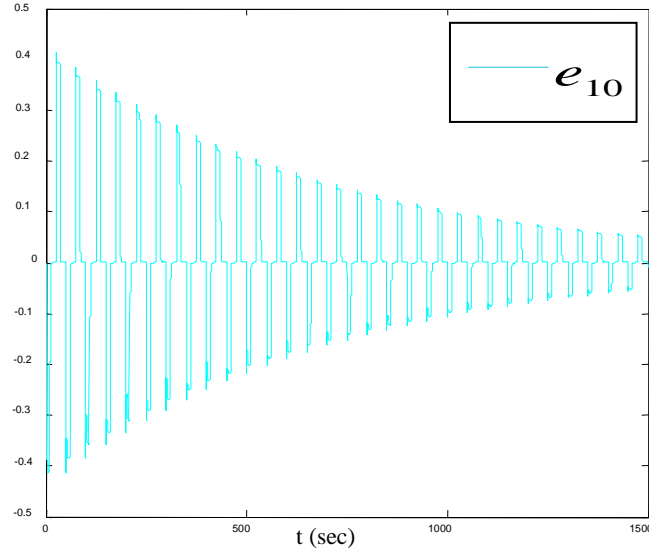


Şekil 5.8 : $G_5(s)$ fonksiyonu için model hatasının değişimi

Şekil 5.7’de kare dalga giriş için $G_5(s)$ ile ifade edilen BMZG dinamik sistem modeli cevabının adaptif IIR filtre ile modelleme çalışması simülasyon sonuçları görülmektedir. Burada $G_5(s)$ dinamik sistem modeli 5 sn zaman gecikmesine sahiptir. Şekil 5.7 (b) ve (c) kıyaslandığı zaman 0-90 saniye aralığında y_5 ile d_5 arasındaki farkın 1400-1500 saniye aralığındaki değerlere göre daha fazla olduğu görülmektedir. Bu simülasyon sonuçlarına göre dikkat çeken bir nokta 5 sn zaman gecikmesi durumunda adaptif filtrenin yükselen kenara yakınsamayı tercih etmesi ve düşen kenarda filtre cevabının dinamik sistem cevabından uzaklaşmasıdır. Bu durum Çizelge 5.2’de görülen zaman gecikmesi arttıkça model hatası parametrelerinin artışı açıklamaktadır. Şekil 5.8’de model hatasının değişimi adaptif filtre çıkışı y_5 ’in dinamik sistem çıkışı d_5 ’e zamanla yakınsadığını göstermektedir.

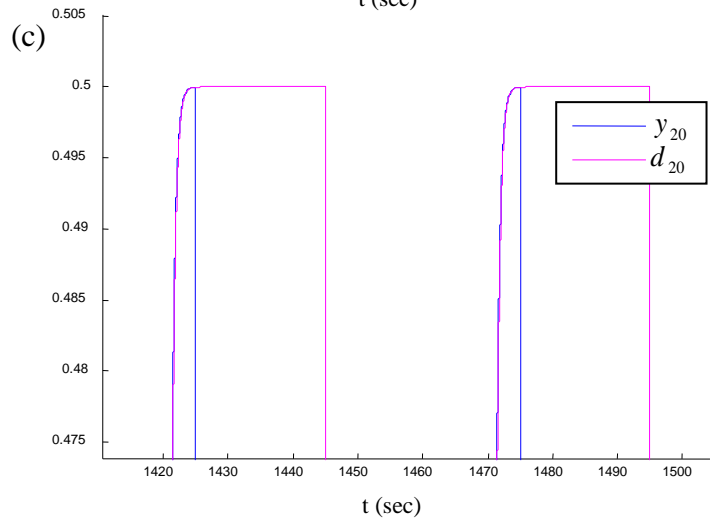
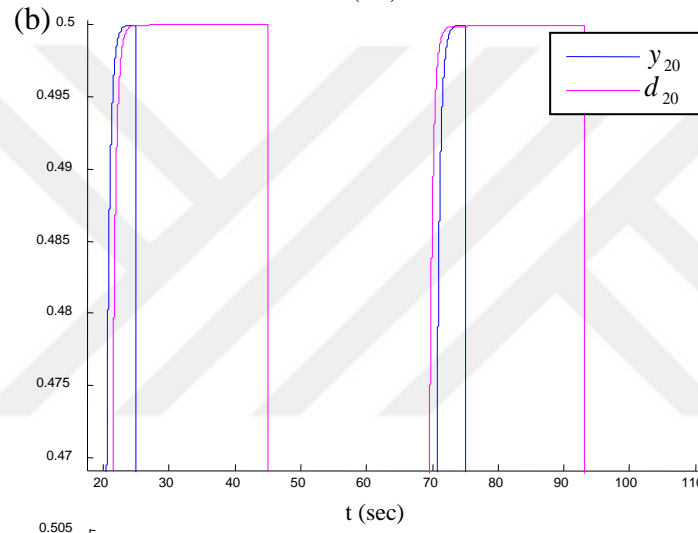
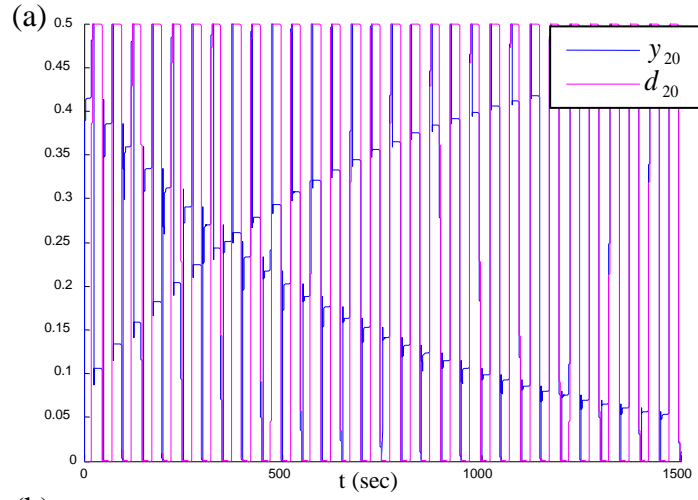


Şekil 5.9 : Kare dalga giriş işareti için $G_{10}(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki yakın görüntüsü, (c) 1400 - 1500 sn aralığındaki yakın görüntüsü

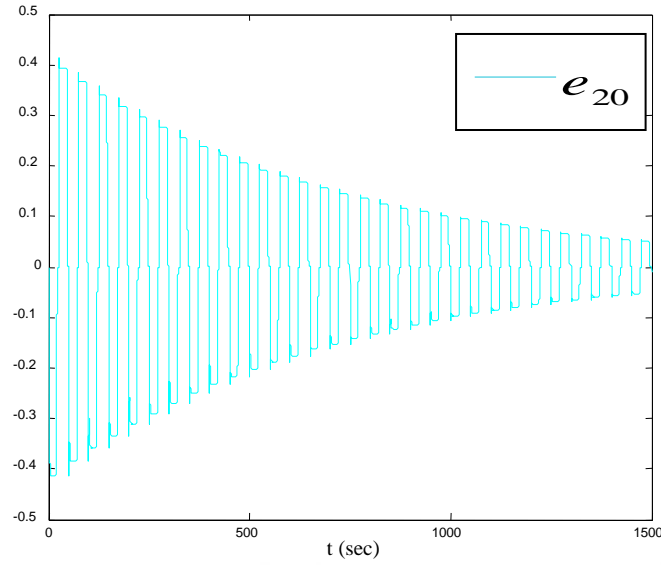


Şekil 5.10 : $G_{10}(s)$ fonksiyonu için model hatasının değişimi

Şekil 5.9'de kare dalga giriş için $G_{10}(s)$ BMZG dinamik sistem modeli cevabının adaptif IIR filtre ile online modelleme çalışması simülasyon sonuçları görülmektedir. $G_{10}(s)$ dinamik sistem modeli zaman gecikmesi 10 saniyedir. Artan zaman gecikmesi adaptif filtrenin cevabının dinamik sistem cevabına yakınsamasını zorlaştırmaktadır. Şekil 5.9 (b) ve (c) kıyaslandığı zaman 0-90 saniye aralığında y_{10} ile d_{10} arasındaki farkın 1400-1500 saniye aralığındaki değerlere göre daha fazla olduğu görülmektedir. Artan zaman gecikmesinin olumsuz etkisi adaptif filtrenin yükselen kenara yakınsamayı tercih etmesi ve düşen kenarda filtre cevabının dinamik sistem cevabından uzaklaşmasının artışıdır. Bu durum Çizelge 5.2'de görülen model hatası parametrelerinin zaman gecikmesi ile artışını açıklamaktadır. Şekil 5.10'da model hatasının değişimi adaptif filtre çıkışı y_{10} 'un dinamik sistem çıkışı d_{10} 'e zamanla yakınsadığını göstermektedir.



Şekil 5.11 : Kare dalga giriş işareti için $G_{20}(s)$ fonksiyonu çıkışı ve bu fonksiyonu modelleyen adaptif IIR filtre çıkışları; (a) tam simülasyon sonucu, (b) 0-90 sn aralığındaki yakın görüntüsü, (c) 1400 - 1500 sn aralığındaki yakın görüntüsü



Şekil 5.12 : $G_{20}(s)$ fonksiyonu için model hatasının değişimi

Şekil 5.11’de kare dalga giriş için $G_{20}(s)$ BMZG dinamik sistem modeli cevabının adaptif IIR filtre ile online modellemesinin simülasyon sonuçları görülmektedir. $G_{20}(s)$ dinamik sistem modeli zaman gecikmesi 20 saniyedir. Artan zaman gecikmesi adaptif filtrenin cevabının dinamik sistem cevabına yakınsamasını zorlaştırmaktadır. Artan zaman gecikmesinin düşen kenarda filtre cevabının dinamik sistem cevabından uzaklaşmasına yol açtığı açıkça görülmektedir. Ancak, Şekil 5.11’de model hatasının zamanla azalması sistem cevaplarının birbirine yaklaştığını göstermektedir.

Çizelge 5.2’de $G_0(s)$, $G_1(s)$, $G_5(s)$, $G_{10}(s)$ ve $G_{20}(s)$ farklı zaman gecikmeli BMZG sistemlerin adaptif IIR filtre ile modelleme performansları sunulmuştur. Adaptif IIR formunda elde edilen sistem modelinin çıkış değeri y_L ile gerçek sistem çıkışı d_L arasındaki farkın modelleme hatasını göstermektedir. Burada L indisi BMZG sistemin zaman gecikmesini ifade etmiştir. Bu çizelgede 3 farklı hata performansı ölçütü sunulmuştur:

$$OMH = \frac{1}{p} \sum_{n=1}^p |d_L[n] - y_L[n]| \quad (5.17)$$

ile ifade edilen Ortalama Mutlak Hata (OMH) performans ölçütü,

$$OKH = \frac{1}{p} \sum_{n=1}^p (d_L[n] - y_L[n])^2 \quad (5.18)$$

ile ifade edilen Ortalama Karesel Hata (OKH) ve diğeri

$$ORH = \frac{1}{p} \sum_{n=1}^p \frac{|d_L[n] - y_L[n]|}{d_L[n]} \quad (5.19)$$

ile ifade edilen Ortalama Relatif Hata (ORH) kullanılmıştır.

Çizelge 5.2 : Farklı zaman gecikmeli BMZG sistemlerin adaptif IIR filtre ile modellenme performansları

BMZG Sistem Fonksiyonları	<i>OMH</i>	<i>OKH</i>	<i>ORH</i>
$G_0(s)$	0.0037	4.0825e-04	0.0795
$G_1(s)$	0.0092	0.0017	29.4650
$G_5(s)$	0.0365	0.0078	162.2700
$G_{10}(s)$	0.0710	0.0158	338.9000
$G_{20}(s)$	0.1399	0.0320	689.5100

6. SONUÇ VE ÖNERİLER

Bu tez çalışmasında yapılan çalışmalar sonucunda elde edilen bulgular şöyle özetlenebilir:

(i) Kontrol sistemlerinde gerçek zamanlı olarak optimal kontrol sinyallerinin üretebilmesi ve sistemi kararlı durumda tutması beklenir. Bu nedenle öncelikle gradyan iniş çözümlerinin kararlılığının incelenmesi kontrol uygulamaları açısından önem taşımaktadır. Çünkü kontrol uygulamalarında optimizasyonu tekrarlama imkanı olmamakta ve çözümlerin kararlı olması ve global minimuma yakınsaması kontrol sisteminin başarısını doğrudan etkilemektedir. Bu nedenle bu çalışmada sürekli zaman gradyan iniş çözümünün Lyapunov kararlılık durumu incelenmiş ve öğrenme katsayısının pozitif olduğu bölgede ($\forall \gamma_i > 0$) sürekli gradyan iniş çözümlerinin Lyapunov kararlı olabileceği gösterilmiştir. Örnek bir analizde De Jong fonksiyonu için nümerik gradyan iniş çözümleri ile sürekli zaman çözümleri ile karşılaştırılmış ve nümerik çözümlenin çözümün kararlılığına etkileri tartışılmıştır. Nümerik gardyan iniş çözümlerinde çok yüksek pozitif öğrenme katsayılarının çözümü kararsızlaştırdığı görülmüştür. Bu nedenle nümerik çözümlerin kararlılığı öğrenme katsayısının için pozitif değerli ve yeterince küçük değerli belirlenmesi gerektiği görülmüştür.

(ii) Nümerik gardyan iniş yöntemi çözüm kararlılığı incelendikten sonra adaptif gardyan iniş kontrol yönteminin birinci dereceden, zaman gecikmeli kararlı plant modellerinin modellemeden bağımsız adaptif kontrolü incelenmiştir. Adaptif gardyan iniş kontrol yöntemi modellemeden bağımsız kontrol için önerilen kontrol ve adaptasyon optimizasyon bloklarından oluşan ve adaptasyon özelliği için bir iç-model kullanan bir yöntemdir [23,96]. Burada kullanılan iç modelin kontrol edilen sistemin dinamiğini temsil kabiliyeti yöntemin etkinliği için önemlidir. Bu tez çalışmasında zaman gecikmeli sistemlerin daha başarılı kontrolü için anlık giriş-çıkış ilişkisini temsil eden iç model zamanla değişen FIR filtre formu olarak alınmış ve adaptif gradyan iniş çözümleri elde edilmiştir. Böylece zamanla değişen FIR model yardımı ile zaman gecikmesinin yol açtığı geciken plant dinamikleride temsil edilebilmiştir. Yöntemin performansı Matlab Simulink ortamında incelenmiş ve klasik PI kontrol ile kıyaslanarak değerlendirilmiştir. Önerilen kontrol çözümünün değişen zaman gecikmelerine adaptasyonun PI kontrolden daha iyi olduğu görülmüştür. Ancak, adaptif gardyan iniş yönteminin yapısal bir zayıflığı olan sadece kararlı plant fonksiyonları için uygulanabilir olması sorunu hala devam etmektedir.

Adaptasyon özelliđi bakımında zayıf olsa da klasik PI kontrolün kararsız plant modellerini kontrol edebilme konusunda üstünlüđü devam etmektedir. Gelecek alıřmalarda adaptif gradyan iniř özümünün kararsız plantlarda uygulanabilmesine dönük alıřmaların yapılması gerektiđi görülmüřtür.

(iii) Model bağımsız kontrol sistemi, kontrollör tasarımında plantın matematiksel modeline ihtiyaç duyulmaması durumunu ifade ediyor. Bu özellik aynı zamanda adaptasyon özelliđinin artırılması anlamına gelmektedir. Adaptif gradyan iniř kontrolünde bunu sağlamak için iç-model (internal model) kullanılmış ve adaptasyon aşaması iç-modelin cevabının kontrol edilen plant fonksiyonu yakın zamanlı cevabına yakınsaması olarak sağlanmıştır. Burada adaptasyon performansı iç-modelin dinamik plant fonksiyonlarını temsil kapasitesine bağımlıdır. Plant fonksiyonlarının zaman gecikmesi ve yüksek mertebeli dinamikleri bu konuda temel zorlukları oluşturmaktadır. Bu soruna bir özüm olabilme potansiyeli nedeni ile adaptif IIR filtrelerin zaman gecikmeli dinamik sistemleri modelleme performansı incelenmiştir. Bu amaçla adaptif IIR filtresi için gradyan iniř özümleri elde edilmiş ve matlab simulink ortamında simülasyonlar yapılmıştır. Adaptif IIR filtrelerin birinci mertebe zaman gecikmeli sistemleri modelleme performansı ve zaman gecikmesinin etkileri araştırılmıştır. Düşük zaman gecikmeleri için modelleme performansı yeterli görülebilmemesine rağmen yüksek zaman gecikmeli sistemler için adaptif IIR filtrelerin yetersiz kaldığı görülmüřtür. Gelecek alıřmalarda daha başarılı model yapılarının araştırılmasının gerektiđi görülmüřtür.

7. KAYNAKÇA

- [1] Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society*, 48(3), 334-334.
- [2] Iscimen, Bilal & Atasoy, Huseyin & Kutlu, Yakup & Yildirim, Serdar & Yildirim, Esen. (2014). Bilgisayar Görmesi ve Gradyan İniş Algoritması Kullanılarak Robot Kol Uygulaması.
- [3] KILIKÇIER, Ç , YILMAZ, E . (2018). BT Görüntülerden Akciğerin Tespiti için Süper Piksel ve Yapay Sinir Ağı Tabanlı Bir Yöntem. *Erzincan Üniversitesi Fen Bilimleri Enstitüsü Dergisi* , 11 (2) , 223-230 . DOI: 10.18185/erzifbed.384268
- [4] Turhan, K., Kurt, B., & Engin, Y. Z. (2013). Yapay Sinir Ağları ile Öğrenci Başarısı Tahmini. *Eğitim ve Bilim*, 38(170).
- [5] Akbaş, C. E., Bozkurt, A., Çetin, A. E., Çetin-Atalay, R., & Üner, A. (2015, May). Multiplication-free neural networks. In *2015 23rd Signal Processing and Communications Applications Conference (SIU)* (pp. 2416-2418). IEEE.
- [6] Amari, S. I. (1993). Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5), 185-196.
- [7] Psaltis, D., Sideris, A., & Yamamura, A. A. (1988). A multilayered neural network controller. *IEEE control systems magazine*, 8(2), 17-21.
- [8] AMASYALI, M. F. Yapay Sinir Ağlarında Parçalı Eğitim. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 8(1), 25-34.
- [9] Hochreiter, S., Younger, A. S., & Conwell, P. R. (2001, August). Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks* (pp. 87-94). Springer, Berlin, Heidelberg.
- [10] Maras, Bahri & Arica, Nafiz & Ertuzun, Aysin. (2018). An Overview of Visual Target Tracking Methods. 7. 7-19.
- [11] Karagülmez, Soner (2017). *Göz Kırpma Tabanlı Sürücü Yorgunluk Değerlendirme Sistemi* (Master's thesis). Retrieved from <https://acikerisim.uludag.edu.tr/bitstream/11452/2601/1/496122.pdf>
- [12] Toplu, M , Körpe, D . (2018). Genel İndirgenmiş Gradyan Metodu ile Eniyileme Çözücüsü Geliştirilmesi. *Sakarya University Journal of Science* , 22 (3) , 960-969 . DOI: 10.16984/saufenbilder.298972
- [13] ÖZDAĞ, M. E. (2020). *DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK ANAYOL TRAFİK ANALİZİ* (Doctoral dissertation).
- [14] Shimizu, K., Ito, S., & Suzuki, S. (1998). Tracking control of general nonlinear systems by direct gradient descent method. *IFAC Proceedings Volumes*, 31(17), 183-188.

- [15] **Jouli, Khalil & Braiek, Naceur.** (2016). A gradient descent control for output tracking of a class of non-minimum phase nonlinear systems. *Journal of Applied Research and Technology*, 14. 10.1016/j.jart.2016.09.006.
- [16] **Naiborhu, J., & Shimizu, K.** (1998). Stabilization of single input nonlinear control systems by gradient descent control algorithm. In *Theory and Practice of Control and Systems* (pp. 544-549).
- [17] **Vinagre, B. M., Petráš, I., Podlubny, I., & Chen, Y. Q.** (2002). Using fractional order adjustment rules and fractional order reference models in model-reference adaptive control. *Nonlinear Dynamics*, 29(1-4), 269-279.
- [18] **Yağmur, N., & Alagoz, B. B. (2019, April).** Comparison of Solutions of Numerical Gradient Descent Method and Continous Time Gradient Descent Dynamics and Lyapunov Stability. In *2019 27th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.
- [19] **Alagoz, B. B., Alisoy, H. Z., Koseoglu, M., & Alagoz, S.** (2017). Modeling and analysis of dielectric materials by using gradient-descent optimization method. *International Journal of Modeling, Simulation, and Scientific Computing*, 8(01), 1750014.
- [20] **Boukis, C. G., Mandic, D. P., Papoulis, E. V., & Constantinides, A. G.** (2003, April). A gradient adaptive step size algorithm for IIR filters. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*. (Vol. 6, pp. VI-85). IEEE.
- [21] **Dheeraj, K., Jacob, J., & Nandakumar, M. P. (2019).** Direct adaptive neural control design for a class of nonlinear multi input multi output systems. *IEEE Access*, 7, 15424-15435.
- [22] **Alagoz, B. B., Tepljakov, A., Ates, A., Petlenkov, E., & Yeroglu, C.** (2019). Time-domain identification of One Noninteger Order Plus Time Delay models from step response measurements. *International Journal of Modeling, Simulation, and Scientific Computing*, 10(01), 1941011.
- [23] **Alagoz, B. B., Kavuran, G., Ates, A., & Yeroglu, C.** (2017). Reference-shaping adaptive control by using gradient descent optimizers. *PloS one*, 12(11).
- [24] **Bıdıklı, B.** (2018). A Self-Tuning PID Control Method for Multi-Input-Multi-Output Nonlinear Systems. *Electrica*, 18(2), 218-226.
- [25] **Url-1** Retrieved May 22, 2020, from <https://tr.wikipedia.org/wiki/Optimizasyon>
- [26] **Winston, W. L.** (2003). *Operations Research: Applicatios and Algorithms* (I. T. Publishing (ed.); 4th ed.).
- [27] **Url-2** Metin, T. *Optimizasyon Modelleri ve Çözüm Metodları*. Retrieved May 22, 2020, from <http://home.ku.edu.tr/~mturkay/indr501/Optimizasyon.pdf>
- [28] **Leontief, W. W.** (1933). *The Use of Indifference Curves in the Analysis of Foreign Trade*. 47, 493–503.
- [29] **Leontief, W. W.** (1936). Quantitative Input and Output Relations in the Economic System of the United States. *Review of Econ. Statistics*, 18, 105–125.

- [30] **Kantorovich, L. V.** (1939). *Mathematical Methods in the Organization and Planning of Production*. *Publication House of the Leningrad State University*, 6(4).
- [31] **Kantorovich, L. V.** & Gavurin, M. K. (1940). *Primenenie matematicheskikh metodov v voprosakh analiza gruzopotokov*. *Problemy povysheniia effektivnosti raboty transporta*, 110-138.
- [32] **Koopmans, T. C.** (1949). Optimum utilization of the transportation system. *Econometrica: Journal of the Econometric Society*, 136-146.
- [33] **Koopmans, T. C.** (1951). Activity analysis of production and allocation. *Activity Analysis of Production and Allocation*.
- [34] **Koopmans, T. C.**, & Reiter, S. (1951). A model of transportation. *Activity analysis of production and allocation*, 222-259.
- [35] **Koopmans, T. C.** (1959). Three essays on the state of economic science.
- [36] **Koopmans, T. C.**, & Bausch, A. F. (1959). Selected topics in economics involving mathematical reasoning. *Siam Review*, 1(2), 79-148.
- [37] **Dantzig, G. B.** (1948, January). Programming in a linear structure. In *Bulletin of the American Mathematical Society* (Vol. 54, No. 11, pp. 1074-1074). 201 CHARLES ST, PROVIDENCE, RI 02940-2213: AMER MATHEMATICAL SOC.
- [38] **Williams, H. P.** (2013). *Model building in mathematical programming*. John Wiley & Sons.
- [39] **Evans, G. W.** (1984). An overview of techniques for solving multiobjective mathematical programs. *Management Science*, 30(11), 1268-1282.
- [40] **Url-3 Retrieved May 22, 2020, from**
<https://www.deeplearningbook.org/contents/numerical.html>
- [41] **Url-4 Retrieved May 22, 2020, from**
<https://www.scribd.com/document/119965884/Optimizasyolln-Teknikleri->
- [42][19] **Url-5 Retrieved May 22, 2020, from**
<https://web.stanford.edu/group/sisl/k12/optimization/MO-unit1-pdfs/1.1optimization.pdf>
- [43] **Url-6 Retrieved May 22, 2020, from** <https://www.vafaeijahan.com/en/wp-content/uploads/2012/02/Test-functions-for-optimization-needs.pdf>
- [44] **Mariño, P. P.** (2016). *Optimization of computer networks: modeling and algorithms: a hands-on approach*. John Wiley & Sons.
- [45] **Url-7 Retrieved May 22, 2020, from**
<https://www.deeplearningbook.org/contents/numerical.html>
- [46] **Pankaj, S., Kumar, J. S., & Nema, R. K.** (2011). Comparative analysis of MIT rule and Lyapunov rule in model reference adaptive control scheme. *Innovative Systems Design and Engineering*, 2(4), 154-162.
- [47] **Bengio, Y.** (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade* (pp. 437-478). Springer, Berlin, Heidelberg.

- [48] **Breuel, T. M.** (2015). The effects of hyperparameters on SGD training of neural networks. *arXiv preprint arXiv:1508.02788*.
- [49] **Smith, L. N.** (2017, March). Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 464-472). IEEE.
- [50] **Bernard, W., & Samuel, D. S.** (1985). Adaptive signal processing. *Englewood Cliffs, NJ: Prentice Hall*.
- [51] **George, A. P., & Powell, W. B.** (2006). Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine learning*, 65(1), 167-198.
- [52] **Duchi, J., Hazan, E., & Singer, Y.** (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul), 2121-2159.
- [53] **Ruder, S.** (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
- [54] **Chinrungrueng, C., & Sequin, C. H.** (1995). Optimal adaptive k-means algorithm with dynamic adjustment of learning rate. *IEEE Transactions on neural networks*, 6(1), 157-169.
- [55] **Plagianakos, V. P., Sotiropoulos, D. G., & Vrahatis, M. N.** (1998, February). An improved backpropagation method with adaptive learning rate. In *Proceeding of the 2nd Intern. Confer. on: Circuits, Systems and Computers*.
- [56] **Qian, N.** (1999). On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1), 145-151.
- [57] **Url-8** Retrieved May 22, 2020, from <https://int8.io/comparison-of-optimization-techniques-stochastic-gradient-descent-momentum-adagrad-and-adadelat/>
- [58] **Zeiler, M. D.** (2012). Adadelat: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- [59] **Tieleman, T., & Hinton, G.** (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 26-31.
- [60] **Kingma, D. P., & Ba, J.** (2015). Adam: A Method for Stochastic Optimization. international conference on learning representations (2015).
- [61] **Yazan, E., & Talu, M. F.** (2017, September). Comparison of the stochastic gradient descent based optimization techniques. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)* (pp. 1-5). IEEE.
- [62] **Çarkacı, N.** (2018). Derin öğrenme uygulamalarında en sık kullanılan Hiper parameteler.
- [63] **Kurt, F.** (2018). *Evrışimli Sinir Ağlarında Hiper Parametrelerin Etkisinin İncelenmesi* (Master's thesis, Eğitim Bilimleri Enstitüsü).
- [64] **Salim, S. N. S., & Zainon, M.** (2010). *Control systems engineering*. Penerbit Universiti Teknikal Malaysia Melaka.
- [65] **Kuo, B. C.** (1987). *Automatic control systems*. Prentice Hall PTR.

- [66] **Alagoz, B. B.** (2018) Kontrol Sistemlerinde Kararlılık Ders Notu: Lyapunov Kararlılık ve Lyapunov Enerji Fonksiyonu, Bilgisayar Mühendisliği, İnönü üniversitesi
- [67] **Khater, A. A., El-Nagar, A. M., El-Bardini, M., & El-Rabaie, N. M.** (2018). Adaptive T–S fuzzy controller using reinforcement learning based on Lyapunov stability. *Journal of the Franklin Institute*, 355(14), 6390-6415.
- [68] **Banakar, A.** (2011). Lyapunov stability analysis of gradient descent-learning algorithm in network training. *ISRN Applied Mathematics*, 2011.
- [69] **Wilson, A.** (2018). *Lyapunov arguments in optimization*. University of California, Berkeley.
- [70] **Serim, A. B. Ö., & Bayram, M.** LYAPUNOV METODU İLE FARK DENKLEMLERİNİN KARARLILIGININ BELİRLENMESİ. *Dumlupınar Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, (013), 31-38.
- [71] **Yagmur, N., & Alagoz, B. B.** (2019, September). Adaptive Gradient Descent Control of Stable, First Order, Time-delay Dynamic Systems According to Time-Varying FIR Filter Model Assumption. In *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)* (pp. 1-5). IEEE.
- [72] **Mazenc, F., & Niculescu, S. I.** (2001). Lyapunov stability analysis for nonlinear delay systems. *Systems & Control Letters*, 42(4), 245-251.
- [73] **Mengüç, E. C., & Acir, N.** (2012). Lyapunov kararlılık teorisi tabanlı hızlı sonsuz darbe cevaplı adaptif filtre tasarımı. *Otomatik Kontrol Türk Milli Komitesi 2012 Ulusal Toplantısı (TOK 2012)*, 768-772.
- [74] **Boyd, S.** (2008). Lecture 12: Basic Lyapunov Theory. *Electrical Engineering*, 363.
- [75] **Shuai, Z., & van den Driessche, P.** (2013). Global stability of infectious disease models using Lyapunov functions. *SIAM Journal on Applied Mathematics*, 73(4), 1513-1532.
- [76] **Shimizu, K., K. S. and H. N.** (1993). *Direct gradient method for nonlinear dynamical systems*.
- [77] **Shimizu, K., Nukumi, H., & Ito, S.** (1995). Direct steepest descent control of nonlinear dynamical systems. *IFAC Proceedings Volumes*, 28(14), 801-806.
- [78] **Naiborhu, J., Nababan, S. M., Saragih, R., & Pranoto, I.** (2005). Direct Gradient Descent Control and Sontag's Formula on Asymptotic Stability of General Nonlinear Control System. *International Journal of Control, Automation, and Systems*, 3(2), 244-251.
- [79] **Shimizu, K., Otsuka, K., & Naiborhu, J.** (1999). Improved direct gradient descent control of general nonlinear systems. In *1999 European Control Conference (ECC)* (pp. 2006-2011). IEEE.
- [80] **Shimizu, K., & Otsuka, K.** (1999, August). Performance improvement of direct gradient descent control for general nonlinear systems. In *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)* (Vol. 1, pp. 699-706). IEEE.

- [81] Shimizu, K., Otsuka, K., & Yamaguchi, T. (2002, May). Direct gradient descent control of general nonlinear systems. In *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)* (Vol. 1, pp. 833-838). IEEE.
- [82] Naiborhu, J., Nababan, S. M., Saragih, R., & Pranoto, I. (2006). Direct gradient descent control as a dynamic feedback control for linear system. *Bulletin of the Malaysian Mathematical Sciences Society*, 29(2).
- [83] Datta, A., & Ioannou, P. A. (1994). Performance analysis and improvement in model reference adaptive control. *IEEE Transactions on Automatic Control*, 39(12), 2370-2387.
- [84] Hsu, L. (1990). Variable structure model-reference adaptive control (VS-MRAC) using only input and output measurements: the general case. *IEEE Transactions on Automatic Control*, 35(11), 1238-1243.
- [85] Goodwin, G. C., & Mayne, D. Q. (1987). A parameter estimation perspective of continuous time model reference adaptive control. *Automatica*, 23(1), 57-70.
- [86] Osburn, P. V., Whitaker, H. P., & Kezer, A. (1961). Comparative studies of model reference adaptive control systems. *Institute of Aeronautical Sciences, Paper*, (61-39).
- [87] Åström, K. J., & Wittenmark, B. (2013). *Adaptive control*. Courier Corporation.
- [88] Landau, Y. D. (1984). Adaptive control: The model reference approach. *IEEE Transactions on Systems, Man, and Cybernetics*, (1), 169-170.
- [89] Luo, Q., An, A., & Wang, M. (2019, July). Model Reference Adaptive Control for Microbial Fuel Cell (MFC). In *Proceedings of the 2019 4th International Conference on Robotics, Control and Automation* (pp. 53-59).
- [90] Cespedes, K., Dogan, K. M., Yildirim, E., Yucelen, T., Gruenwald, B. C., & Chakravarthy, A. (2020). Experimental Results of a Model Reference Adaptive Control Approach on an Interconnected Uncertain Dynamical System. In *AIAA Scitech 2020 Forum* (p. 1081).
- [91] Jain, P., & Nigam, M. J. (2013). Design of a model reference adaptive controller using modified MIT rule for a second order system. *Advance in Electronic and Electric Engineering*, 3(4), 477-484.
- [92] Shekhar, A., & Sharma, A. (2018, August). Review of Model Reference Adaptive Control. In *2018 International Conference on Information, Communication, Engineering and Technology (ICICET)* (pp. 1-5). IEEE.
- [93] Kavuran, G., Ates, A., Alagoz, B. B., & Yeroglu, C. (2017). An experimental study on model reference adaptive control with fractional order adjustment rules for DC rotor control. *Journal of Control Engineering and Applied Informatics*, 19(4), 101-111.
- [94] Alagöz, B. B., Kavuran, G., Ateş, A., Yeroğlu, C., & Alisoy, H. Multi-loop Model Reference Adaptive PID Control for Fault-Tolerance. *Balkan Journal of Electrical and Computer Engineering*, 7(3), 276-285.
- [95] Tepljakov, A., Alagoz, B. B., Gonzalez, E., Petlenkov, E., & Yeroglu, C. (2018). Model reference adaptive control scheme for retuning method-based fractional-order PID control with disturbance rejection applied to closed-

loop control of a magnetic levitation system. *Journal of Circuits, Systems and Computers*, 27(11), 1850176.

- [96] **Alagoz, B. B., Tepljakov, A., Kavuran, G., & Alisoy, H.** (2018, September). Adaptive Control of Nonlinear TRMS Model by Using Gradient Descent Optimizers. In *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)* (pp. 1-6). IEEE.
- [97] **Toulis, P., Tran, D., & Airoidi, E.** (2016, May). Towards stability and optimality in stochastic gradient descent. In *Artificial Intelligence and Statistics* (pp. 1290-1298).
- [98] **Panicucci, B., Pappalardo, M., & Passacantando, M.** (2009). A globally convergent descent method for nonsmooth variational inequalities. *Computational Optimization and Applications*, 43(2), 197-211.
- [99] **Baldi, P.** (1995). Gradient descent learning algorithm overview: A general dynamical systems perspective. *IEEE Transactions on neural networks*, 6(1), 182-195.
- [100] **Pohlheim, H.** (2005). Geatbx examples examples of objective functions. *Documentation for GEATbx version 3.7 (Genetic and Evolutionary Algorithm Toolbox for use with Matlab)*, 5-6.
- [101] **Fletcher, R.** (2013). *Practical methods of optimization*. John Wiley & Sons.
- [102] **TAN, R. K., & BORA, Ş.** (2017). MODELLEME VE BENZETİM ORTAMINDA PARAMETRE OPTİMİZASYONU VE KULLANILAN TEKNİKLER. *Mühendislik Bilimleri ve Tasarım Dergisi*, 5(3), 685-697.
- [103] **Parzen, E.** (1979). Nonparametric statistical data modeling. *Journal of the American statistical association*, 74(365), 105-121.
- [104] **Fan, J., & Li, R.** (2004). New estimation and model selection procedures for semiparametric modeling in longitudinal data analysis. *Journal of the American Statistical Association*, 99(467), 710-723.
- [105] **Manogaran, G., & Lopez, D.** (2018). Health data analytics using scalable logistic regression with stochastic gradient descent. *International Journal of Advanced Intelligence Paradigms*, 10(1-2), 118-132.
- [106] **Rupp, M., Tkatchenko, A., Müller, K. R., & Von Lilienfeld, O. A.** (2012). Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, 108(5), 058301.
- [107] **David, R., & Alla, H.** (1994). Petri nets for modeling of dynamic systems: A survey. *Automatica*, 30(2), 175-202.
- [108] **Berger, E. J.** (2002). Friction modeling for dynamic system simulation. *Appl. Mech. Rev.*, 55(6), 535-577.
- [109] **Cai, G., Chen, B. M., Peng, K., Dong, M., & Lee, T. H.** (2006, June). Modeling and control system design for a UAV helicopter. In *2006 14th Mediterranean Conference on Control and Automation* (pp. 1-6). IEEE.
- [110] **Galán, O., Romagnoli, J. A., Palazoğlu, A., & Arkun, Y.** (2003). Gap metric concept and implications for multilinear model-based controller design. *Industrial & engineering chemistry research*, 42(10), 2189-2197.

- [111] **Hanna, A. I., & Mandic, D. P.** (2002). Nonlinear FIR adaptive filters with a gradient adaptive amplitude in the nonlinearity. *IEEE Signal Processing Letters*, 9(8), 253-255.
- [112] **Mandic, D. P., Hanna, A. I., & Razaz, M.** (2001). A normalized gradient descent algorithm for nonlinear adaptive filters using a gradient adaptive step size. *IEEE Signal Processing Letters*, 8(11), 295-297.
- [113] **Ng, S. C., Chung, C. Y., Leung, S. H., & Luk, A.** (1994, April). Fast convergent genetic search for adaptive IIR filtering. In *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing* (Vol. 3, pp. III-105). IEEE.
- [114] **Panda, G., Pradhan, P. M., & Majhi, B.** (2011). IIR system identification using cat swarm optimization. *Expert Systems with Applications*, 38(10), 12671-12683.



8. ÖZGEÇMİŞ

Ad-Soyad : Nagihan YAĞMUR
Doğum Tarihi ve Yeri : 09.03.1995 MALATYA
E-posta : nagihan.yagmur@dpu.edu.tr

ÖĞRENİM DURUMU:

- **Lisans** : 2017, İnönü Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü
- **Yüksek Lisans** : 2020, İnönü Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği

MESLEKİ DENEYİM:

- 2019- ... Dumlupınar Üniversitesi, Bilgisayar Mühendisliği Bölümünde Araştırma Görevlisi.

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN ÇALIŞMALAR (Makaller, Bildiriler, Patentler v.b.)

- **Yağmur, N., & Alagoz, B. B. (2019, April)**. Comparison of Solutions of Numerical Gradient Descent Method and Continous Time Gradient Descent Dynamics and Lyapunov Stability. In *2019 27th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.
- **Yagmur, N., & Alagoz, B. B. (2019, September)**. Adaptive Gradient Descent Control of Stable, First Order, Time-delay Dynamic Systems According to Time-Varying FIR Filter Model Assumption. In *2019 International Artificial Intelligence and Data Processing Symposium (IDAP)* (pp. 1-5). IEEE.