

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

**MEMETİK ALGORİTMA KULLANARAK PID
DENETLEYİCİ TASARIMI**

**Tezi Hazırlayan
Rüştü AKAY**

**Tezi Yöneten
Yrd. Doç. Dr. Adem KALINLI**

**Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi**

**Ocak 2006
KAYSERİ**

Yrd. Doç. Dr. Adem KALINLI danışmanlığında **Rüştü AKAY** tarafından hazırlanan “**Memetik Algoritma Kullanarak PID Denetleyici Tasarımı**” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

31.01.2006

JÜRİ:

Başkan : Prof. Dr. Kenan Danışman

Üye : Prof. Dr. Derviş Karaboğa

Üye : Yrd. Doç. Dr. Adem Kalınlı

ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulunun 17/02/2006 tarih ve 2006/06-06 sayılı kararı ile onaylanmıştır.

17.02.2006.



N. Ayyıldız
Enstitü Müdürü

Prof. Dr. Nusret Ayyıldız

TEŞEKKÜR

Bu tezin hazırlanması esnasında zamanını, kıymetli bilgi ve yardımlarını esirgemeyen saygı değer hocam Yrd. Doç. Dr. Adem KALINLI'ya, optimizasyon konusunu bizlere sevdiren saygı değer hocam Prof. Dr. Derviş KARABOĞA'ya, öğrenmeyi ve araştırmayı öğreten tüm hocalarıma teşekkürlerimi sunarım.

MEMETİK ALGORİTMA KULLANARAK PID DENETLEYİCİ TASARIMI

Rüştü AKAY

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Ocak 2006

Tez Danışmanı: Yrd. Doç. Dr. Adem KALINLI

ÖZET

Zeki problem çözme tekniklerini temel alan, bölgesel ve küresel araştırma algoritmalarının etkin operatörlerinin bir arada kullanılmasıyla oluşturulan memetik algoritmalar çeşitli alanlardaki optimizasyon problemlerinin çözümünde ümit verici yaklaşımlar olarak görülmektedir. Bu çalışmada, öncelikle genetik algoritma ve benzetilmiş tavlama algoritmasının bir arada kullanılmasıyla bir memetik algoritma modeli gerçekleştirilmiş ve gerçekleştirilen algoritmanın değişik zorluk derecelerine sahip sürekli test fonksiyonlarının optimizasyonundaki başarımı incelenmiştir.

Endüstriyel süreçlerin kontrolünde PID denetleyiciler çok yaygın bir şekilde kullanılmaktadır. PID denetleyicilerin tasarımı, arzu edilen denetim kriterlerini sağlayacak şekilde oransal, integral ve türev kazancı olarak tanımlanan üç parametrenin optimum değerlere ayarlanması ile gerçekleştirilmektedir. Nispeten basit olan bazı sistemlere ait denetleyici tasarımları klasik metotlarla başarılı bir şekilde gerçekleştirilebilirken, sistemler karmaşıktıkça klasik metotlarla gerçekleştirilen tasarımların başarımı oldukça azalmaktadır. Sistemlerin anlaşılabilirliğinin artması ve daha karmaşık bir hal alması, denetleyici tasarımında daha etkili metotlara olan ihtiyacın artmasına neden olmuştur. Bu nedenle, son yıllarda araştırmacıların ilgisi zor ve büyük boyutlu optimizasyon problemlerinin çözümü için önerilen sezgisel algoritmalar üzerine kaymaya başlamıştır.

Bu tezde, PID denetleyici tasarımı amacıyla memetik algoritmaya dayalı bir yaklaşım önerilmiştir. Önerilen yaklaşımla değişik özelliklere sahip çeşitli sistemler için PID denetleyici tasarımları gerçekleştirilerek, elde edilen sonuçlar literatürde yer alan diğer bazı metotlara ait sonuçlarla karşılaştırılmıştır. Elde edilen sonuçlardan önerilen yaklaşımın PID denetleyici tasarımı amacıyla başarıyla kullanılabileceği görülmüştür.

Anahtar Kelimeler: Genetik Algoritma, Benzetilmiş Tavlama Algoritması, Memetik Algoritma, PID Denetleyici, Denetleyici Tasarımı.

TUNING PID CONTROLLER PARAMETERS USING MEMETIC ALGORITHM

Rüştü AKAY

Erciyes University, Graduate School of Natural and Applied Sciences

MS.c. Thesis, January 2006

Thesis Supervisor: Assist. Prof. Dr. Adem KALINLI

ABSTRACT

Memetic Algorithm that is based on intelligent problem solving techniques and consists of effective operators of local and global search algorithms is applied to some optimization problems successfully. In this thesis, we examined a kind of Memetic Algorithm which uses Genetic Algorithm and Simulated Annealing Algorithm and tried to improve the performance of the algorithm by adding new units to the algorithm and the algorithm performance was tested on continuous benchmark functions with different difficulty levels.

PID controllers, popular industrial control units, are mostly enough to control most of the systems by adjusting proportional, derivative and integral gains to arrange optimum merits. While it is possible to design a simple control system by conventional methods, as the system gets complicated these methods are not sufficient. By the improvement in understanding control system applications and the increment in the complexity of systems, it is needed to have more effective methods in controller design. Therefore, researchers tend to develop heuristic methods such as the one in this thesis to solve optimization problems.

In this work, some models of PID controller design having different properties were carried out by the proposed method which is based on Memetic Algorithm and the method was compared with the results with some methods in the literature. From the results obtained, it can be said that the proposed methodology can be successfully used to determine optimum PID parameters.

Keywords: Genetic Algorithm Simulated Annealing Algorithm, Memetic Algorithm, PID Controller, Controller Design.

İÇİNDEKİLER

KABUL VE ONAY	i
TEŞEKKÜR.....	ii
ÖZET	iii
ABSTRACT	iv
ŞEKİLLER LİSTESİ	viii
TABLOLAR LİSTESİ	x
KISALTMALAR VE SİMGELER.....	xi
1. BÖLÜM	
GİRİŞ	1
2. BÖLÜM	
OPTİMİZASYON VE TEMEL KAVRAMLAR	4
2.1. Optimizasyon Kavramı	4
2.2. Bölgesel Minimum, Küresel Minimum, Uygulanabilir Bölge.....	5
2.3. Arama Yöntemleri.....	6
2.4. Optimizasyon Problemlerinin Sınıflandırılması	8
2.5. Optimizasyon Metotlarının Sınıflandırılması	9
2.6. Sezgisel Algoritmalar	10
2.7. Karmaşıklık Teorisi.....	13
3. BÖLÜM	
MEMETİK ALGORİTMALAR	15
3.1. Giriş.....	15
3.2. Memetik Algoritma	15
3.3. Genetik Algoritma.....	19
3.3.1. GA'nın Klasik Optimizasyon Tekniklerinden Farklılıkları	20
3.3.2. GA'nın Çalışma Esasları.....	22
3.3.2.1. Kodlama	23
3.3.2.2. Başlangıç Yoğunluğunun Oluşturulması	23
3.3.2.3. Uygunluk Fonksiyonu	24

3.3.2.4. Doğal Seçim	24
3.3.2.5. Çaprazlama	25
3.3.2.6. Mutasyon	26
3.3.2.7. Yer Değiştirme	27
3.3.2.8. Genetik Algoritmada Kullanılan Kontrol Parametreleri	27
3.4. Benzetilmiş Tavlama Algoritması	28
3.4.1. Probleme Özel Seçenekler	30
3.4.2. Algoritmanın Kendisine Ait Genel Seçenekler	31
3.4.2.1. Teorik Soğutma Stratejileri	31
3.4.2.2. Basit Soğutma Stratejileri	32
3.5. Gerçekleştirilen Memetik Algoritma Modeli	34

4. BÖLÜM

MEMETİK ALGORİTMA KULLANARAK SÜREKLİ TEST FONKSİYONLARIN OPTİMİZASYONU	39
4.1. Giriş	39
4.2. Memetik Algoritma ile Sürekli Fonksiyonların Optimizasyonu	39
4.3. Benzetim Sonuçları	41

5. BÖLÜM

KONTROL SİSTEMLERİ	51
5.1. Giriş	51
5.2. Kontrol Sistemleri ve Türleri	51
5.3. Kontrol Sistemlerinin Tasarım İlkeleri	54
5.4. Geçici Durum Parametreleri	56
5.5. Süreçlerin Matematiksel Modellenmesi	57
5.6. Temel Denetim Organları	59
5.6.1. İkili veya Aç-Kapa Denetim Organı	60
5.6.2. Orantı Denetim Etkisi (P-Denetim)	60
5.6.3. Integral Denetim Etkisi ve Orantı Artı İntegral Denetim Organı (PI-Denetim) ...	61
5.6.4. Türev Denetim Etkisi ve Orantı Artı Türev Denetim Organı (PD-Denetim)	62
5.6.5. Orantı Artı Integral Artı Türev Denetim Organı (PID-Denetim)	63
5.7. Sistemlerin Kararlılığı	64

5.7.1. Routh-Hurwitz Kararlılık Ölçütü	65
5.7.2. Nyquist Kararlılık Ölçütü	67
5.8. Performans İndisleri	68
5.9. PID Kontrol Ediciler	70
5.9.1. Tek Girişli ve Tek Çıkışlı Sistem Tasarım Metotları	73
5.9.1.1. Ziegler-Nichols Metodu	74
5.9.2. Çok Girişli ve Çok Çıkışlı Sistem Tasarım Metotları	78
 6. BÖLÜM	
MEMETİK ALGORİTMA KULLANILARAK DENETLEYİCİ TASARIMI	79
6.1. Giriş	79
6.2. Memetik Algoritma ile Denetleyici Tasarımı	81
 7. BÖLÜM	
SONUÇLAR	102
 KAYNAKLAR	104
ÖZGEÇMİŞ	111

ŞEKİLLER LİSTESİ

Şekil 2.1. Bölgesel minimum ve küresel minimum kavramları.....	6
Şekil 2.2.(a) Sürekli araştırma uzayı, (b) Ayrık araştırma uzayı.....	9
Şekil 3.1. Memetik algoritmanın blok şeması.....	17
Şekil 3.2. Rulet seçme tekniği.....	25
Şekil 3.3. Tek noktalı ve iki noktalı çaprazlama işlemi	26
Şekil 3.4. İkili kodlamada mutasyon uygulaması	26
Şekil 3.5. Isıl işlem kademeleri	28
Şekil 3.6. Basamak tipte sıcaklık düşürülmesi.....	32
Şekil 3.7. Sürekli olarak sıcaklığın düşürülmesi.....	33
Şekil 3.8. Monoton olmayan şekilde sıcaklığın düşürülmesi.....	33
Şekil 3.9. Gerçekleştirilen memetik algoritma modeli.....	35
Şekil 3.10. Uyarlanır mutasyon oranı	37
Şekil 4.1. F1 fonksiyonu için (a) TA, Paralel TA, TKKO, Paralel KKO, MA ve (b) GA ile elde edilen sonuçların dağılımı.....	42
Şekil 4.2. F2 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı	43
Şekil 4.3. F3 fonksiyonu için (a) TA, Paralel TA, GA, Paralel KKO ve MA, (b) TKKO ile elde edilen sonuçların dağılımı.....	44
Şekil 4.4. F4 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı	45
Şekil 4.5. F5 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı	46
Şekil 4.6. F6 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı	47
Şekil 4.7. F7 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı	48
Şekil 5.1. Sistem blok şeması.....	53
Şekil 5.2. Açık çevrimli sistem blok şeması	53
Şekil 5.3. Kapalı çevrimli sistem blok şeması	53
Şekil 5.4. Bir kontrol sisteminin birim basamak cevabı	57
Şekil 5.5. Ardışık kontrol sisteminin temel blok diyagramı	58
Şekil 5.6. Geribeslemeli bağlantı	59

Şekil 5.7. Kontrol edilen bir sistemin blok diyagramı	60
Şekil 5.8. T_i ayar değerinin sistem cevabı üzerindeki etkileri.....	62
Şekil 5.9. T_d ayar değerinin sistem cevabı üzerindeki etkileri	63
Şekil 5.10. Kazanç katsayısına bağlı olarak kutupsal eğri değişimi	68
Şekil 5.11. IAE hata fonksiyonu grafiği.....	69
Şekil 5.12. PID Denetleyicili Sistem.....	70
Şekil 5.13. Uyarlanırlı kontrol edicinin basit gösterimi	73
Şekil 5.14. Açık çevrimli basamak tepkisi.....	74
Şekil 5.15. PID denetimli bir sistem	76
Şekil 6.1. Sistem 1 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.....	83
Şekil 6.2. Sistem 2 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.....	86
Şekil 6.3. Farklı denetleyiciler için Sistem 2'nin basamak cevapları	88
Şekil 6.4. Sistem 3 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.....	90
Şekil 6.5. Farklı denetleyiciler için Sistem 3'ün basamak cevapları	91
Şekil 6.6. Sistem 4 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.....	93
Şekil 6.7. Farklı denetleyiciler için Sistem 4'ün basamak cevapları	94
Şekil 6.8. Sistem 5 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.....	96
Şekil 6.9. Farklı denetleyiciler için Sistem 5'in basamak cevapları	98
Şekil 6.10. Sistem 6 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.	100
Şekil 6.11. Farklı denetleyiciler için Sistem 6'nın basamak cevapları	101

TABLOLAR LİSTESİ

Tablo 3.1. Kontrol parametreleri için önerilen değerler	28
Tablo 4.1. Benzetimde kullanılan nümerik test fonksiyonları	40
Tablo 4.2. Test fonksiyonlarının parametre sayısı, optimum çözüm değerleri, parametre araştırma uzayı ve çözümü temsil etmekte kullanılan toplam bit sayıları	41
Tablo 5.1. Orantı, integral ve türev kazançlarının etkileri	73
Tablo 5.2. Sürekli döngü için Z-N ayar kuralları	74
Tablo 5.3. Süreç eğrisi metodu için Z-N ayar kuralları	75
Tablo 6.1. Sistem 1 için Z-N ve MA ile belirlenen tasarım sonuçları	82
Tablo 6.2. Z-N, GA, MA için geçici durum parametreleri	84
Tablo 6.3. Sistem 2 için Z-N ve MA ile belirlenen tasarım sonuçları	87
Tablo 6.4. Sistem 2 için tasarım sonuçlarının karşılaştırılması	87
Tablo 6.5. Sistem 3 için Z-N ve MA ile belirlenen tasarım sonuçları	89
Tablo 6.6. Sistem 3 için tasarım sonuçlarının karşılaştırılması	91
Tablo 6.7. Sistem 4 için Z-N ve MA ile belirlenen tasarım sonuçları	92
Tablo 6.8. Sistem 4 için tasarım sonuçlarının karşılaştırılması	94
Tablo 6.9. Sistem 5 için Z-N ve MA ile belirlenen tasarım sonuçları	95
Tablo 6.10. Sistem 5 için tasarım sonuçlarının karşılaştırılması	97
Tablo 6.11. Sistem 6 için Z-N ve MA ile belirlenen tasarım sonuçları	99
Tablo 6.12. Sistem 6 için tasarım sonuçlarının karşılaştırılması	99

KISALTMALAR VE SİMGELER

EA	: Evrimsel Algoritma (Evolution Algorithms)
GA	: Genetik Algoritma (Genetic Algorithms)
BT	: Benzetilmiş Tavlama Algoritması (Simulated Annealing)
TA	: Tabu Araştırma Algoritması (Tabu Search)
KKO	: Karınca Koloni Optimizasyonu (Ant Colony Optimization)
YSA	: Yapay Sinir Ağları (Artificial Neural Network)
MA	: Memetik Algoritma (Memetic Algorithm)
TT	: Tepe Tırmanma Algoritması
t_g	: Gecikme zamanı
t_y	: Yükselme zamanı
t_t	: Tepe zamanı
t_0	: Oturma zamanı
M_p	: Maksimum aşma
K_p	: Oransal kazanç
K_i	: Integral kazanç
K_d	: Türev kazanç
T_i	: Integral zaman sabiti
T_d	: Türev zaman sabiti
PID	: Oransal-Integral-Türev (Proportional-Integral-Derivative)
SISO	: Tek Girişli Tek Çıkışlı (Single Input Single Output)
MIMO	: Çok Girişli Çok Çıkışlı (Multi Input Multi Output)
Z-N	: Ziegler-Nichols

1. BÖLÜM

GİRİŞ

Yapay zeka, bilgisayar bilimi, operasyonel araştırma ve ilgili alanlardaki temel konulardan birisi optimizasyondur. Optimizasyon kelime anlamı olarak “iyileştirme”, “daha iyi yapma” olarak tanımlanabilir. Optimize edilmesi gereken ise verilen sınırlamaları sağlayacak şekilde parametre değerlerinin bulunması hedeflenen bir optimizasyon problemidir. Problemin parametre değerleri bulunurken mümkün olan en iyi çözümü, makul bir süre içerisinde bulmak hedeflenmektedir.

Optimizasyon problemleri basit ve zor problemler olmak üzere iki ana grupta toplanabilir. Mümkün olan farklı aday çözümlerin kaliteleri hakkında açıkça fikir yürütülebilen problemler basit problemler olarak bilinir. Kabul edilebilir bir zaman içerisinde en iyi çözümün bulunmasının garanti edilmediği problemler ise zor problem olarak tanımlanır [1]. Bununla beraber bilim ve endüstri ile ilgili birçok problem temel olarak zordur.

Algoritmalar, matematiksel terimlerle veya belirli kurallara göre tanımlanarak problemleri çözmek için kullanılan yöntemlerdir. Basit problemlerin çözümünde birçok klasik algoritma başarılı sonuçlar vermektedir. Klasik algoritmaların çoğunlukla probleme özel olması, genel olmaması ve problemin matematiksel fonksiyonlarla tanımlanmasını gerektirmesi gibi dezavantajlarından dolayı problemler zorlaştıkça bu algoritmalarla elde edilen sonuçlarda kabul edilebilir seviyelerden uzaklaşmaktadır. Bu nedenle üretim, kontrol, haberleşme ve taşıma gibi pratik alanlardaki optimizasyon problemlerinin büyük zorluğu, araştırmacıları genel amaçlı esnek ve performansı yüksek yeni algoritmalar geliştirmeye teşvik etmiştir. Son zamanlarda böyle algoritmaları türetmek için araştırmacıların ilgisi yapay zekâ ve tabii bilimler üzerine yoğunlaşmıştır. Geliştirilen bu algoritmalar zor ve büyük boyutlu problemlerin kesin çözümlerini bulmayı garanti edemezler, ancak optimuma yakın bir çözüm bulma

özeliğine sahiptirler. Bu algoritmalar yaklaşık algoritmalar (approximate) veya sezgisel (heuristic) algoritmalar olarak adlandırılmaktadır. Bu algoritmaların popüler olanlarından bazıları şunlardır: genetik algoritmalar (Genetic Algorithms, GA), tabu araştırma algoritması (Tabu Search, TA), benzetilmiş tavlama algoritması (Simulated Annealing, BT), karınca kolonisi optimizasyonu (Ant Colony Optimization, KKO), memetik algoritmalar (Memetic Algorithms, MA) ve yapay sinir ağları (Artificial Neural Network, YSA).

Kontrol sistemleri belirli bir amaca yönelik olarak kendisini veya diğer bir sistemi kumanda etmek, yönlendirmek veya ayarlamak üzere birleştirilen fiziksel organlar kümesidir [2]. Kontrol sistemleri uygulamaları genelde elektrik, mekanik ve kimyasal bölümlerden oluşmaktadır. Fakat politik, sosyal ve iş hayatının dinamiklerinin anlaşılabilirliğinin artması ile birlikte kontrol mühendisliğinin sınırları oldukça genişlemiş bulunmaktadır. Bu kadar geniş kullanım alanlarına sahip olan kontrol sistemleri tasarımında istenen, ihtiyacımız olan sistemin temel parametrelerinin özelliklerini, tanımlamalarını ve kuruluşunu gerçekleştirmektir. Tasarımı gerçekleştirmek için, sistem hedeflerini çok iyi bir şekilde tanımlamak ve anlamak gerekir. Gerçekçi bir tasarım yapılabilmesi için de zaman kısıtlamasına, sistemin optimizasyonuna ve sistemin parametrik olmasına önem verilmelidir. Bazı basit sistem tasarımlarını klasik metotlarla yapmak mümkün iken, sistemler karmaşıktıkça klasik metotlar yeterli olmamaktadır. Dolayısıyla optimizasyonda zor problem olarak nitelendirilen karmaşık kontrol sistemleri tasarımında optimizasyon metotları sıkça kullanılmaktadır.

Kontrol sistemleri denildiğinde ilk akla gelen, endüstriyel süreçlerin %80'inden fazlasında kullanılmakta olan üç terimli PID (Proportional-Integral-Derivative) kontrol edicidir. PID denetleyici yapısının basit olması ve çalışma şartlarının geniş bir aralığında oldukça kararlı olması nedeniyle yaygın olarak tercih edilir. PID denetleyici üç temel denetim etkisinin (**P**roportional, **I**ntegral, **D**erivative) üstünlüklerini tek bir birim içinde birleştiren bir denetim etkisidir. Geçmişte PID kontrol ediciler frekans analiz metotları yardımı ile ayarlanır iken pratik uygulamalardaki kontrol sistemlerinin karmaşıklığının artması bu metotların yerine alternatif metotlar geliştirmeyi gerektirmiştir. Dolayısıyla frekans analiz metotlarına alternatif olarak daha kısa sürede daha iyi tasarım yapılmasına olanak sağlayan PID denetleyici algoritmaları hızlı bir şekilde gelişmiştir.

Yukarda bahsedilen optimizasyon algoritmalarından KKO, GA ve BT algoritmalarının PID denetleyici parametrelerinin bulunması amacıyla kullanılmasına ilişkin literatürde çeşitli sayıda çalışma bulunmaktadır. Bununla beraber, bu ve benzeri algoritmaların her birinin dezavantajlarını ortadan kaldırmak ve başarımlarını artırmak amacı ile paralel çalışan algoritmalar geliştirilmeye başlanmıştır. Geliştirilen bu paralel algoritmalarla PID denetleyici parametrelerinin bulunmasına ilişkin pek fazla çalışma bulunmamaktadır. Bu tez çalışmasının amaçlarından biri GA ve BT algoritmasının bir arada kullanılmasıyla oluşturulan memetik algoritma modelinin sürekli optimizasyon problemlerindeki başarısını incelemektir. Diğer bir amaç ise, ortaya konulan memetik algoritma modeli ile PID denetleyicili sistemlerin tasarımını gerçekleştirmektir.

Bu amaçlar doğrultusunda tez şu şekilde bölümlendirilmiştir:

Bölüm-2’de optimizasyon ve ilgili bazı temel kavramlar açıklanmıştır.

Bölüm-3’te memetik algoritma ve literatürde bulunan ayrık ve sürekli problemler için geliştirilmiş türleri hakkında bilgi verilerek gerçekleştirilen memetik algoritma modeli tanıtılmıştır.

Bölüm-4’te memetik algoritmanın sürekli test fonksiyonları üzerindeki performansı test edilmiştir.

Bölüm-5’de kontrol sistemi tasarımı ve PID denetleyiciler tanıtılmış, sistem tasarımında kullanılan klasik metotlardan bazıları tanıtılmıştır.

Bölüm-6’da gerçekleştirilen memetik algoritma modeli ile farklı özelliklere sahip PID denetleyici tasarımları gerçekleştirilmiş ve bulunan sonuçlar literatürdeki sonuçlarla kıyaslanmıştır.

Bölüm-7’de tez çalışmasının kısa bir değerlendirilmesi yapılarak elde edilen sonuçlar yorumlanmıştır.

2. BÖLÜM

OPTİMİZASYON VE TEMEL KAVRAMLAR

2.1. Optimizasyon Kavramı

Genel olarak *optimum* en iyi, en uygun anlamına gelir. Optimizasyon, belirlenmiş sınırlamalar dâhilinde bir problemin mümkün olabilecek çözümleri arasından en iyi olanı seçme işlemi olarak tanımlanabilir. En iyi çözümü aranan probleme de optimizasyon problemi denir [3].

Mühendislik uygulamalarında birçok problem optimizasyon problemi olarak modellenebilir. Dijital sinyal işleme, devre tasarımı, veritabanı tasarımı, mekanik tasarım, kimyasal süreç kontrol, tesis organizasyonu, fabrikaların üretim ve dağıtım planlaması, ağ tasarımı, trafik planlaması ve uçak tasarımı örnek olarak verilebilecek uygulamalardan bir kısmıdır.

Her optimizasyon problemi amaç fonksiyonu, sınırlamalar ve değişkenler olmak üzere üç temel unsurdan oluşur.

Amaç fonksiyonu; optimum değeri aranan bir fonksiyondur. Problemin türüne göre en iyi kavramı değişir, amaç fonksiyonun bir maliyet veya hata olduğu bir optimizasyon probleminde optimum değer maliyeti veya hatayı en aza indiren değer iken, amaç fonksiyonun karlılık veya verimlilik olduğu bir başka problemde optimum değer karlılığı veya verimliliği en yükseğe taşıyan değerdir. Bazı optimizasyon problemlerinde birden fazla amaç fonksiyonu bulunabilir. Bu tür problemler *çok amaçlı optimizasyon problemi* olarak adlandırılır. Eğer bir optimizasyon probleminde çok sayıda amaç fonksiyonu varsa çoğunlukla çözüm de zordur. Çünkü bir fonksiyonu optimum değere getiren değişken değerleri diğer bir amaç fonksiyonunda verimsiz kalabilir. Bu nedenle problemin çözümünü kolaylaştırmak için çok sayıda amaç

fonksiyonu içeren problemler tek bir amaç fonksiyonuna indirgenmeye çalışılır. Bazı optimizasyon problemlerinde ise amaç fonksiyonu bulunmaz, örneğin bir entegre devrenin yerleşim planını belirleme probleminde amaç, sınırlamaları sağlayan değişkenler kümesi bulmaktır. Burada optimum değeri aranan bir fonksiyon yoktur. Bu tür problemler *uygunluk problemi* olarak adlandırılır. Değeri minimize edilecek bir optimizasyon problemi aşağıdaki şekilde ifade edilebilir.

$$\text{Min. } f(x), x = (x_1, x_2, \dots, x_n)$$

Burada $f(x)$ minimize edilecek amaç fonksiyonu ve $x_1, x_2, \dots, x_n$ de optimum çözüm vektörüdür.

Sınırlamalar; Optimizasyon probleminde kullanılan değişkenlerin belirli değerleri alabilmesini veya belirli değerleri almamasını sağlamak amacıyla tanımlanır. Sınırlamalar probleme göre değişiklik gösterir ve probleme özeldir. Sınırlamalar aşağıda gösterildiği gibi eşitlik veya eşitsizlik biçiminde olabileceği gibi her optimizasyon problemi için kesinlikle gerekli olan bir unsur değildir. Problemin özelliğine bağlı olarak bazı optimizasyon problemlerinde hiçbir sınırlama olmayabilir. Aşağıda sınırlamalarla ilgili bazı ifade şekilleri gösterilmiştir.

$$\begin{aligned} h_i(x) &= 0, & i &= 1, 2, 3, \dots, n \\ g_j(x) &\leq 0, & j &= 1, 2, 3, \dots, l \\ x &\in R^n \end{aligned}$$

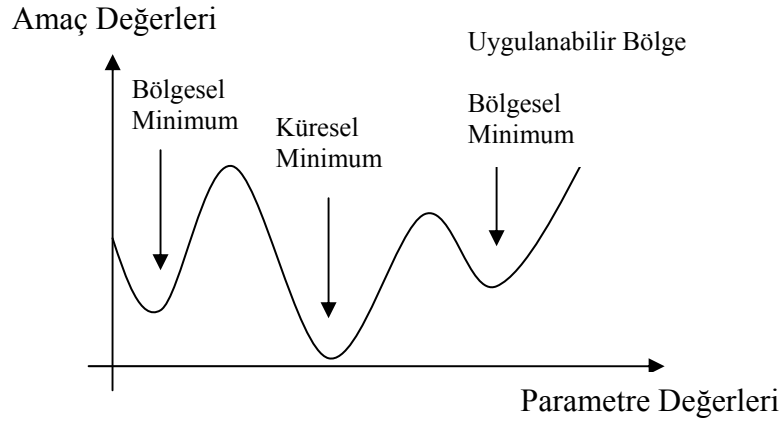
Değişkenler; Optimizasyon problemlerinde amaç fonksiyonunun değerini etkileyen bilinmeyenlerdir. Değişkenleri tam olarak tanımlamadan amaç fonksiyonu ve sınırlamaları tanımlamak mümkün değildir. Bütün optimizasyon problemlerinin ortak noktası optimum değerleri bulunması gereken değişkenlerdir. Optimizasyon problemlerinde probleme göre değişiklik gösteren değişkenlerin kusursuz tanımlanmış olması gerekir.

2.2. Bölgesel Minimum, Küresel Minimum, Uygulanabilir Bölge

Bir optimizasyon probleminde tüm model kısıtlarını sağlayan seçenekler topluluğu *uygulanabilir bölge* (fizibil bölge) olarak adlandırılır.

Değişken değerleri uygun bir bölge içerisinde amaç fonksiyonunun muhtemel çözümleri içinde minimizasyon problemi için en küçük, maksimizasyon problemi için en büyük değeri veren nokta *bölgesel optimum*, değişken değerlerinin uygulanabilir bölgede verdiği en iyi çözüm değeri ise *küresel optimum* (optimum çözüm) olarak tanımlanır. Bu kavramlar Şekil 2.1’de gösterilmiştir. Bununla beraber optimizasyon probleminin uygulanabilir bölge içinde bir bölgesel optimum noktaya takılmasına ise *erken yakınsama* denir.

Optimum noktada sağlanması gereken şartlar, gerek şartlar olarak adlandırılır. Gerek şartları sağlayamayan noktalar optimum çözüm olamaz. Ancak, tek başına gerek şartların sağlanması da o noktayı optimum çözüm yapmaz. Gerek şartlarla birlikte optimum ve optimum olmayan noktaları ayırt etmek için yeter şartlar kullanılır. Gerek şartlar sağlandıktan sonra yeter şartlarda sağlanıyorsa o noktanın optimum çözüm noktası olduğu söylenebilir.



Şekil 2.1. Bölgesel minimum ve küresel minimum kavramları.

2.3. Arama Yöntemleri

Optimizasyon metotları problem ve problem üstü olmak üzere iki düzeyde tanımlanır. Problem düzeyinde optimizasyon işleminde, bir problem kümesi içerisinde her problem tek tek diğerlerinden bağımsız olarak çözülür. Problem üstü düzeyde çeşitli problemlerden oluşan bir küme bir bütün olarak göz önüne alınarak optimum çözüm aranır. Problem üstü düzeyde optimizasyon metotları problem düzeyinde optimizasyon metotlarını yönlendirmek amacıyla da kullanılır [4].

Optimizasyon algoritmalarında çeşitli arama stratejileri kullanılır. Bunlardan en yaygın olarak kullanılanlara yorucu (exhaustive), rasgele (random), aç gözlü (greedy), tepe tırmanma (hill-climbing), sezgisel (heuristic), belirleyici (deterministic) ve ihtimalci (stochastic) arama stratejileri örnek verilebilir [5].

Yorucu aramada giriş veri kümesinin tüm kombinasyonları test edilerek araştırma uzayında en iyi çözüm aranır. Tüm kombinasyonların testini gerektirdiğinden dolayı pratikte uygulaması zor bir metottur.

Rasgele aramada giriş veri kümesinin rasgele üretilen kombinasyonları makul bir çözüm bulununcaya kadar denenir. Küresel çözüme yakınsama özelliği yoktur. Rasgele arama metodunda genellikle kısa zamanda iyi çözümler bulunmaz.

Açgözlü arama rasgele aramaya benzerlik gösterir. Açgözlü arama giriş veri kümesinin rasgele üretilen kombinasyonlarını dener ve iyi bir çözüm bulduğunda bunu saklar. Daha sonra bu iyi çözüm etrafında araştırmaya devam eder. Bu metotta açgözlü kelimesi bölgesel ölçekte daha iyi çözümü kabul etmek yönünde tercihin yüksek oluşunu ifade eder. Bölgesel ölçekte aramayı kötüye götüren çözümler kabul edilemez. Oysa küresel optimum kimi zaman yeni araştırma yönü olarak bölgesel ölçekte kötü olan çözümleri kabul etmekle bulunabilmektedir. Açgözlü arama hızlı sonuç vermesine karşın küresel optimumu bulma bakımından genelde yetersizdir.

Tepe tırmanma metodu bölgesel optimumu aşabilmek için araştırma yönü olarak bölgesel optimum etrafında daha kötü çözümlerin kabul edilmesi gerektiğini temel alır. Bu metotta arama, hata veya maliyet değerini daha yükseğe çıkaran tarafa yönlendirildiği için arama işlemi tepe tırmanmayı çağrıştırır. Arama esnasında elde edilen çözümler iyiye giderken bir noktadan sonra kötüye giderse, o noktanın bir sırt noktası olduğu veya tersi yönde gelişim durumunda gelişimin yön değiştirdiği noktanın bir dip noktası olduğu anlaşılır. Bu metot türev alma işleminin verdiği neticeyi bir başka biçimde arayıp deneyerek bulur ve genellikle doğrusal olmayan problemlerde kullanılır.

Sezgisel metot, akıllı tahminlere dayalı buluşsal veya yeni çözümlerin keşfine götüren bulgulara dayalı arama yöntemidir. Sezgisel terimi ile tanımlanan algoritmalarda araştırma uzayı çok büyük olan problemlerde çözümün aranmasını sınırlayan bir kural, çözüm stratejisi gibi etmenler tanımlanır. Belirlenen hedefe ulaşmak için verilen

sınırlamalar içerisinde geliştirilen çözüm stratejisi ile bulunan çeşitli alternatif çözümlerden en etkili olanlar seçilir. Her karar sürecinde verilecek kararlar değişebilir. Sezgisel stratejiler arama zamanını kısaltır, optimum çözüme yakın çözümler sunar fakat optimum çözümü garanti edemezler.

Belirleyici yaklaşım probleme özeldir ve statik bir yapı kullanır. Bu stratejide mevcut çözüme göre bir sonraki aşamada elde edilecek çözümler belirlidir ve sabittir.

İhtimalci yaklaşım dinamik bir yapıya sahiptir. Bu optimizasyonda eski çözümlerin kalitesi nispetinde değerler alan bir olasılık dağılım fonksiyonuyla bir rasgele sayı üreticiden faydalanarak yeni çözümler belirlenir. Karar mekanizması olasılık tabanlı seçimlere dayanır. Her karar sürecinde verilecek kararlar değişebilir.

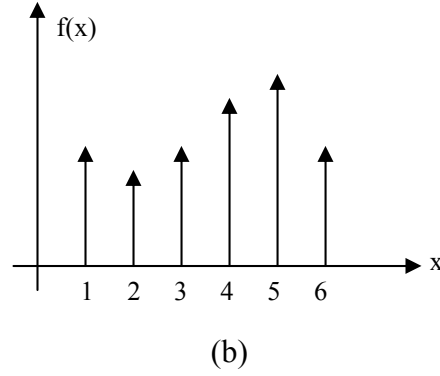
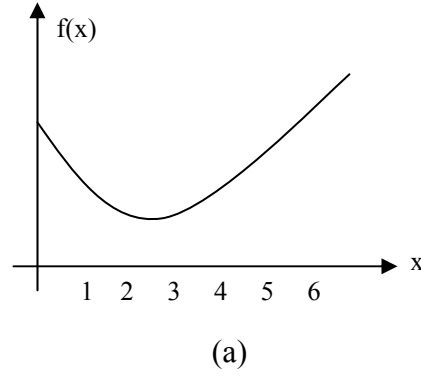
Optimizasyon algoritmaları genel olarak önce belirleyici bir yöntemle hesaplanabilen en iyi çözümü başlangıç çözümü olarak tespit eder ve daha sonra bu çözümü ihtimalci ve sezgisel stratejiler kullanarak geliştirmeye çalışırlar.

2.4. Optimizasyon Problemlerinin Sınıflandırılması

Optimizasyon problemleri çeşitli şekillerde sınıflandırılırlar. Bir optimizasyon probleminde amaç fonksiyonunun değişkenlerinde herhangi bir sınırlama varsa *sınırlamalı optimizasyon*, herhangi bir sınırlama yoksa *sınırlamasız optimizasyon* problemi olarak isimlendirilir.

Optimizasyon problemlerinde diğer bir sınıflandırma ise amaç fonksiyonun ve sınırlamalarla ilgili fonksiyonların doğrusal olup olmamalarına göre yapılır. Optimizasyon problemi doğrusal amaç ve sınırlama fonksiyonlarına sahip ise *doğrusal optimizasyon problemi*, bu fonksiyonlardan herhangi biri doğrusal değil ise *doğrusal olmayan optimizasyon problemi* olarak isimlendirilir. Doğrusal ve doğrusal olmayan fonksiyonlara örnek vermek gerekirse; grafiği doğrulardan oluşan $y = Ax + B$ şeklinde birinci dereceden eşitliği ile tanımlanan fonksiyon doğrusal bir fonksiyon iken, grafiği eğrilerden oluşan $y = e(x)$, $\log(x)$, x^n şeklinde tanımlanan fonksiyonlar doğrusal olmayan fonksiyonlardır [6].

Diğer bir gruplandırma ise ayrık (combinatorial) ve sürekli (continuous) optimizasyon problemleri şeklindedir. Optimizasyon problemindeki bilinmeyenler uygulanabilir bölgede tüm değerleri alabiliyorlarsa *sürekli optimizasyon problemi*, eğer değişkenler sonlu sayıda belirli değerler alabiliyorlarsa *ayrık optimizasyon problemi* olarak adlandırılırlar. Şekil 2.2’de bu problemlere örnek olabilecek araştırma uzayları gösterilmiştir.



Şekil 2.2.(a) Sürekli araştırma uzayı, (b) Ayrık araştırma uzayı.

2.5. Optimizasyon Metotlarının Sınıflandırılması

Optimizasyon problemlerini çözmek için geliştirilen metotlar genel olarak iki gruba ayrılırlar; direkt metotlar ve optimalite kriterine dayalı metotlar (indirekt metotlar).

Direkt metotlarda araştırmaya tahmini bir başlangıç çözümü ile başlanır. Bu başlangıç çözümü genellikle optimalite şartlarını sağlayamayacağından bu şartlar sağlanana kadar başlangıç çözümü algoritma tarafından iteratif olarak geliştirilir. Bu tür bir yaklaşımla optimum çözümleri bulmak için çözüm uzayı araştırılmış olur.

İndirekt metotlar ise optimalite şartlarına göre çözüm arayan minimizasyon teknikleridir. Optimalite kriteri bir fonksiyonun minimum noktalarının sağlaması gereken şartlardır. İlk önce gerekli ve yeterli şartlar yazılır, sonra bu şartlar bölgesel minimuma aday noktalar için çözülür.

2.6. Sezgisel Algoritmalar

Herhangi bir amacı gerçekleştirmek veya hedefe varmak için çeşitli alternatif hareketlerden etkili olanlara karar vermek amacı ile tanımlanan kriterler veya bilgisayar metotlarıdır. Bu tür algoritmalar çözüm uzayında optimum çözüme yakınsaması ispat edilemeyen algoritmalar olarak da adlandırılır. Sezgisel algoritmalar yakınsama özelliğine sahiptir, ama kesin çözümü garanti edemezler ve sadece kesin çözüme yakın bir çözümü garanti edebilirler [3].

Sezgisel yaklaşımlı modern çalışmalar 1948’de Polya ile başlamıştır [7]. O yıllarda yönetim bilimi, hareket araştırma ve birçok pratik problemin çözümünde mükemmel olmayan fakat etkili sezgisel yöntemlerin uygulanmasıyla başarılı çözümler elde edilmiştir [8]. 1960’larda, optimizasyon teknikleri üzerinde çalışmalar yoğunlaşmıştır. Çalışmaların artması ile algoritmaların gelişimi de başlamıştır.

Birçok sezgisel algoritma çoğunlukla belirli türdeki problemlerin çözümünde başarılı sonuçlar üretmektedir. Yani bir problem için iyi sonuç verirken başka bir problem için başarılı sonuçlar vermeyebilir. Bundan dolayı daha genel ve başarılı algoritmaların geliştirilmesi için sosyal, biyoloji, fizik, bilgisayar gibi bilimler temel alınarak yapılan çalışmaların sayısı sürekli artmaktadır. Bu tür yaklaşımlara modern sezgisel yaklaşımlar adı verilmektedir. Bu algoritmaların popüler olanlarından bazıları şunlardır: biyoloji biliminin prensiplerini temel alan evrimsel algoritmalar (Evolution Algorithms, EA), zeki problem çözmenin genel kurallarından esinlenerek geliştirilen tabu araştırma algoritması (Tabu Search, TA), fizik biliminden esinlenerek geliştirilen benzetilmiş tavlama algoritması (Simulated Annealing, BT), karınca kolonilerinin davranışlarını örnek alan karınca kolonisi optimizasyonu (Ant Colony Optimization, KKO) ve insan beyninin çalışma prensibinden esinlenilerek geliştirilen yapay sinir ağları (Artificial Neural Network, YSA).

Evrimsel algoritmalar doğadaki evrimsel süreçleri model olarak kullanan bilgisayara dayalı problem çözme teknikleridir. Evrimsel algoritmanın temeli, en iyi uyum sağlayabilenin yaşayabilmesine dayalı biyolojik evrime dayanır. Evrimsel algoritmaların bir alt dalı olan genetik algoritmalarda, doğal seçim ve genetik evrim kurallarına dayanarak geliştirilen bir algoritma modelidir. GA ilk olarak Holland tarafından önerilmiştir [9]. Doğal evrimdeki temel amaç, canlıların yaşadıkları çevrede hayatta kalabilmeleri için gerekli gelişimi sağlamasıdır. GA'nın her çevrimi bir nesil olarak adlandırılır. Başlangıç popülasyonu her nesilde doğal seçme, mutasyon ve çaprazlama gibi genetik operatörlerle geliştirilmeye çalışılır. GA'da genetik işlevler, her bir birey için elde edilen uygunluk değerlerinin değerlendirilmesi ile başlar. Değerlendirme işlevi ile bireylerin uygunluk değerleri baz alınarak bir sonraki nesile aktarılma oranı belirlenir. Ölçeklendirilmiş uygunluk değerleri baz alınarak, bireylerin bir sonraki nesle aktarılma oranı belirlenmiş ve kötü bireyler atılarak yerlerine iyi bireyler yerleştirilmiştir. Böylece çaprazlamaya girecek bireylerin iyi bireylerden oluşması sağlanmış ve iyi bireylerin sonraki nesile katkıları arttırılmıştır. En son nesil, geçmiş nesillere göre en iyi uyumu sağlayan nesildir [10]. Temel bir GA, paralel yapısı nedeniyle araştırma uzayının etkin bölgelerini oldukça çabuk bulabilir. Ancak olasılık tabanlı yaklaşımlar gösterdiği için birçok durumda bölgesel yakınsama problemine sahip olmakta veya küresel optimuma yakınsaması kabul edilebilir zamanın ötesinde bir süre gerektirebilmektedir. GA araştırmanın geçmiş adımlarına ait bilgileri kümülatif olarak biriktirmekle beraber, değerlendirilmiş çözümlerin bilgisini tutan bir mekanizma kullanmadığı için, benzer yada aynı çözümleri de birçok defa tekrar değerlendirebilmektedir. Bu nedenle GA bazı problemler için aşırı fazla değerlendirme sayısına ihtiyaç duyabilmektedir.

TA algoritması zeki problem çözme prensiplerine dayalı olarak Glover tarafından 1986'da önerilmiştir [11,12]. TA algoritması temel olarak belirli bir problem üzerine elde edilen ilk çözüm etrafındaki komşuluklar oluşturmaktır. Komşuluk mekanizması ele alınarak birbirini izleyen seçilmiş çözümlerden daha iyi olanlar hafızaya alınır (Tabu listesi). TA algoritması sahip olduğu bu hafıza ile araştırmanın geçmiş adımları hakkında kayıt tutmakta ve bu kayıtları araştırma uzayında yeni çözümler oluşturulması ve keşfedilmesi için kullanmaktadır. Oluşturulan bu tabu listesi kötü sonuç veren bölgelerde daha fazla işlem yapılmamasını sağlamaktadır. Böylece algoritmanın

verimliliği de artmış olmaktadır. Tabu listesinin en önemli özelliklerinden biriside mevcut tabu listesinin aday komşu çözümler ile karşılaştırıldıktan sonra bir sıralama ve karşılaştırma işlemi yaparak kendisini yenileyebilmesidir. TA algoritması iteratif bir algoritmadır ve tek bir çözümden hareketle küresel optimum değere ulaşmaya çalışır. Yasaklama ve serbest bırakma stratejileri ile de algoritmanın bölgesel optimumdan kurtulması sağlanmaya çalışılır. Birçok problem için başarılı sonuçlar üretmesine karşın, TA algoritmasının seri yapısı küresel optimumun bulunduğu bölgeye erişebilmesi için gereken zamanın uzamasına neden olabilmektedir.

BT algoritması ilk olarak Kirkpatrick, Gelatt ve Vecchi tarafından önerilmiş, olasılık tabanlı sezgisel bir algoritmadır [13]. BT algoritması, Metropolis algoritmasından esinlenerek, katıların fiziksel tavlama işlemi ile kombinasyonel optimizasyon problemlerin çözümü arasındaki benzerliğe dayalı olarak ortaya konmuştur. Fiziksel tavlama, bir katının düşük enerjili durumlarının elde edilmesi sürecidir. Eritilen katının ısısının çok yavaş düşürülmesi ile katının düşük enerjili durumuna ulaşması sağlanır. BT algoritması iteratif yapısı nedeniyle bölgesel araştırmada başarılıdır. BT algoritması yerel arama yöntemlerinin yerel bir minimuma ulaştıktan sonra küresel minimum için daha fazla arama yapmamasından kaynaklanan eksikliğini gidermeye çalışan bir yöntemdir. Bununla birlikte, BT algoritması araştırmanın geçmiş adımlarıyla ilgili bir kayıt tutmadığından ve herhangi bir gelişme olmadığında gelecek çözümü olasılık tabanlı kurallar ile belirlediği için benzer çözümleri defalarca değerlendirebilmektedir. Ayrıca algoritmanın yakınsama hızı da başlangıç çözümünün bulunduğu noktaya fazlaca bağlı olmaktadır. Eğer başlangıç çözümü küresel optimumun bulunduğu bölgeye çok uzak olursa, o bölgeye ulaşması çok uzun zaman gerektirebilmektedir.

KKO, Dorigo ve arkadaşları tarafından önerilen sezgisel algoritmalarından birisidir [14,15]. Algoritma gerçek karıncaların yön ve yiyecek bulma stratejilerine dayalıdır. Karıncalar, yiyecek ile yuvaları arasındaki en kısa yolu bulma kabiliyetine sahiptirler. Karıncalar yürürken yolları üzerine kimyasal bir madde (feromon) bırakmakta ve birbirleriyle haberleşmede bu maddenin yaydığı kokuyu kullanmaktadırlar. Tüm karıncaların hızlarının ve yollara bıraktıkları koku miktarının eşit olduğu kabul edildiğinde, daha kısa yollarda birim zamanda daha fazla koku maddesi oluşacaktır. Dolayısıyla karıncaların büyük çoğunluğu bu en kısa yolu seçecektir. KKO algoritması yukarda tanımlanan gerçek karınca kolonilerinin yapmış olduğu doğal optimizasyon

işleminin yapay bir modelidir. KKO algoritması birçok problemde etkili olmasına karşın giriş veri kümesi büyük olan problemlerde erken yakınsamayla karşı karşıyadır. Bu erken yakınsama problemini aşmak amacıyla KKO algoritması için çeşitli modeller önerilmiştir [16,17,18].

Yapay Sinir Ağları, insan beyninin çalışma prensibini kendine model edinmiş yapay sistemlerdir. Bir YSA birbirleri ile paralel ve birbirine bağlı nöronların hiyerarşik organizasyonundan oluşur. YSA'da bilgi geleneksel metotların aksine nöronlar arasındaki bağlantılarda gizlidir. Bu bağlantıların geniş bir alana yayılmış olmaları YSA'ya yayılmış hafıza özelliği verir. Yayılmış hafıza özelliği YSA'ya hata toleransı, eksik bilgi ile çalışabilmek gibi özellikler kazandırır. Bir YSA tabakalar halinde dizilmiş nöronlardan oluşur. Ağ tipine bağlı olarak bir tabakadaki nöron, diğer tabakalardaki nöronlarla veya aynı tabakadaki başka nöronlarla bağlantılıdır. YSA'lar geleneksel metotlara oranla bir problemi, problemin kendi özelliğine ait bir takım matematiksel formüller kullanarak çözmezler, matematiksel formüller yerine problemi örnekler üzerinden öğrenebilirler ve değişen şartlara adapte olabilirler. Bu gibi avantajlarından dolayı özellikle mühendislik de çok geniş bir uygulama alanına sahiptirler [19-20].

2.7. Karmaşıklık Teorisi

Algoritmaların ne kadar karmaşık olduğunu, ne kadar hızlı işlediğini, işlem süresinin ne kadar uzun olduğunu ve problemlerin ne kadar zor olduğunu araştıran matematik dalı karmaşıklık teorisi ile açıklanır [21].

Karmaşıklık teorisi her problemin ayrı ayrı algoritmik çözümlerinden çok bir problem sınıfına ait tüm problemleri çözebilecek genel algoritmalar ile ilgilenir. Bu algoritmaların gerektirdiği işlem zamanını ve karmaşıklık derecesini ölçer. İşlem zamanı aynı zamanda algoritmaların yeterliliğinin bir ölçüsüdür. Çünkü bilgisayar teknolojisindeki hızlı gelişmeler birim işlem zamanını ve kullanılan hafıza miktarı gibi gereksinimleri önemsiz kılmaktadır. Bu nedenle algoritmaların işlem karmaşıklığı onun verimliliğini ölçmede standart bir yaklaşımdır.

Bir problemin boyutu (giriş veri miktarı) n ile ölçülür ve n boyutlu bir problemin bir

işleminde gerektirdiği hesap miktarı N ile gösterilir. N ile n arasında fonksiyonel bir ilişki kurulabilir çünkü problemin boyutu n büyüdükçe şüphesiz bir işlem için gerekli hesap miktarı N de büyür. Fakat bu büyüme oranı problem türüne göre farklılık gösterir. Bu farklılık problemlerin zorluk dereceleri ile ilişkilidir ve problem türlerinin sınıflandırılmasında kullanılır [22-24].

3. BÖLÜM

MEMETİK ALGORİTMALAR

3.1. Giriş

Memetik algoritma, zeki problem çözme özelliklerini temel alan ve özellikle ayrık optimizasyon problemleri için geliştirilmiş bir sezgisel algoritmadır. Algoritma popülasyon tabanlı küresel araştırma algoritmaları ile iteratif küresel araştırma algoritmaları veya bölgesel araştırma algoritmalarının bir arada kullanılmasıyla oluşturulmuştur. Popülasyon tabanlı ve iteratif algoritmaların bir arada kullanılması memetik algoritmanın en belirgin özelliğidir. Memetik algoritmaların başarısı da kullanılan bu farklı algoritmaların sinerjisine bağlıdır [25]. Memetik algoritma’da farklı algoritmalarla problem ile ilgili elde edilen bilgiler karşılaştırma-birleştirme gibi stratejilerle değerlendirilmekte ve kullanılan algoritmaların üstün özelliklerinden faydalanılarak küresel araştırma gerçekleştirilmektedir. Memetik algoritma uygulaması ilk olarak Norman ve Moscato tarafından 1989 yılında geleneksel genetik algoritma ve benzetilmiş tavlama algoritmasının bir arada kullanılmasıyla gerçekleştirilmiştir [26]. Norman ve Moscato tarafından gezgin satıcı problemi için uygulanan bu yaklaşım günümüzde kullanılan birçok memetik algoritma modeli için temel oluşturmuştur. Bu bölümde memetik algoritma tanıtılmış ve gerçekleştirilen memetik algoritma modeli izah edilmiştir.

3.2. Memetik Algoritma

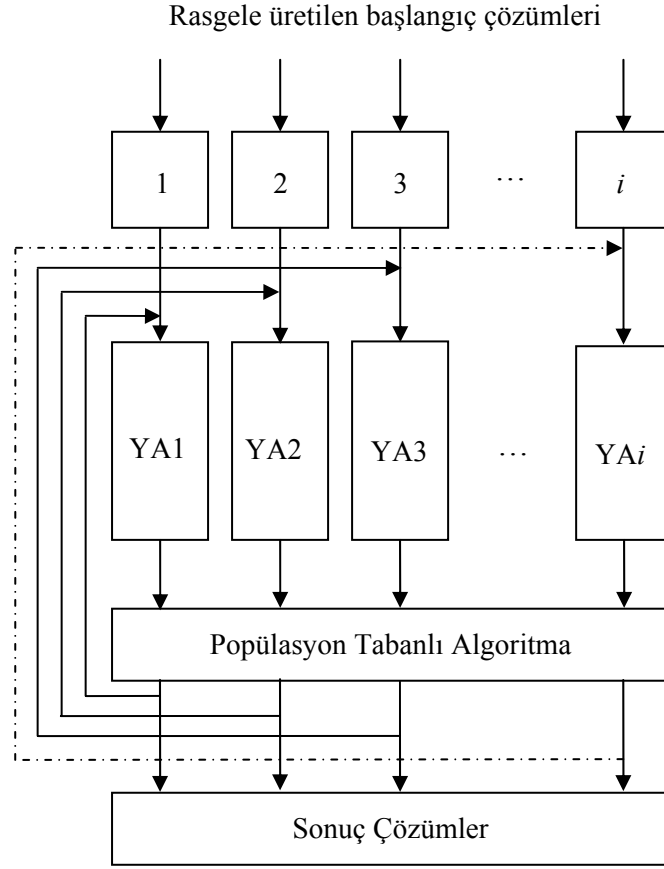
Memetik algoritmanın yerel arama algoritmaları ve küresel arama algoritmalarının bir arada kullanılmasıyla oluşturulduğu yukarıda söylenmişti. Yerel arama algoritmaları arama ve optimizasyon problemlerinin çözümünde en çok kullanılan yöntemlerdendir. Yerel araştırma mekanizması araştırmayı yerel optimum seviyeye veya daha önceden belirlenmiş bir seviyeye ulaştıran bir yaklaşımdır. Bu algoritmalar problemin eldeki bir

çözümünü her yeni iterasyonla daha iyi bir çözüme götürmeye çalışır ve araştırma iyileşmenin durduğu ana kadar devam eder. Yerel arama algoritmalarının optimum çözüme ulaşması büyük oranda başlangıç çözümünün küresel optimumun bulunduğu bölgeye uzaklığına ve iterasyon sayısına bağlıdır. Bütün bunlarla birlikte etkin, verimli ve kolay uygulanabilir olma özellikleriyle çok yaygın kullanılırlar.

Küresel arama algoritmalarında ise, yerel arama algoritmalarından farklı olarak yerel optimallikten kaçmak için gelişmeleri önleyen bazı hareketlerinde kabullenilmesi prensibi vardır.

Memetik algoritmalarda popülasyon tabanlı küresel araştırma algoritması olarak çoğunlukla genetik algoritmalar, yerel arama algoritması olarak da tabu araştırma algoritması, benzetilmiş tavlama algoritması veya tepe tırmanma algoritmaları kullanılmaktadır. Yukarda bahsedildiği gibi memetik algoritmaların bölgesel arama ve küresel arama için ayrı ayrı metotlar kullanması algoritmanın başarımını artırmaktadır.

Memetik algoritmalar paralel çalışan algoritmalarlardır. Paralel algoritmalar geleneksel seri algoritmaların aksine, bazı görev parçalarını veya bir problemin çözümüne ulaşmayı sağlayacak alt parçaların her birini bir anda farklı işleyici üniteler üzerinde çalıştırabilmektedir. Paralellik donanımsal, yazılımsal veya hem donanımsal hem yazılımsal olarak sağlanabilir. Donanımsal paralellik algoritmanın en az iki veya daha fazla işlemci üzerinde çalışmasıyla sağlanır. Yazılımda paralellik ise, donanımsal paralellikten farklı anlamda birden fazla algoritmanın aynı veri üzerinde çözümler araması ve daha önceden belirlenmiş bir anda birbirleriyle haberleşmesiyle sağlanır. Paralel mimarilerin genel olarak kısa zamanda daha uygun çözümlerin elde edilmesini kolaylaştırdığı söylenebilir [27-31]. Bu özelliklerinden ötürü çok işlemcili mimari ve paralel çalışan algoritmalar normal algoritmalara göre son zamanlarda daha yaygın kullanılmaktadırlar. Şekil 3.1’de memetik algoritmanın genel blok diyagramı verilmiştir.



Şekil 3.1. Memetik algoritmanın blok şeması.

Şekil 3.1’den görüldüğü gibi, memetik algoritmada paralel çalışan yerel araştırma algoritmaları (YA) farklı başlangıç noktalarından çözüm uzayını araştırmakta ve belirli bir işlem sayısı sonunda bulunan aday çözümler diğer çözümlerle birleştirilerek ve/veya karşılaştırılarak popülasyon tabanlı diğer algoritmaya aktarılmaktadır. Karşılaştırma işlemi genel bir karşılaştırma metodu olabileceği gibi, genetik algoritmadaki seçim işleminin benzeri de olabilir. Seçilen bireylerin birleştirilme işlemi ise genetik algoritmadaki çaprazla mekanizması veya yeni çözümler üretmenin başka yolları şeklinde anlaşılabilir. Birleştirme sayesinde bilginin karşılıklı değişimi gerçekleşmiş olur. Bu karşılaştırma ve birleştirme işlemi istenen kriter sağlanana kadar tekrarlanır. Bu yapılan işlemler sayesinde popülasyonda ölçülebilir bir çeşitlilik sağlanmış olacaktır. Karşılaştırma yaklaşımında kendi içerisinde daha iyiyi arayan çözümlerin sonuçları gruplanarak karşılaştırılmakta ve diğer grup sonuçlarıyla birleştirilmektedir. Grupların karşılaştırılması ve birleştirilmesi sonucu popülasyonda çok farklı verilerin etkileşimi

sağlanmaktadır. Popülasyon tabanlı algoritma tarafından geliştirilen çözümler belirli bir işlem sayısından sonra tekrar geliştirilmek üzere yerel arama algoritmalarına aktarılmaktadır. Bu süreç araştırma boyunca devam etmektedir.

Literatürde farklı şekillerde oluşturulmuş memetik algoritma modelleri vardır. En çok rastlanılan memetik algoritma modellerini şu şekilde sıralayabiliriz: Genetik algoritmanın tabu araştırma algoritmasıyla (GA+TA), genetik algoritmanın benzetilmiş tavlama algoritmasıyla (GA+BT) ve genetik algoritmanın tepe tırmanma algoritmasıyla (GA+TT) birlikte kullanılması. Bu algoritmaların sürekli optimizasyon problemlerine uygulanmasına ilişkin literatürde çeşitli sayıda çalışmalar mevcuttur [32-35].

Memetik algoritma modelinde popülasyon tabanlı algoritma olarak çoğunlukla GA'nın kullanılması, GA topluluklarının memetik algoritmanın GA'nın bir türü olduğunu savunmalarına sebep olmaktadır. Goldberg "Zen and the Art of Genetic Algorithms" adlı kitabında memetik algoritmanın bir çeşit hibrid GA olduğunu savunmuştur [36]. Paralel GA'lar özellikle karmaşık ve büyük optimizasyon problemleri için geliştirilen iki veya daha fazla temel GA'nın paralel çalışması suretiyle elde edilen bir modeldir. Belirli zamanlarda paralel çalışan GA'lar arasında sağlanan haberleşme ile bilgi alışverişi gerçekleştirilmektedir. Aslında paralel çalışan GA'lar da belirli seviyelerde birbirlerinin çözümlerinden faydalanarak memetik algoritmanın bir örneğini oluşturmaktadır. Genel olarak GA canlılardaki biyolojik değişimden esinlenirken, memetik algoritma kültürel değişimi taklit etmeye çalışır. GA'nın temelini oluşturan genlerdeki değişimler yeni nesillerde fark edilirken, memetik algoritmalarda ki gen olarak nitelendireceğimiz meme'lerdeki değişim ise yeni nesile geçmeden insanlar arası etkileşimle veya adaptasyonla gözlenebilir.

Açıkça söylenebilir ki, memetik algoritmalar kendilerini oluşturan diğer algoritmaların etkin operatörlerinin birleşimidir. Bu birleşimde, birleştirilen algoritmaların üstün özelliklerinden faydalanılmaktadır. Memetik algoritma ile elde edilen sonuçlar diğer algoritmalara göre kıyaslandığında daha iyi çözümler çok daha kısa sürelerde bulunabilmektedir. Örneğin Cheng ve Gen yapmış oldukları çalışmada GA ve BT birleşimiyle oluşturdıkları memetik algoritma modelinde GA'nın etkin çözüm uzayı örnekleme, ihtimalci tepe tırmanma (stochastic hill-climbing), aramada bir sonraki aşamayı hatırlama gibi üstün özelliklerinden faydalanırken BT'nin de basit donanım,

küçük popülasyon ve etkin hafıza kullanımı gibi avantajlarını bir araya getirmişlerdir. Bu çalışmada elde edilen sonuçlar yalnız başına GA ve yalnız başına BT algoritmaları ile yapılan çözümlerle kıyaslandığında memetik algoritmanın çok daha etkin olduğu gösterilmiştir [33].

Bu çalışmada, GA ve BT bir arada kullanılarak memetik algoritmanın bir türü oluşturulmuştur. İlerleyen konularda önce GA ve BT algoritmaları hakkında bilgi verilmiş, sonra gerçekleştirilen memetik algoritma modeli tanıtılmıştır.

3.3. Genetik Algoritma

Genetik algoritmalar yapay zekanın bir kolu olan evrimsel hesaplama tekniğinin bir parçasıdır. Adından da anlaşıldığı üzere, evrimsel hesaplama tekniğinin bir parçası olan genetik algoritma Darwin'in evrim teorisinden esinlenerek oluşturulmuştur. Herhangi bir problemin genetik algoritma ile çözümü, tabii seleksiyon ile canlılarda bulunan genetik gelişimi simüle ederek, problemi sanal olarak evrimden geçirmek suretiyle yapılmaktadır. Genetik algoritmanın temelleri J. Holland tarafından atılmıştır [9]. Michigan Üniversitesinde doğaya uyumu esas alan dersler vermekte olan Holland 1970'li yılların başlarında arkadaşları ve öğrencileri ile birlikte Charles Darwin'in (1859) doğal evrim teorisinden yola çıkarak genetik algoritma kuramı üzerinde çalışmaya başlamıştır.

Doğal evrim teorisine göre, doğada mevcut canlılardan bulundukları ortamlara uyum sağlayamayan zayıf üyeler yok olurlar. Ortamlarına uyum sağlayan sağlam üyeler hayatta kalırlar. Her yeni nesil bir önceki neslin en iyi özelliklerini alır ve zamanla çevre şartlarında meydana gelebilecek değişikliklere uyum sağlayacak şekilde değişir ve gelişir. En son nesil, geçmiş nesillere göre en iyi uyumu sağlayan nesildir [10].

Doğal evrimdeki ana amaç, canlıların yaşadıkları çevrede hayatta kalabilmeleri için gerekli gelişimin sağlanmasıdır. Doğadaki canlıların tabiatı hakkındaki bilgiler, onların kromozomlarında kodlanmıştır. Bu kromozomlarda canlının tüm kişisel özellikleri yer alır ve türlerindeki en iyi özelliklere sahip üyelerin özellikleri bir sonraki nesile geçer. Türlerinin zayıf üyeleri ise yaşamak ve bir sonraki nesile özelliklerini aktarmak konusunda fazla şansa sahip değildirler.

Genetik algoritmanın çalışma esasları da doğanın temel prensipleriyle paraleldir. Problemin çözüm kümesinden bir topluluk oluşturulur. Topluluğun üyeleri doğadaki üremeyi taklit eden işlemlere tabi tutulur ve bu üyelerden yeni bir nesil oluşturulur. Her nesilden en kötü üyeler atılır ve en iyi üyeler yaşamaya ve üremeye devam ederler. Oluşturulan her nesille problemin en iyi çözümüne daha fazla yaklaşılr. Bu işlemler sonucunda ulaşılan en son nesildeki en iyi üye çözüm olarak kabul edilir. Bu üye kesin olarak en ideal çözüm olmayabilir ancak ideale en yakın çözümdür.

Holland bu konuyla ilgili çalışmalarını 1975 yılında daha sonraları GA'nın teorik altyapısı sayılan "Adaptation in Natural and Artificial Systems (Doğal ve Yapay Sistemlerde Adaptasyon)" adlı bir kitapta toplamıştır. Goldberg 1989 yılında "Genetic Algorithms in Search Optimisation and Machine Learning" adlı kitabını yayınladıktan sonra GA'nın geniş alanlarda ki uygulamaları artmıştır [37].

Genetik algoritmanın çalışma temellerinin atılmasının ardından çalışma esasları üzerine yoğunlaşmıştır. Syswerda (1989) çaprazlama işlemi üzerinde çalışmalar yapmış ve ikili, tekli ve düzgün çaprazlama sonuçlarını karşılaştırmıştır [38]. Greenwel, Angus ve Finck (1995) GA için en uygun mutasyon ihtimalini araştırmışlardır [39]. De Jong (1975), Schaffer (1989) ve Grefenstette (1986) GA'da kullanılan kontrol parametreleri üzerinde çalışmışlardır [40-42]. 1996'dan sonra GA konusunda yapılan çalışmalarda çok büyük bir artış olmuştur.

Genetik algoritmanın kullanım alanları her geçen gün genişlemektedir. Günümüzde uçakların parametrik tasarımından uluslararası güvenlik modellerine kadar her konuda uygulamalar mevcuttur. GA ile çözümlenen problemlerin hemen hepsi çok geniş bir çözüm uzayının taranmasını gerektirmektedir. Bu çözüm uzayının geleneksel yöntemlerle taranması çok uzun sürmekte, genetik algoritmayla ise kısa bir sürede kabul edilebilir sonuçlar alınabilmektedir.

3.3.1. GA'nın Klasik Optimizasyon Tekniklerinden Farklılıkları

Genel olarak bir GA'nın klasik optimizasyon tekniklerinden farkları aşağıdaki şekilde özetlenebilir [37].

1. GA problemdeki parametrelerin kendileriyle ilgilenmez; onun yerine parametrelerin kodlanmış halleriyle ilgilenir.

Genetik algoritmalar, sonlu bir alfabe üzerinden sonlu uzunlukta diziler kodlamak için, optimizasyon probleminin doğal parametre setini kullanır. Örneğin, herhangi bir fonksiyon maksimize edilmek istendiğinde, diğer optimizasyon metotları parametre değerlerini küçük miktarlarda değiştirerek amaç fonksiyon değerinin en yüksek olduğu yeri bulmaya çalışırlar. Genetik algoritmalarla optimizasyon işleminin yürütülmesinde ise ilk adım, parametresinin sonlu uzunlukta dizi olarak değişik şekillerde kodlanması ve kodlanan parametre ile çalışarak optimizasyonun sağlanmasıdır. Parametrelerin kodlanması sonucunda, diğer metotları sınırlayan birtakım özelliklerde de büyük ölçüde serbestlik sağlanmış olmaktadır.

2. GA çözüm kümesinde tek bir noktadan değil, noktalar topluluğundan araştırma yapar.

Birçok optimizasyon probleminde, tanım aralığı içindeki tek bir noktadan hareketle bazı geçiş kurallarına göre bir sonraki inceleme noktası bulunur. Bu noktadan noktaya yönelme metodu çok sayıda tepe noktası bulunan araştırma uzayı için risklidir. Çünkü bölgesel tepe değerine yaklaşma hatası oluşabilir. GA’larda çok sayıda noktadan oluşan bir veri tabanı ile (dizilerin popülasyonu) çalışıldığından aynı anda pek çok tepe noktasına atlanabilir. Böylece noktadan noktaya geçme metodundaki yanlış tepe noktasının bulunması olasılığı azalır ve bölgesel optimuma yakınsama riski ortadan kalkarak küresel optimuma ulaşma olasılığı artar. Sonuç olarak, GA araştırmaya dizilerden oluşan bir popülasyon ile başlar ve bu dizilerden daha başarılı popülasyonlar üretmeye çalışır.

3. GA klasik tekniklerin aksine yardımcı bilgi veya türev bilgisine ihtiyaç duymaz; sadece uygunluk fonksiyonu hesaplarıyla en iyiyi bulmayı hedefler.

Çoğu araştırma tekniklerinin doğru bir şekilde çalışması için yardımcı bazı bilgilere ihtiyaçları vardır. Örneğin, gradient tekniklerinde tepe değerine doğru yükselmenin olup olmadığını anlamak için nümerik ya da analitik olarak hesaplanan türev bilgisi gereklidir. Ayrıca farklı araştırma problemleri için de çok farklı yardımcı bilgilere ihtiyaç olabilir. Bunun aksine GA’larda hiçbir yardımcı bilgiye ihtiyaç yoktur. Verimli

bir araştırma yapmak için gerekli olan, her bir dizinin değerlendirileceği bir amaç fonksiyonudur. Optimizasyon sırasında problemle ilgili özel birtakım bilgilerin kullanılmaması, GA'ların performansını yükseltmede son derece etkilidir.

4. GA sebep sonuç ilişkileri yerine, olasılıklı değişim metotlarını kullanır.

Genetik algoritmalar araştırmaya yön vermek için, birçok optimizasyon tekniğinin aksine, olasılığa dayalı geçiş kuralları kullanır. Bunun sonucunda araştırmanın, araştırma uzayının hangi bölgesine doğru yöneleceğine karar vermek için rasgele seçim tekniği kullanılır.

Bahsedilen bu dört farklılığın bir arada bulunması, genetik algoritmaların gürbüzlüğüne ve sonuca ulaşma üstünlüğüne olumlu yönde katkıda bulunur.

3.3.2. GA'nın Çalışma Esasları

Basit bir GA her biri algoritmanın performansını önemli derecede etkileyen beş temel elemandan oluşur. Bu elemanlar; gösterim, başlangıç popülasyonunun oluşturulma şekli, uygunluk veya kalite değerlendirme fonksiyonu, genetik operatörler ve kontrol parametreleridir [43].

GA diğer araştırma tekniklerinden farklı olarak popülasyon olarak adlandırılan bir başlangıç çözüm seti ile çalışmaya başlar. Popülasyonun kromozom olarak adlandırılan her üyesi problemin elimizdeki çözümlerinden birini temsil eder. Kromozomlar üreme olarak adlandırılan başarılı yinelenmelerle geliştirilir. Her üremede kromozomlar bazı uyum ölçüleri kullanılarak değerlendirilir. Yeni bir nesil yaratmak için çocuk olarak adlandırılan yeni kromozomlar iki kromozomun çaprazlama operasyonu ile ya da mutasyon operasyonu ile oluşturulurlar. Uyum değerlerine göre ebeveynlerle çocuklar arasında yeni bir nesil seçilir ve diğer bireyler topluluğun dışına atılır. En iyi uyum değerlerine sahip bireylerin seçilme olasılığı daha fazladır. Pek çok nesilden sonra en iyi uyum fonksiyonuna sahip kromozom çözüm olarak alınır. Basit bir genetik algoritmanın temel adımları aşağıda verilmiştir.

Aşağıda temel adımları verilen temel bir GA'nın elbette herhangi bir gerçek problemin çözümünde çok iyi bir performans ortaya koyması beklenemez. Burada belirtilen temel

operatörlere bazı diğer operatörler ilave edilerek algoritmanın performansı geliştirilmeye çalışılmaktadır.

Adım 1. Başlangıç çözümlerinin bir popülasyonunu oluştur (Başlangıç yoğunluğu)

Adım 2. Popülasyondaki her bir çözümün uygunluk değerini hesapla

Adım 3. Araştırmayı durdurma için kriter sağlanıyorsa araştırmayı durdur. Aksi taktirde

Tabii seleksiyon işlemini uygula

Çaprazlama işlemini uygula

Mutasyon işlemini uygula

Adım 4. Adım 2 ye git.

3.3.2.1. Kodlama

Kodlama problemdeki değişkenlerin genetik algorithmada kullanılabilecek şekilde modellenmesidir. Değişkenlerin modellenmesiyle kromozomlar oluşur. Kodlama yaygın olarak ikili kodlama ve gerçek değerli kodlama olmak üzere iki şekilde yapılmaktadır. İkili kodlamada 0 ve 1 değerleri kullanılır. Gerçek değerli kodlamada ise değişkenler gerçek değerleriyle modellenir. Kodlama örnekleri aşağıda görülmektedir.

$$A = [1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0] \quad , \text{ikili kodlama}$$

$$B = [12.34 \ 25.00 \ 35.46 \ 102.34] \quad , \text{gerçek değerli kodlama}$$

Düşük boyutlu ve yüksek hassasiyet gerektirmeyen nümerik problemlerde kromozomu ikili düzendeki sayılar dizisiyle ifade etmek çok tercih edilen bir temsil şeklidir. Basit olması, genetik operatörlerin kolay uyarlanabilmesi ve bilgisayar tarafından daha hızlı bir biçimde işlenebilmesi tercih sebebidir. Tersine çok boyutlu ve yüksek hassasiyet gerektiren problemlerde ise ikili düzendeki sayılarla temsil algoritmanın performansını düşürmektedir. Kromozom dizisinin uzunluğu parametre değerlerine ve istenen hassasiyete bağlıdır [44]. Örneğin parametre değer aralığı (-100,100) ve gerekli hassasiyet ondalık noktadan sonra altı basamak ise (-100,100) aralığının iki yüz milyon eşit aralığa bölünmesi gerekir. Bu da dizi uzunluğunun çok büyük olmasını gerektirir.

3.3.2.2. Başlangıç Yoğunluğunun Oluşturulması

Genetik algorithmalarda başlangıç yoğunluğu oluşturulurken farklı yollar izlenebilir.

Şayet problem için başlangıçta çözümler kabaca biliniyorsa bu durumda başlangıç yoğunluğu bu çözümlerden oluşturulabilir. Çünkü bu yöntem, optimal çözüm bulmada zaman açısından tasarruf sağlamaktadır. Çözümler kabaca bilinmiyorsa genetik algoritmaların başlangıç yoğunluğu çoğunlukla rasgele sayı üreticisi kullanılarak oluşturulur.

3.3.2.3. Uygunluk Fonksiyonu

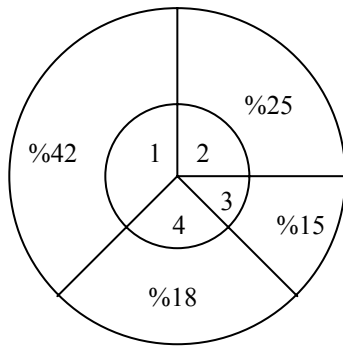
Uygunluk fonksiyonu kromozomların problemde amaçlanan çözüme ne kadar yaklaştığını hesaplayan fonksiyondur. Uygunluk fonksiyonu GA ile çözülecek problem arasındaki bağlantıdır. Bu fonksiyonla değişkenlerin modellendiği kromozomların çözülecek problemdeki performansları ölçülür. Amaç fonksiyonundan elde edilen kaba sonuçların kullanımı araştırmanın başlangıcında yeterli olabilir. Ama çoğunlukla araştırma ilerledikçe iyi ile daha iyi çözümler arasındaki farkı ayırt etmek zorlaşır. Bu durumda genetik algoritmanın optimal bir çözüm bulması oldukça zorlaşır. Bu yüzden kalite değerlendirme biriminde normalizasyon işlemine gerek duyulur. Araştırma bir taraftan iyi çözümlerin bulunduğu tarafa doğru odaklanmaya çalışırken, diğer taraftan algoritmanın küresel optimizasyon işlemini gerçekleştirebilmesi için araştırmaya yeterli bir ıraksamanın sağlanması gerekir. Aksi takdirde erken yakınsama problemi ortaya çıkacak ve tüm araştırma uzayının taranması mümkün olmayacaktır. Normalizasyon işleminde en yaygın olarak kullanılan teknik ölçekleme penceresi (scalling window) işlemidir [42].

3.3.2.4. Doğal Seçim

Doğal seçim uyum fonksiyonuna göre, en iyi sonucu veren kromozomların bir sonraki nesilin oluşturulmasına etki etmek üzere seçilmesidir. Bu mekanizma kaliteli olan bireylerin hayatta kalma prensibine göre çalışır. Tabiatı bu seçme işlemi çevre tarafından kontrol edilir. Yapay sistemlerde ise amaç fonksiyonu ile diğer kalite değerlendirme işlemleri tarafından kontrol edilir. Bireyler bir kuşaktan diğerine geçerken kalite değerlerine göre seçilme şanslarına sahip olurlar. Seçim işlemlerinden bazıları rulet tekerleği metodu, turnuva metodu ve derecelendirmeli (sıralı) seçimdir.

Rulet tekerleği metodunda kromozomlar uygunluk fonksiyonu değerlerine göre bir rulet

etrafında gruplanır. Rasgele sayı üretici tarafından üretilen bir değer yardımıyla rulet tekerleği üzerinden bir birey seçilir. Bu yaklaşımda uygunluk fonksiyonu değeri daha iyi olan bireyin seçilme olasılığı daha yüksek olmaktadır [45]. Rulet tekerleği seçiminde eğer bireylerin uygunluk değerleri arasındaki fark çok fazla ise sorunla karşılaşılabilir. Örneğin, en iyi kromozomun uyumluluğu %90'ın üzerinde ise diğer kromozomların seçilme şansı çok azalacaktır. Turnuva metodunda ise tüm kromozomlardan rasgele seçilen 2 (veya daha fazla) kromozom kendi aralarında karşılaştırılır ve uyum fonksiyon değeri daha iyi olan seçilir. Derecelendirmeli seçim metodunda ise kromozomlar uyum fonksiyon değerlerine göre sıralanırlar ve bu sıralamaya göre orantılı olarak seçilme şansına sahip olurlar.



Dizi	Uyum fonksiyon değerleri (kalitesi)	Seçilebilme Oranı
1	210	%42
2	125	%25
3	75	%15
4	90	%18

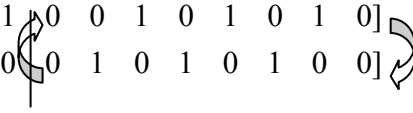
Şekil 3.2. Rulet seçme tekniği

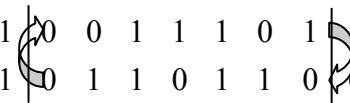
3.3.2.5. Çaprazlama

Çaprazlama işlemi gerçek dünyada çiftleşme işleminin algoritmadaki karşılığıdır. Bir sonraki neslin üremesi için seçilen kromozomlar ikiye bölünür ve eşleştirilerek çaprazlamaya tâbi tutulurlar. Çaprazlama operatörü bu iki yapıdan yeni iki yapı meydana getirmek için kullanılır. Tek noktalı çaprazlama, iki noktalı çaprazlama ve düzgün çaprazlama yaygın olarak kullanılan çaprazlama yöntemlerinden bazılarıdır.

Tek noktalı çaprazlama işleminde kromozomlar rasgele belirlenen bir noktadan bölünür

ve noktanın sağında kalan parçaları yer değiştirir. İki noktalı çaprazlamada ise kromozomlar iki noktadan parçalara ayrılır ve bu noktaların arasında kalan kısımlar yer değiştirir. Düzgün çaprazlamada kromozomlar bölünmez. Önce rasgele olarak çaprazlama kalıbı oluşturulur; daha sonra bu kalıba göre kromozomların genleri yer değiştirir. Tek noktalı ve iki noktalı çaprazlama örnekleri Şekil 3.3'te görülmektedir.

$$\begin{array}{l}
 k_1 = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0] \\
 k_2 = [0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0]
 \end{array}$$


$$\begin{array}{l}
 k_1 = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0] \\
 k_2 = [0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0]
 \end{array}$$


Şekil 3.3. Tek noktalı ve iki noktalı çaprazlama işlemi

3.3.2.6. Mutasyon

Mutasyon operatörü de, çaprazlama operatörü gibi tabiatta bulunan genetik mutasyon olayını simüle eder ve genetik algoritmanın başarısında önemli rol oynar. Mutasyon işlemi, yeni oluşan kromozomların bazı genlerinde meydana gelen rasgele değişim işlemidir. Mutasyon işlemi uygulanacak olan genler oldukça küçük tutulan bir mutasyon oranına göre rasgele olarak belirlenir. İkili kodlamada mutasyon uygulanan genin önceki değeri 0 ise 1 olarak, 1 ise 0 olarak değiştirilir. Gerçek değerli kodlamada genlerin değeri ise, rasgele belirlenen oranda değiştirilir. Şekil 3.4'de ikili kodlamada mutasyon işlemi gösterilmiştir.

mutasyon uygulanan gen

↓

$$\begin{array}{l}
 k'_1 = [0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0] \\
 k''_1 = [0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0]
 \end{array}$$


Şekil 3.4. İkili kodlamada mutasyon uygulaması

3.3.2.7. Yer Değiştirme

Belirtilen genetik algoritma işlemleriyle yeni kromozomlar oluşturulduktan sonra yeni ve eski nesil kromozomlardan bir çözüm grubu oluşturulur. Bu işlem nesilsel yer değiştirme ve durağan durumlu yer değiştirme olmak üzere iki şekilde yapılabilir. Nesilsel yer değiştirmeye göre eski nesil kromozomların tamamı silinip yeni nesil kromozomların tamamı kullanılır. Durağan durumlu yer değiştirmeye göre ise, eski neslin en iyi kromozomları yeni nesilde de yer alır. Eski neslin en iyi kromozomunun yeni nesildeki herhangi bir kromozomla yer değiştirme işlemine elitizm denir [45,46].

3.3.2.8. Genetik Algoritmada Kullanılan Kontrol Parametreleri

Genetik algoritmanın çalışması esnasında kullanılan popülasyon büyüklüğü, çaprazlama oranı ve mutasyon oranı gibi parametreler vardır. Bu parametrelerin uygun şekilde belirlenmesi algoritmanın performansı üzerinde son derece önemlidir.

Popülasyon büyüklüğü: Popülasyon sayısının küçülmesi, yetersiz örneklemeye sebep olacak, ıraksamayı sağlamak zorlaşacak ve araştırma bölgesel bir optimal noktaya sürüklenecektir. Tersine popülasyon sayısının çok büyük seçilmesi ise araştırma zamanının aşırı uzamasına neden olacaktır. Bu sebeplerden dolayı probleme uygun bir popülasyon büyüklüğü belirlenmelidir.

Çaprazlama Oranı: Yüksek bir çaprazlama oranı araştırma uzayının çok hızlı bir şekilde araştırılmasına sebep olacaktır. Diğerlerine göre daha iyi olan bireyler yeni üreme işlemlerinden sonra çok hızlı bir şekilde bozulacaktır. Düşük bir çaprazlama oranı ise, üreme sonucu oluşan yeni nesile çok az sayıda yeni ve farklı bireylerin girmesine sebep olacak, araştırmayı belli bir yerde tıkayacaktır.

Mutasyon Oranı: Mutasyon operatörü araştırma uzayına yeni bireylerin girmesini sağlar. Yüksek bir mutasyon oranı, araştırmaya aşırı bir rasgelelik kazandıracak ve ıraksamayı hızlandıracaktır. Tersine düşük bir mutasyon oranı ise ıraksamayı yavaşlatacak ve araştırma uzayının tamamen araştırılmasını engelleyecektir. Dolayısıyla optimal bir çözümde takılı kalınacaktır.

De Jong, Schaffer ve Grefenstette kontrol parametrelerinin genetik algoritmanın

performansı üzerindeki etkisini incelemek için bazı test problemleri üzerinde çeşitli deneysel çalışmalar yapmışlardır. Bu çalışmalar sonucunda kontrol parametreleri için çeşitli değerler önermişlerdir. Önerilen bu değerler Tablo 3.1’de verilmiştir [40-42].

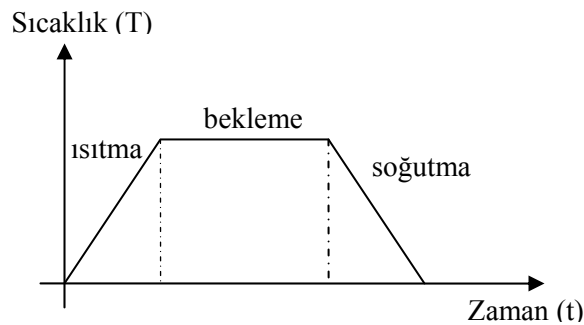
Tablo 3.1. Kontrol parametreleri için önerilen değerler

Parametre	De Jong	Schaffer	Grefenstette
Popülasyon Büyüklüğü	50–100	20–30	30
Çaprazlama Oranı	0.6	0.75–0.95	0.95
Mutasyon Oranı	0.001	0.005–0.01	0.01

Bununla birlikte optimal parametre değerleri problemden probleme değişmektedir. Bundan dolayı iyi bir performans elde etmek için parametre değerlerinin verilen problem için elde edilmesi gerekmektedir.

3.4. Benzetilmiş Tavlama Algoritması

Olasılık tabanlı sezgisel bir algoritma olan benzetilmiş tavlama algoritması ilk olarak Kirkpatrick, Gelatt ve Vecchi tarafından önerilmiştir [13]. Benzetilmiş tavlama algoritması katıların fiziksel tavlama işlemi ve kombinatoriyal optimizasyon problemlerinin çözümü arasındaki benzerlik üzerine dayalıdır. Isıl işlem sayesinde malzemenin belirli bir sıcaklığa kadar ısıtılması (ısıtma) bu sıcaklığın uygun bir süre tutulması (bekleme) ve belirli bir programa uygun olarak sıcaklığın düşürülmesi (soğutma) ile üç kademede özellik değişimleri sağlanır.



Şekil 3.5. Isıl işlem kademeleri

Katının ergime noktasına kadar ısıtılması ve devamında mükemmel kafes yapılı durumda kristalize olana kadar yavaşça soğutulması işlemine tavlama denilmektedir. Malzemelerin atomları yüksek sıcaklıklarda yüksek enerji seviyelerindedir ve düzgün yerleşimler için daha fazla hareket serbestliğine sahiptirler. Düzgün yapılı bir kristal sağlandığında, sistem minimum enerjiye sahiptir. Sıcaklık azaltıldıkça, atomik enerji düşer. Eğer soğutma işlemi çok hızlı gerçekleşirse, kristal yapıda bozukluklar ve düzensizlikler ortaya çıkacaktır.

Fiziksel tavlama işlemi, Monte Carlo tekniği üzerine dayalı olarak Metropolis ve arkadaşları tarafından modellenmiştir [47]. Verilen bir T sıcaklığında, sistem enerjilerinin olasılık dağılımı aşağıda verilen termodinamik kanunu ile belirlenir:

$$P(E) = e^{-E/(kT)} \quad (3.1)$$

Burada, E sistem enerjisini, k ise Boltzmann sabitidir.

Küçük bir karışıklıkla sistem durumunda değişiklik yaratılması halinde, Metropolis algoritmasına göre sistemin yeni enerjisi hesaplanır. Eğer enerji azalmış ise, sistem bu yeni duruma geçer. Eğer enerji artmış ise yeni durumun kabul edilip edilmemesine, Eşitlik (3.1) de verilen olasılık formülü kullanılarak karar verilir: Uniform dağılımdan $(0,1)$ aralığında rasgele bir δ sayısı üretilir ve Eşitlik (3.2) de verilen şart sağlanıyorsa yeni çözüm, mevcut çözüm olarak kabul edilir. Aksi takdirde, mevcut çözüm değiştirilmez.

$$\delta \leq e^{\Delta E/T} \quad (3.2)$$

Burada, ΔE , iki durumun enerji seviyeleri arasındaki farktır. Bu kabul kriteri *Metropolis kriteri* olarak bilinir. Eşitlik (3.1)' e göre, yüksek sıcaklıklarda tüm enerji durumları için $P(E)$, 1'e yakınsar. Düşük sıcaklıklarda bile sistemin yüksek enerji seviyesine sahip olması küçük bir olasılıkla görülebilir. Bu nedenle enerjilerin istatistiksel dağılımı, sistemin bir yerel enerji minimumundan çıkmasına izin verir.

Kombinatoryal optimizasyon problemi ve tavlama işlemi arasındaki benzerlikte, katının durumları optimizasyon probleminin mümkün olan çözümlerini temsil eder ve bu durumların enerjileri çözümler için hesaplanan amaç fonksiyon değerlerine karşılık

gelir. Minimum enerji durumu problem için optimal çözümü ifade eder, hızlı soğutma işlemi ise yerel optimum olarak görülebilir. Benzetilmiş tavlama algoritması iteratif bir algoritmadır, yani algoritma çözüm uzayında sayıların vektörü formundaki tek bir çözümü sürekli olarak geliştirme şeklinde çalışır. Temel bir benzetilmiş tavlama algoritmasının adımları aşağıda verilmiştir:

Adım 1. Bir başlangıç çözümü üret, S .

Adım 2. Bir $s' \in N(S)$ çözümü seç ve amaç fonksiyon değerlerindeki farkı hesapla.

$$\Delta = C(S) - C(s')$$

Adım 3. Eğer, i) s' , S den daha iyi ($\Delta > 0$) yada (ii) $\delta \leq e^{\Delta E / T}$ ise,

s' nü yeni çözüm olarak tayin et ($S \leftarrow s'$)

Aksi takdirde, mevcut çözümü tut.

Adım 4. Sıcaklığı güncelle.

Adım 5. Eğer Durdurma kriteri sağlanıyor ise dur, Aksi takdirde Adım 2' ye git.

Benzetilmiş tavlama algoritmasının bir probleme uygulanması amacıyla tasarımında karar verilmesi gereken kavramları, probleme özel seçenekler ve algoritmanın kendisine ait genel seçenekler şeklinde iki grupta toplamak mümkündür [48-50].

3.4.1. Probleme Özel Seçenekler

Karar verilmesi gereken seçeneklerin bir kısmı çözülmesi hedeflenen optimizasyon problemi ile ilgilidir. Bu seçeneklerin tanımlamalarını değiştirmek suretiyle algoritmandan daha iyi performans elde edilebilir. Probleme özgü seçimler aşağıdaki şekilde sıralanabilir.

- Problem, mümkün çözümlerin kümesi tanımlanabilecek şekilde formüle edilmelidir. Çözümlerin temsili (kodlama yapısı), minimize edilecek bir amaç fonksiyon tanımlanması ve bir başlangıç çözümü üretilmelidir.
- Komşu çözümlerin üretilmesini sağlayan komşu üretme mekanizması tanımlanmalıdır. Algoritmanın performansı kullanılan komşuluk yapısına oldukça bağlıdır.
- Hareket seçimleri için uygun stratejilerin tanımlanması

3.4.2. Algoritmanın Kendisine Ait Genel Seçenekler

Genel seçenekler soğutma stratejisinin elamanlarını tanımlar. Soğutma planına ait aşağıda verilen seçimler belirgin şekilde yapılmalıdır.

- Sıcaklık parametresi T 'nin başlangıç değerinin tayin edilmesi
- Soğutma oranı ve güncelleme kuralının tanımlanması
- Her bir sıcaklıkta icra edilecek iterasyonların sayısının belirlenmesi
- Algoritmayı durdurmak için durdurma kriterinin tayin edilmesi

Benzetilmiş tavlama algoritmasının performansı önemli derecede seçilen soğutma tarifesine bağlıdır. Uygun bir soğutma tarifi ile çoğu optimizasyon problemi için optmale yakın çözümler elde edilebilir. Önerilen en eski soğutma tarifi Kirckpatrick ve arkadaşlarının fiziksel tavlama ile olan benzerliğe dayanarak ileri sürdükleri plandır. Bu tavlama planına göre, maddenin sıvı safhaya ulaştığında tüm parçacıkların rasgele düzenlenmesini taklit etmek için, T sıcaklık parametresinin başlangıç değeri, denenen tüm hareketler kabul edilecek kadar yüksek seçilmiştir. Sıcaklık parametresinin değerini azaltmak için ise, oransal bir sıcaklık fonksiyonu kullanarak sabit bir r için, $T(t+1)=r.T(t)$ dikkate alınmıştır. Burada, r değeri 1'den küçük fakat 1'e yakın bir sabittir ve genellikle değeri 0.8 ile 0.99 arasında bir değer almaktadır. Bu sıcaklık fonksiyonu ile sıcaklık parametresinin değeri, sıfıra yaklaştıkça daha da yavaş azalmaktadır. Sıcaklık parametresinin her değerinde gerçekleştirilecek yeterli tekrar sayısı sabit bir üst sınıra göre belirlenerek problemin, fiziksel tavlamadaki ısı dengeye karşılık gelen bir denge durumuna ulaşması amaçlanmaktadır. Bu tavlama planı ile sıcaklık parametresinin her değerinde elde edilen çözüm, belli sayıda ardışık sıcaklık değişimleri boyunca aynı kalırsa benzetilmiş tavlama algoritması durdurulmaktadır. Buna göre elde edilen son durum, fiziksel tavlamadaki *donma durumuna (frozen state)* karşılık gelmektedir. Takip eden bölümde önerilen soğutma tarifeleri verilmektedir.

3.4.2.1. Teorik Soğutma Stratejileri

Teorik soğutma stratejileri benzetilmiş tavlama algoritmasının 1.0'a eşit olasılıkla küresel optimal çözümlere yakınsamasını sağlar. Teorik soğutma stratejilerinde iki tip formülasyon ortaya çıkmıştır.

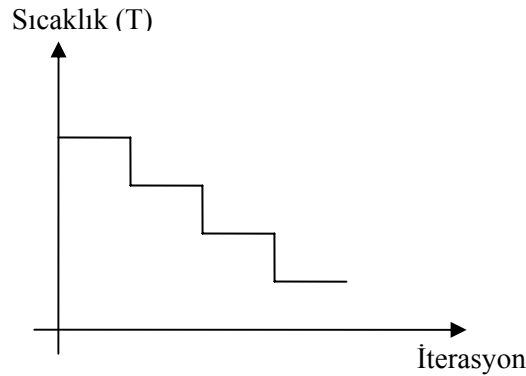
Homojen algoritma: Bu algoritmada sıcaklık, sıfıra erişinceye kadar sonsuz kademeler dizisi şeklinde azaltılır. Verilen bir sıcaklık değerinde gerçekleştirilen iterasyon sayısı belirlenir.

Homojen olmayan algoritma: Bu algoritmada da yine sıcaklık sıfıra erişinceye kadar sonsuz kademeler dizisi şeklinde azaltılır. Bununla birlikte her bir kademe sadece tek bir iterasyon hareketini temsil etmektedir.

3.4.2.2. Basit Soğutma Stratejileri

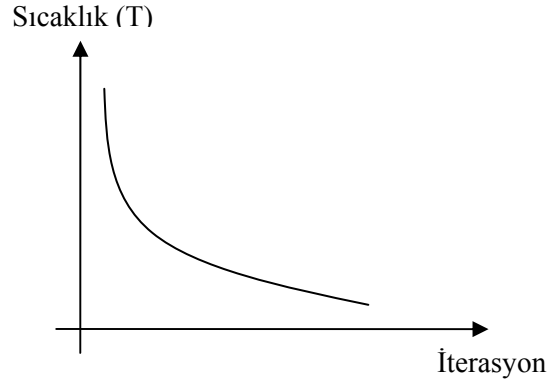
Pratikte basit soğutma stratejileri teorik soğutma stratejilerine göre daha başarılı neticeler üretmektedir. Bunun sebebi teorik olanların yakınsamayı ispat edebilme ihtiyacı tarafından motive edilmesi, ama bununda yakınsamayı yavaşlatması olabilir.

Basamak tip sıcaklık düşürme stratejisi: Bu kategori, belirli bazı kurallara göre sıcaklığı bir kademe düşürmeden verilen sıcaklık değerinde sabit bir Markov zincir uzunluğunun (L) uygulandığı uygulamaları içerir.



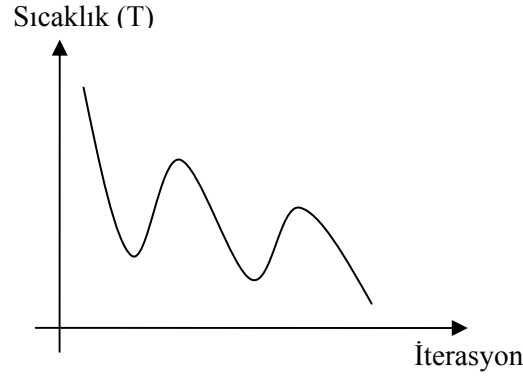
Şekil 3.6. Basamak tipte sıcaklık düşürülmesi

Sürekli sıcaklık düşürme stratejileri: Bu kategori her sıcaklık değerinde sadece bir hareketin yapıldığı soğutma tarifelerini içerir. Sıcaklık T , her iterasyondan sonra düşürülerek sürekli azaltılır. Şekil 3.7’de sürekli sıcaklık düşürme stratejisi gösterilmiştir.



Şekil 3.7. Sürekli olarak sıcaklığın düşürülmesi.

Monotonik olmayan sıcaklık düşürme stratejileri: Monoton olmayan sıcaklık düşürme stratejisinde, her iç değişimden sonra sıcaklık yavaş yavaş azaltılmaktadır. Komşuluğun tamamen araştırılması sonucu herhangi bir hareket kabul edilemez ise sıcaklık artışı meydana getirilmektedir. Şekil 3.8’de monoton olmayan sıcaklık düşürme stratejisi gösterilmiştir.



Şekil 3.8. Monoton olmayan şekilde sıcaklığın düşürülmesi.

Bu strateji yukarda bahsettiğimiz diğer soğutma stratejilerinin dezavantajlarını yok etmek için geliştirilmiştir. Önceki soğutma tarifelerinde, komşuluk genellikle rasgele alternatif çözümler üretmek için araştırılmakta ve sıcaklığın düşük değerlerinde, mevcut çözümden daha kötü çözümlerin kabul edilebilirlik olasılığı çok azalmaktadır.

Son yıllara kadar benzetilmiş tavlama algoritması çoğunlukla ayrık formdaki optimizasyon problemlerine uygulanmakta ve başarılı sonuçlar alınmakta iken, son zamanlarda farklı mühendislik sahalarından değişik nümerik optimizasyon problemlerine de uygulanmıştır [51-53].

3.5. Gerçekleştirilen Memetik Algoritma Modeli

Genetik algoritma ve benzetilmiş tavlama algoritmasının bir arada kullanılmasıyla oluşturulan model memetik algoritma için iyi bir örnektir. Bu model ilk olarak Brown, Huntley ve Spillane tarafından permutasyon problemi çözümünde kullanılmıştır [54]. Oluşturulan memetik algoritma modelinde GA ve BT'nin etkin operatörlerinin birleşimi sağlanmıştır. Bu birleşimde, birleştirilen algoritmaların üstün özelliklerinden faydalanılmıştır. GA'nın etkin çözüm uzayı örnekleme, ihtimalci tepe tırmanma, aramada bir sonraki aşamayı hatırlama gibi üstün özelliklerinden faydalanırken, BT'nin de basit donanım, küçük popülasyon ve etkin hafıza kullanımı gibi avantajları bir araya getirilmiştir. Gerçekleştirilen memetik algoritma modelinin temel adımları aşağıda verilmiştir:

Adım 1. Genetik algoritmanın parametre değerlerinin belirlenmesi

Adım 2. Benzetilmiş tavlama kullanarak genetik algoritma için başlangıç popülasyonunun oluşturulması

Adım 3. Genetik algoritma kullanılarak iyi aday çözümler üretilmesi

Adım 4. Elde edilen her iyi aday çözüm için

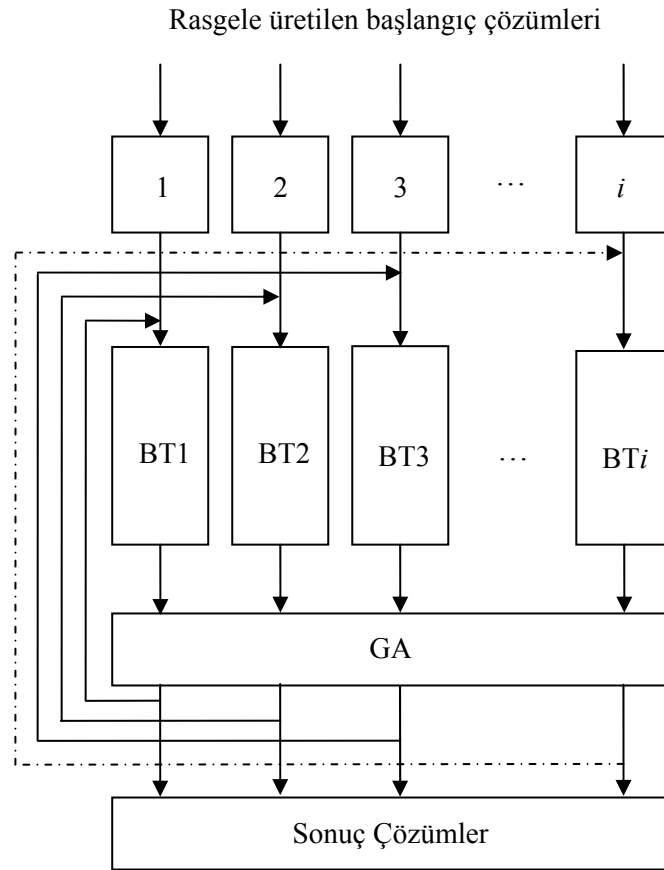
Benzetilmiş tavlama algoritmasının parametrelerini yenile

Benzetilmiş tavlama kullanarak çözümleri geliştir

Adım 5. Adım 3 ve Adım 4'ü tekrarla

Bu modelde birden fazla benzetilmiş tavlama algoritması birbirinden bağımsız olarak araştırma uzayında çözüm aramaktadır. Her bir BT için araştırma uzayı diğerlerinden ayrılmıştır. Bu nedenle bir BT'ye ait sıcaklık değeri, çözüm kalitesi, her bir sıcaklıkta icra edilecek iterasyon sayısı gibi parametreler bir başka BT'ye ait araştırma uzayında doğrudan etkili değildir. Paralel çalışan BT'ler arasındaki etkileşim ve bilgi paylaşımı belirli zaman dilimlerinde bulunan çözümlerin genetik algoritmaya aktarılmasıyla sağlanmaktadır. Bunun için belirli iterasyon değerlerinde her bir BT'nin bulunduğu en iyi

çözüm değerleri genetik algoritmanın popülasyonunu oluşturmaktadır. Genetik algoritma için başlangıç popülasyon değerlerinin belirlenmesinin ardından algoritmaya ait operatörler kullanılarak iyi çözüm etrafında daha iyi yeni çözümler aranmaktadır. Genetik algoritma sonrasında elde edilen yeni çözümler tekrar iyileştirilmek için birbirinden bağımsız çalışan BT'lere verilmektedir. İyileştirme işlemi benzer şekilde tekrarlanarak devam etmektedir. Bu işlemler esnasında mevcut küresel en iyi çözümden daha iyi bir çözüm bulunmuşsa bu çözüm yeni küresel en iyi çözüm olarak saklanarak, küresel en iyi çözümün sürekli olarak korunması sağlanmaktadır. Gerçekleştirilen memetik algoritma modelinin blok diyagramı Şekil 3.9'da verilmiştir.



Şekil 3.9. Gerçekleştirilen memetik algoritma modeli

Şekil 3.9'da görüldüğü gibi paralel çalışan i adet BT algoritması araştırma uzayını rasgele üretilen başlangıç noktalarından taramaya başlar. Belirli bir iterasyondan sonra her bir BT'den elde edilen en iyi çözümler hafızaya alınarak genetik algoritmaya aktarılır. Bu sayede BT'ler tarafından geliştirilen bölgesel en iyi çözümler arasında bilgi

paylaşımı sağlanmış olur.

Genetik algoritma ve benzetilmiş tavlama algoritmalarının etkin operatörlerinin birleştirilmesiyle oluşturulan memetik algoritma her iki algoritmanın da kontrol parametrelerinin etkin şekilde ayarlanmasına ihtiyaç duymaktadır. Bu parametrelerin uygun şekilde belirlenmesinin algoritmanın performansı üzerinde son derece etkili olduğu yapılan benzetim çalışmalarında görülmüştür. İlerleyen konularda detaylı olarak verileceği gibi bu parametre değerleri sürekli test fonksiyonlarının optimizasyonunda ve kontrol sistem tasarımlarında problemin özelliklerine göre farklı değerlerde seçilmiştir. Ayarlanacak parametre değerlerinin fazla olması algoritmanın en belirgin dezavantajı olarak söylenebilir.

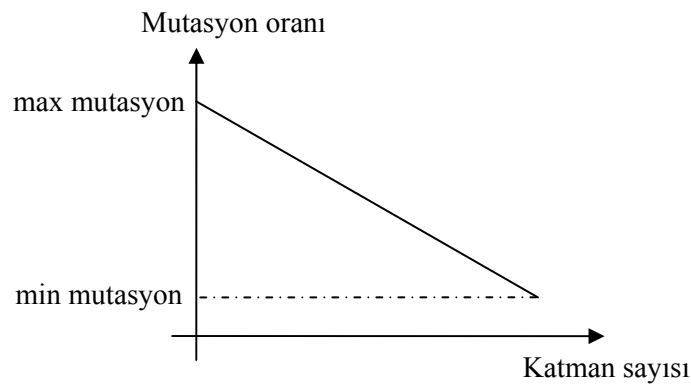
Uygulaması gerçekleştirilen MA modelinde ikili kodlama ve gerçek değerli kodlamanın her ikisi de kullanılmıştır. Yapılan çalışmalarda, parametre sayısı az olan ve yüksek hassasiyet gerektirmeyen test fonksiyon çözümlerinde ikili kodlama yönteminin algoritmanın performansını artırdığı görülürken, tersine çok boyutlu ve yüksek hassasiyet gerektiren test fonksiyonlarında ikili kodlama yönteminin algoritmanın performansını düşürdüğü görülmüştür. Kontrol sistem tasarımlarında optimize edilecek denetleyici parametre sayısının az olması, çaprazlama ve mutasyon gibi genetik operatörlerin kolaylıkla uyarlanabilmesi ve bilgisayar tarafından daha hızlı bir biçimde işlenebilmesinden dolayı genetik algoritmada ikili kodlama kullanılırken, sayısal işlemlerin kolaylıkla yapılabilmesi için benzetilmiş tavlama algoritmasında gerçek değerli kodlama kullanılmıştır.

Memetik algoritmada paralel çalışan BT'ler için başlangıç çözümleri verilen aralıklarda rasgele sayı üretici tarafından oluşturulmuştur. BT ile elde edilen iyileştirilmiş çözümler genetik algoritmanın başlangıç yoğunluğunu oluştururken, paralel çalışan BT sayısı genetik algoritmanın popülasyon büyüklüğünü belirlemektedir. Popülasyon büyüklüğü literatürdeki çalışmalara benzer şekilde 10, 20, 30, 40 ve 50 birey şeklinde ve rasgele seçilerek her problem için uygun popülasyon büyüklüğü belirlenmiştir.

Gerçekleştirilen memetik algoritma modelinde çaprazlama ve mutasyon oranları araştırmaya aşırı bir rasgelelik kazandırmayacak, yerel optimal çözümde takılmayacak ve araştırma uzayını makul bir süre içerisinde tarayacak şekilde belirlenmeye

çalışılmıştır. Çaprazlama oranı %10 - %100 aralığında %10'luk artışlarla incelenmiş ve en iyi sonuçlar %70 ve %90 aralığında elde edilmiştir. Her bir evrimde yapılacak çaprazlama sayısı popülasyon büyüklüğüne bağlı olarak ayarlanırken, çaprazlama yapılacak bireylerin seçimi derecelendirmeli seçim metodu kullanılarak çözümler uyum fonksiyon değerlerine göre sıralanmış bu sıralamaya göre orantılı olarak seçilme şansına sahip olmuşlardır. Çaprazlama operatörü uygulanırken rasgele eşleştirilen bireylerin oluşturduğu kromozom dizisi rasgele seçilen iki noktadan parçalara ayrılıp bu noktaların arasında kalan kısımlar çaprazlama oranına bağlı olarak yer değiştirilmiştir (iki noktalı çaprazlama).

Algoritmanın ilerleyen aşamalarda çözümleri gittikçe iyileştireceği ve iyileşen bu çözüm adaylarının daha az düzeltilmeye ihtiyaç duyacağı düşüncesinden hareketle, gerçekleştirilen modelde mutasyon oranının uyarlanır olarak belirlenmesi yoluna gidilmiştir. Bu amaçla, araştırmanın başlangıcında mutasyon oranı için minimum ve maksimum değer aralığı seçilmiş, araştırmanın maksimum mutasyon oranı ile başlamasının ardından ilerleyen aşamalarında mutasyon oranı minimum değere doğru düzenli olarak azaltılmıştır. Minimum ve maksimum mutasyon oranları belirlenirken literatüre uygun olarak 0.01 ile 0.1 aralığı dikkate alınmıştır. Her bir evrimde mutasyona uğrayacak birey sayısı çaprazlama sayısına benzer şekilde başlangıç popülasyon büyüklüğüne bağlı olarak ayarlanırken mutasyon yapılacak bireyler yine derecelendirmeli seçim metodu kullanılarak seçilmiştir. Çaprazlama ve mutasyon yapılacak bireylerin seçiminde derecelendirmeli seçim metodu kullanılmasıyla etkin çözümlerin önemi arttırılmaya çalışılmıştır. Uyarlanır mutasyon oranı Şekil 3.10'da gösterilmiştir.



Şekil 3.10. Uyarlanır mutasyon oranı

GA'nın her bir evriminde yeni popülasyon oluşturulurken popülasyon büyüklüğüne bağlı olarak dışardan rasgele çözümlerde popülasyona dahil edilerek araştırma uzayının daha etkili bir şekilde taranması sağlanmaya çalışılmıştır.

Paralel çalışan benzetilmiş tavlama algoritmaları için komşu çözümlerin üretilmesini sağlayan komşu üretme mekanizması olarak sıcaklığa göre değişen rasgele seçim işlemi kullanılmıştır. Algoritmanın iterasyon sayısı probleme göre farklılık gösterirken, soğutma tarifi olarak basamak tipi sıcaklık düşürme stratejisi tercih edilmiştir. Paralel olarak çalışan bütün yerel araştırma algoritmalarının sıcaklık parametresi T 'nin başlangıç değeri, sıcaklık azaltma oranı ve güncelleme kuralı tanımlanan aralıklarda rasgele sayı üretici tarafından belirlenmiştir. Aralıklar tanımlanırken literatüre uygun olarak başlangıç sıcaklık değeri 450–550, sıcaklık düşürme katsayısı 0.7–0.9 aralığında dikkate alınmıştır.

4. BÖLÜM

MEMETİK ALGORİTMA KULLANARAK SÜREKLİ TEST FONKSİYONLARIN OPTİMİZASYONU

4.1. Giriş

Memetik algoritma 3. Bölümde açıklandığı gibi bölgesel ve küresel arama için ayrı ayrı metotlar kullanan ve paralel çalışan algoritmalarındandır. Algoritma farklı algoritmaların avantajlarını bir arada kullandığından, genel olarak sadece bölgesel veya sadece küresel araştırma yapan algoritmalara nazaran daha başarılı sonuçlar sağlaması beklenmektedir.

Bu bölümde, gerçekleştirilen memetik algoritma modelinin sürekli test fonksiyonlarının optimizasyonundaki performansı incelenerek, elde edilen neticeler literatürde yer alan bazı algoritmalara ait sonuçlarla karşılaştırılmıştır.

4.2. Memetik Algoritma ile Sürekli Fonksiyonların Optimizasyonu

Önerilen memetik algoritmasının performansının test edilmesi amacıyla literatürde yaygın olarak kullanılan yedi adet sürekli test fonksiyonu dikkate alınmıştır. Tablo 4.1’de verilen bu fonksiyonlardan her biri farklı bir karmaşıklık derecesini yansıtmaktadır.

Sphere (F1), tek modlu, dışbükey ve simetrik bir fonksiyondur. Rosenbrock (F2), zor olarak bilinen parabolik yapıda içbükey bir fonksiyondur. Step (F3) fonksiyonu genellikle araştırma yönü hakkında ipucu vermeyen düz yüzeylerde algoritmaların çözüm geliştirmesini ölçmek için kullanılır.

Foxholes (F4), çok sayıda bölgesel minimum içermektedir ve bu fonksiyon ile algoritmaların bölgesel minimumdan kurtulabilmeleri test edilmektedir. F5 fonksiyonu;

[-10,10] aralığında 40.000 bölgesel minimum içermektedir. Griewank (F6), doğrusal olmayan çok modlu bir fonksiyondur. Rastrigin (F7) ise geniş araştırma uzayı bulunan çok sayıda bölgesel minimum nedeniyle oldukça zor bir problemdir. Test fonksiyonlarının parametre sayısı, optimum çözüm değerleri, parametre araştırma uzayı ve çözümü temsil etmekte kullanılan toplam bit sayıları Tablo 4.2’de verilmiştir.

Tablo 4.1. Benzetimde kullanılan nümerik test fonksiyonları

Gösterim	İsim	Fonksiyonlar
F1	Sphere	$f_1 = \sum_{i=1}^3 x_i^2$
F2	Rosenbrock	$f_2 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$
F3	Step	$f_3 = \sum_{i=1}^5 [x_i]$ [x _i], x _i 'ye eşit veya daha küçük en büyük tamsayıyı gösterir.
F4	Foxholes	$f_4 = [0.002 + \sum_{j=1}^{25} (j + \sum_{i=1}^2 (x_i - a_{ij})^6)^{-1}]^{-1}$ $\{a_{1j}, a_{2j}\}_{j=1}^{25} = \{(-32,-32), (-16,-32), (0,-32), (16,-32), (32,-32), (-32,-16), (-16,-16), (0,-16), (16,-16), (32,-16), \dots, (-32,32), (-16,32), (0,32), (16,32), (32,32)\}$
F5		$f_5 = (x_1^2 + x_2^2) / 2 - \cos(20\pi x_1) \cos(20\pi x_2) + 2$
F6	Griewank	$f_6 = 1 + \sum_{i=1}^{10} \left(\frac{x_i^2}{4000} \right) - \prod_{i=1}^{10} \left(\cos \left(\frac{x_i}{\sqrt{i}} \right) \right)$
F7	Rastrigin	$f_7 = 20A + \sum_{i=1}^{20} (x_i^2 - 10 \cos(2\pi x_i))$, $A = 10$

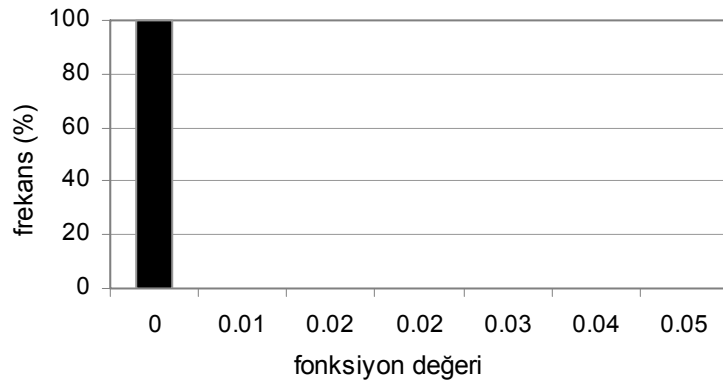
Tablo 4.2. Test fonksiyonlarının parametre sayısı, optimum çözüm değerleri, parametre araştırma uzayı ve çözümü temsil etmekte kullanılan toplam bit sayıları

Fonksiyon	Parametre sayısı	Çözümler		Parametre Sınırları		Çözümün bit sayısı
		x_i	$f(x)$	Enküçük	Enbüyük	
F1	4	0.0	0.0	-5.12	5.12	40
F2	2	1.0	0.0	-2.048	2.048	32
F3	5	-5.12	-30.0	-5.12	5.12	50
F4	2	-32.0	1.0	-65536	65536	40
F5	2	0.0	1.0	-10	10	36
F6	10	0.0	0.0	-600	600	200
F7	20	0.0	0.0	-5.12	5.12	400

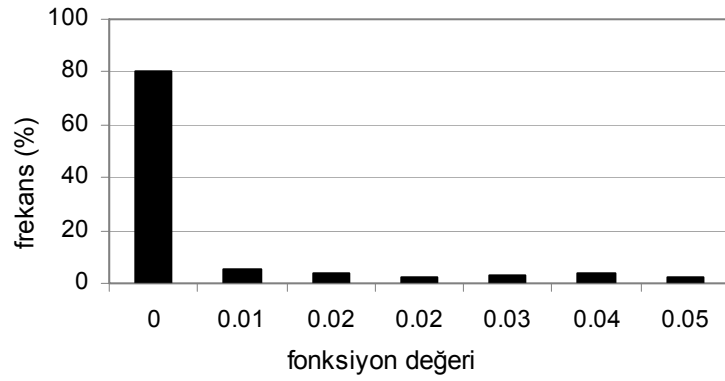
4.3. Benzetim Sonuçları

Benzetim çalışmalarında, önerilen memetik algoritma modelinde paralel olarak çalışan benzetilmiş tavlama algoritması sayısı ilk beş fonksiyon için 10, F6 ve F7 fonksiyonları için 20 alınmıştır. Her bir kademede benzetilmiş tavlama algoritması için iterasyon sayısı ilk beş fonksiyon için 100, F6 ve F7 için 250 olarak belirlenmiştir. Her bir katmandaki genetik algoritma için değerlendirme sayısı 50 olarak belirlenmiş, çaprazlama oranı sabit tutulurken, mutasyon oranı her bir kademede uyarlanır olarak değiştirilmiştir. Uyarlanır mutasyon oranı uygulamasında, mutasyon oranı belirli bir alt ve üst sınır arasında belirlenmiştir. Bu sınır gerçekleştirilen birçok denemenin ardından 0.01–0.1 olarak belirlenmiştir. Önerilen algoritmanın sürekli test fonksiyonlarındaki performansı tabu araştırma (TA), paralel tabu araştırma (Paralel TA), turlayan karınca kolonisi optimizasyonu (TKKO), paralel karınca koloni optimizasyonu (Paralel KKO) ve genetik algoritma ile elde edilen sonuçlarla karşılaştırılmıştır [18,55]. Karşılaştırma amacıyla değerlendirme sayısı ilk beş fonksiyon için 16800, son iki fonksiyon için 50000 olarak belirlenmiş ve algoritmanın çalışması belirtilen değerlendirme sayısı sonunda sonlandırılmıştır. Algoritmanın diğer parametre değerleri, mutasyona uğrayacak birey sayısı 3, çaprazlama oranı 0.9, çaprazlama yapılacak birey sayısı 3, bir sonraki nesle aktarılacak elit birey sayısı 2, her bir evrimde araştırmaya dışardan katılacak birey sayısı 1 olarak seçilmiştir. Her bir test fonksiyonu için farklı başlangıç

çözümleri ile 20'şer deneme gerçekleştirilmiştir. Önerilen modelin performansını incelemek için gerçekleştirilen denemelerden elde edilen sonuçlara ait histogramlar Şekil 4.1-4.8'de TA, Paralel TA, GA, TKKO ve Paralel KKO algoritmaları ile elde edilen sonuçlarla karşılaştırmalı olarak verilmiştir.

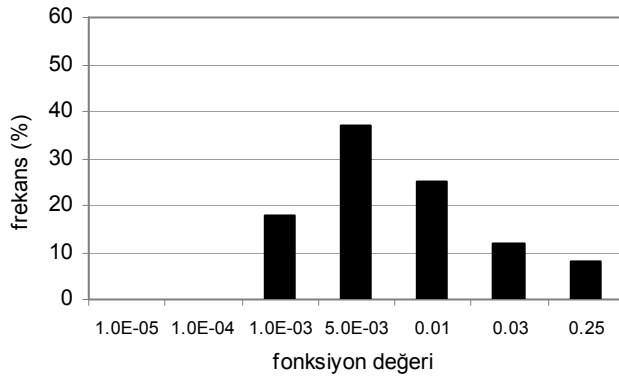


(a)

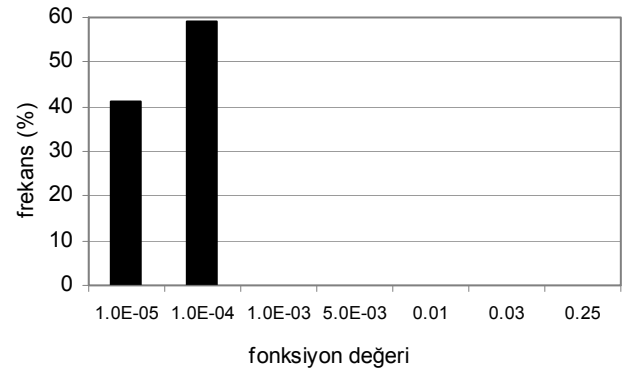


(b)

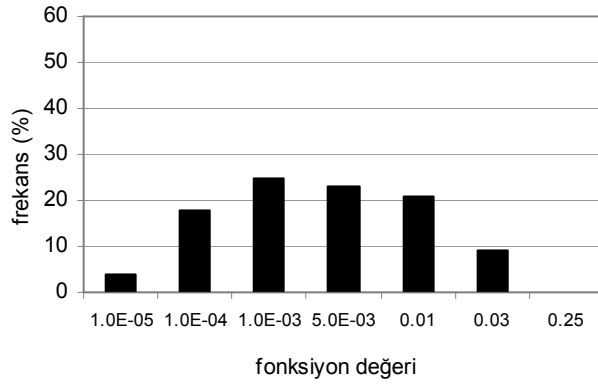
Şekil 4.1. F1 fonksiyonu için (a) TA, Paralel TA, TKKO, Paralel KKO, MA ve (b) GA ile elde edilen sonuçların dağılımı



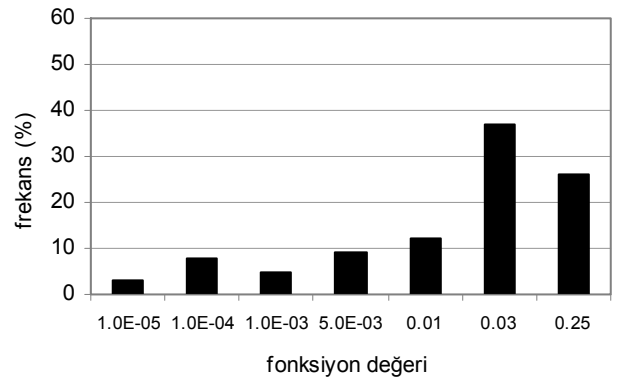
(a) TA



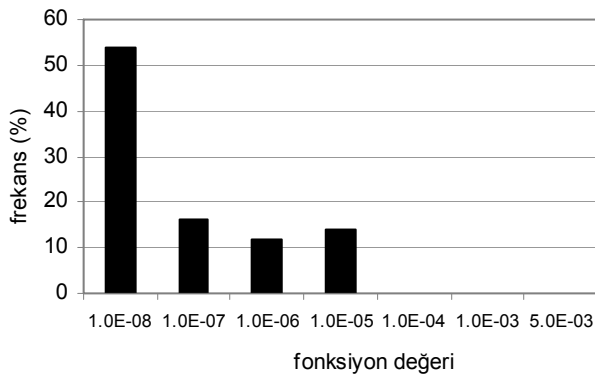
(b) Paralel TA



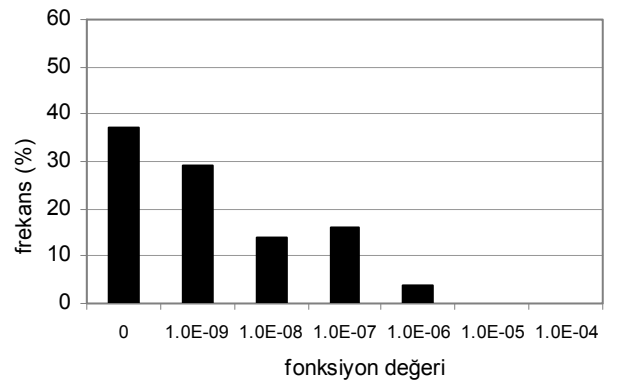
(c) GA



(d) TKKO

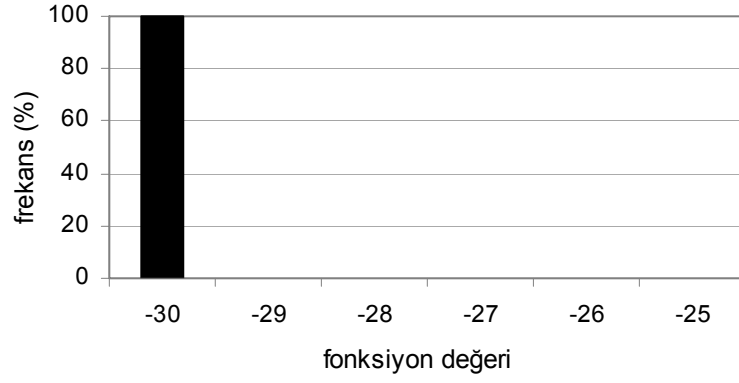


(e) Paralel KKO

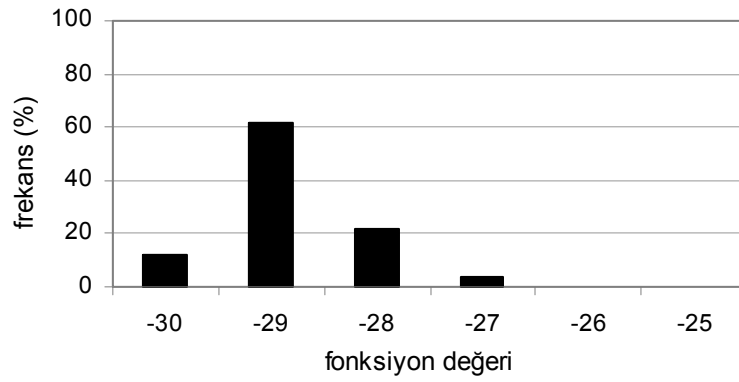


(f) MA

Şekil 4.2. F2 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı

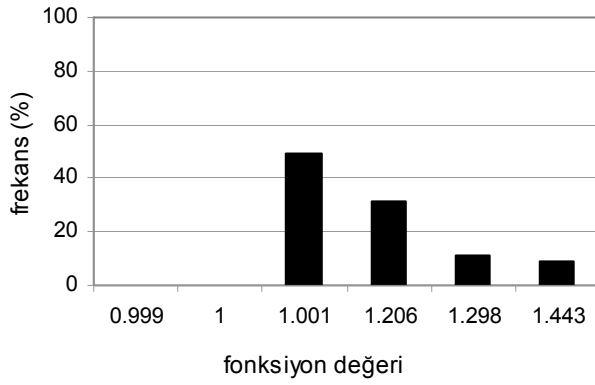


(a) TA, Paralel TA, GA, Paralel KKO ve MA

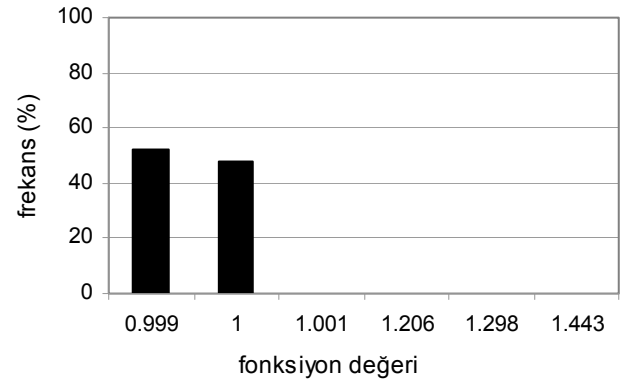


(b) TKKO

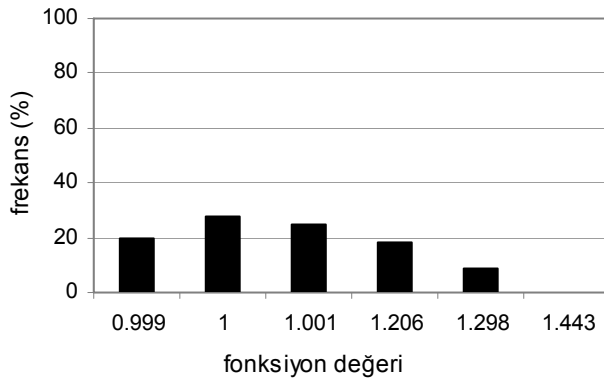
Şekil 4.3. F3 fonksiyonu için (a) TA, Paralel TA, GA, Paralel KKO ve MA, (b) TKKO ile elde edilen sonuçların dağılımı



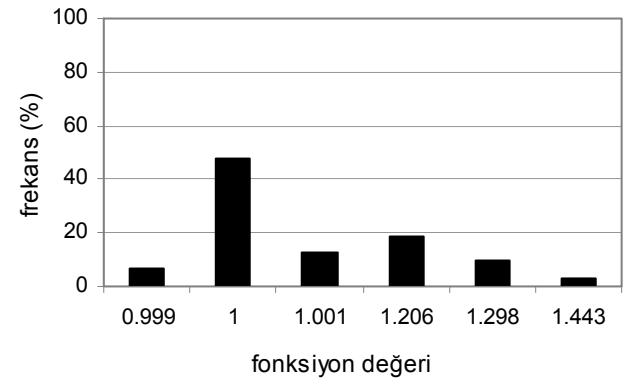
(a) TA



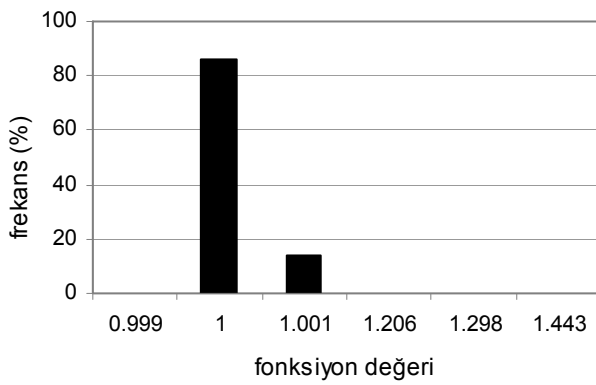
(b) Paralel TA



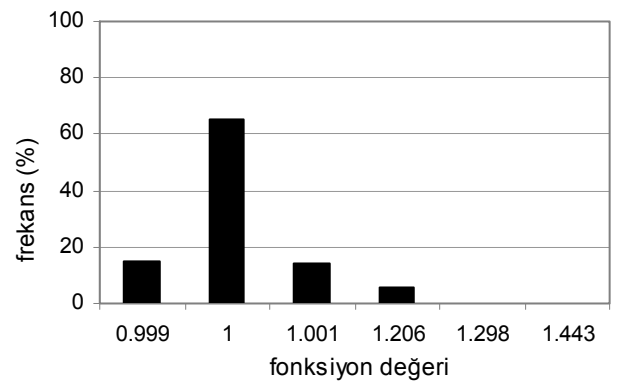
(c) GA



(d) TKKO

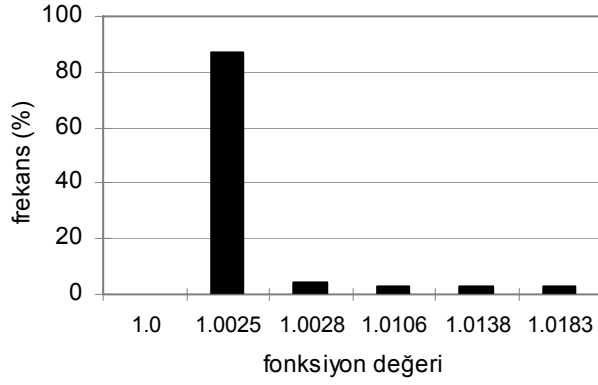


(e) Paralel KKO

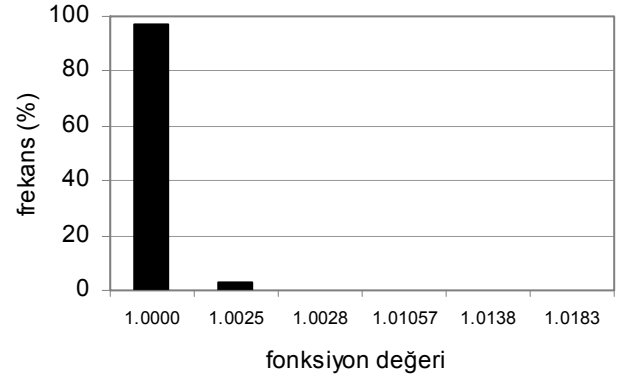


(f) MA

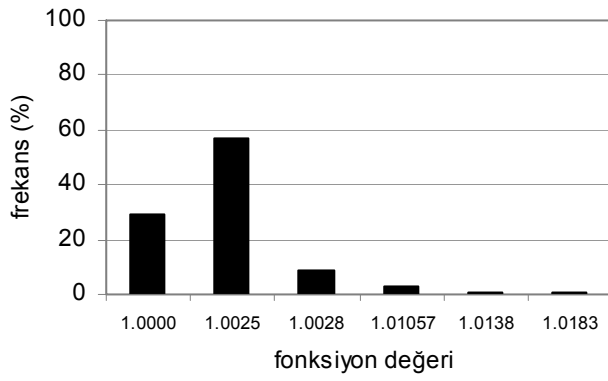
Şekil 4.4. F4 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı



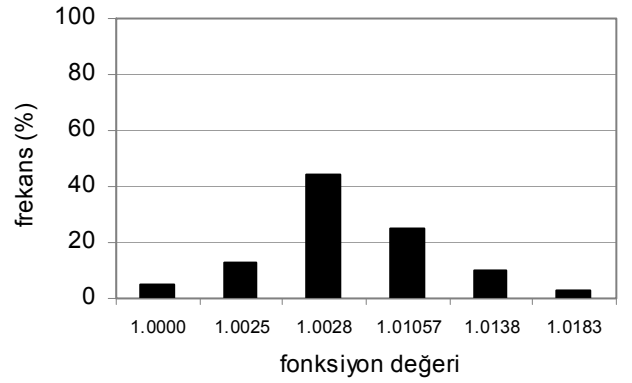
(a) TA



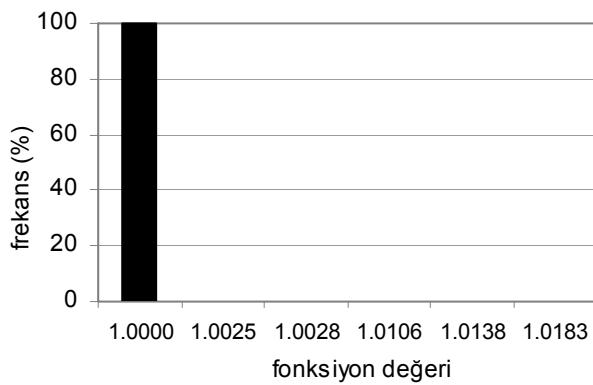
(b) Paralel TA



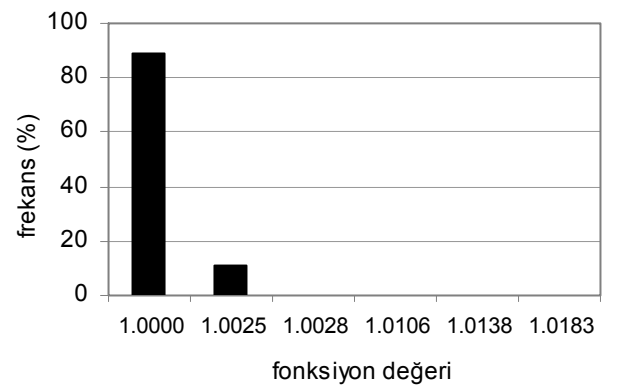
(c) GA



(d) TKKO

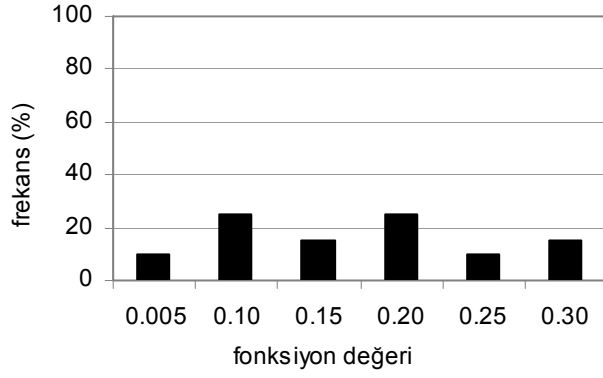


(e) Paralel KKO

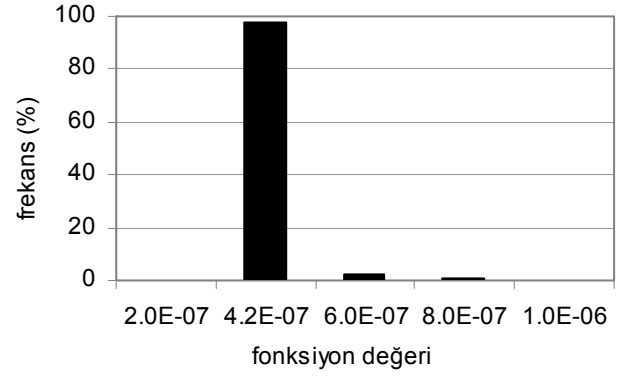


(f) MA

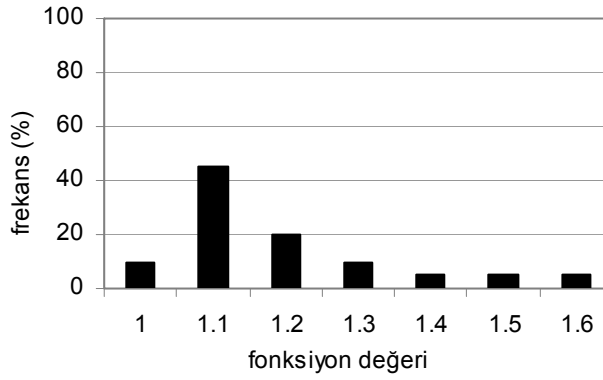
Şekil 4.5. F5 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı



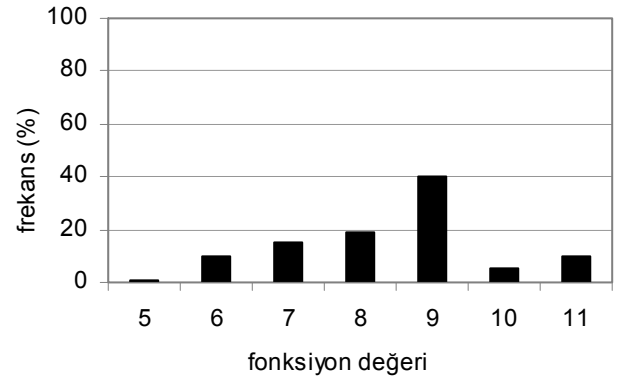
(a) TA



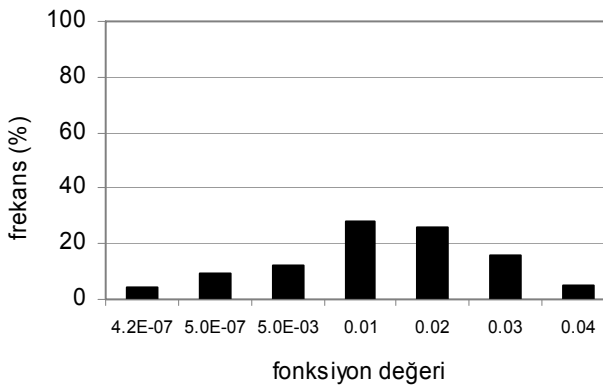
(b) Paralel TA



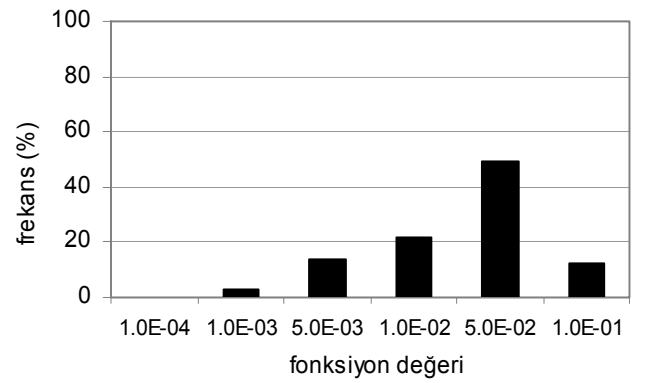
(c) GA



(d) TKKO

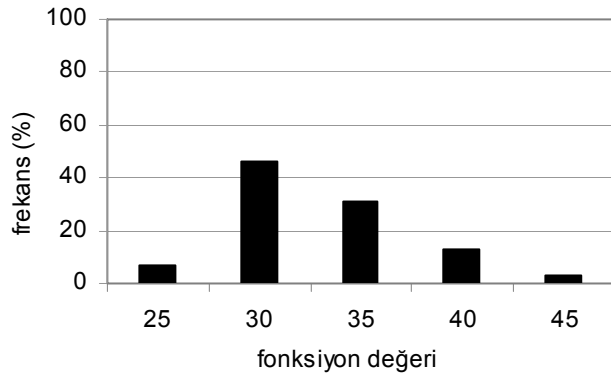


(e) Paralel KKO

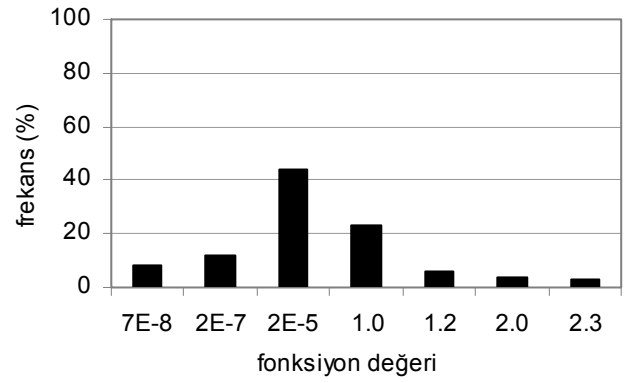


(f) MA

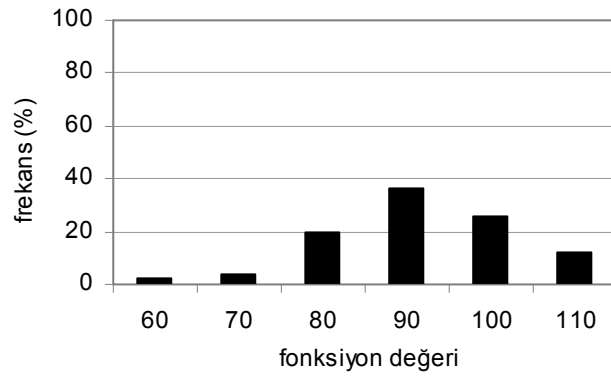
Şekil 4.6. F6 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı



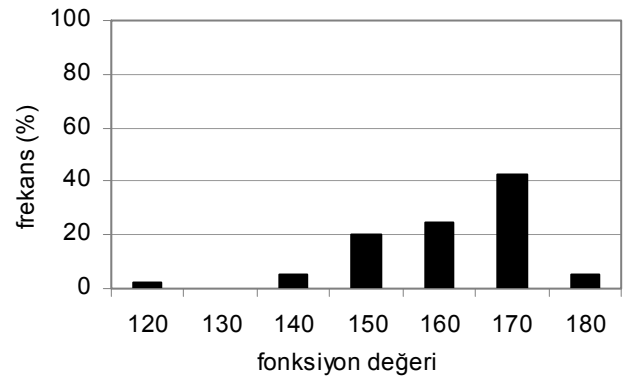
(a) TA



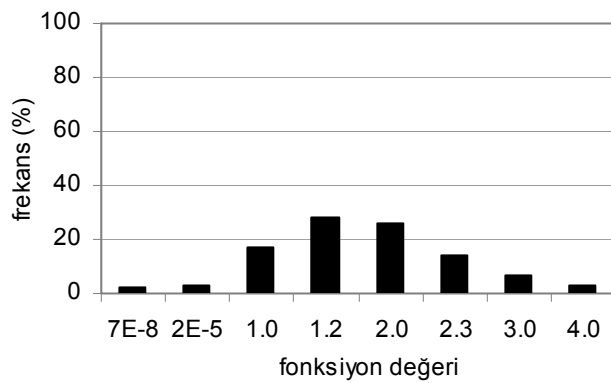
(b) Paralel TA



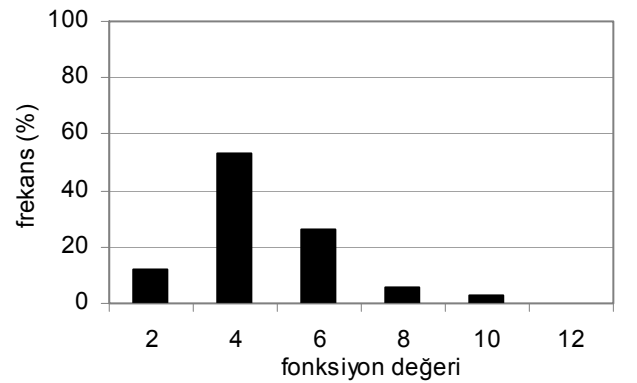
(c) GA



(d) TKKO



(e) Paralel KKO



(f) MA

Şekil 4.7. F7 fonksiyonu için (a) TA, (b) Paralel TA, (c) GA, (d) TKKO, (e) Paralel KKO ve (f) MA ile elde edilen sonuçların dağılımı

F1 fonksiyonu için yapılan denemelere ait sonuçlar Şekil 4.1’de gösterilmiştir. Tek modlu ve nispeten kolay olan bu fonksiyon için GA hariç tüm algoritmalar yapılan denemelerin tamamında optimum çözümü bulabilmiştir.

Literatürde zor bir fonksiyon olarak bilinen F2 fonksiyonu için yapılan denemelerden elde edilen sonuçlar Şekil 4.2’de verilmiştir. Bu sonuçlar incelendiğinde memetik algoritma ile elde edilen sonuçların diğer beş algoritmaya göre çok daha başarılı olduğu görülmektedir.

Araştırma yönü hakkında ipucu vermeyen F3 fonksiyonu için yapılan denemelerden elde edilen sonuçlar Şekil 4.3’de gösterilmiştir. Bu fonksiyon için TKKO algoritması dışındaki diğer tüm algoritmalar yapılan denemelerin tamamında optimum çözümü bulabilmiştir.

Çok sayıda bölgesel minimum içeren F4 fonksiyonu için yapılan denemelerden elde edilen sonuçlar Şekil 4.4’de verilmiştir. Bu fonksiyon için yapılan denemelerde en yüksek başarımlar Paralel KKO ve Paralel TA algoritmaları ile sağlanmıştır. Her iki algortmada yapılan denemelerin büyük çoğunluğunda optimum çözümü sağlamıştır. Memetik algoritma ile gerçekleştirilen denemelerde %65’in üzerinde bir oranla optimum çözüme ulaşılmıştır. Memetik algoritma ile sağlanan çözümlerin başarımları Paralel KKO ve Paralel TA algoritmaları ile sağlanan çözümlere nazaran biraz daha düşük olmakla beraber, diğer üç algoritma ile sağlanan çözümlere nazaran açıkça daha yüksektir.

Çok sayıda bölgesel minimum içeren bir başka fonksiyon olan F5 için yapılan denemelerin sonuçları Şekil 4.5’de verilmiştir. Bu fonksiyon için Paralel KKO, Paralel TA ve memetik algoritma benzer performans sergilerken, diğer üç algoritmanın performansı bu algoritmalara göre daha düşük kalmıştır.

Doğrusal olmayan, çok modlu ve ilk beş fonksiyona göre daha zor olan 10 parametrelili ve 20 parametrelili F6 ve F7 fonksiyonları için yapılan denemelerden elde edilen sonuçlar Şekil 4.6 ve Şekil 4.7’de verilmiştir. Bu fonksiyonlar için en yüksek başarımlar Paralel TA, Paralel KKO algoritmaları ile sağlanmıştır. Bu fonksiyonlar için memetik algoritma ile elde edilen çözümlerin kaliteleri bu iki algortmaya nazaran biraz daha düşük olmakla beraber, diğer üç algortmaya göre oldukça yüksektir.

Histogramlar incelendiğinde gerçekleştirilen memetik algoritma modeli F1 ve F3 fonksiyonları için yapılan tüm denemelerde optimum çözümü bulabilmiştir. Gerçekleştirilen memetik algoritma modeli F2 fonksiyonu için diğer beş algoritmadan daha iyi performans sergilerken F4 ve F5 fonksiyonları için Paralel TA ve Paralel KKO algoritmalarıyla benzer performans sergilemiştir. Ancak, F6 ve F7 fonksiyonları için Paralel TA ve Paralel KKO algoritması ile elde edilen çözümlerin diğer algoritmalara göre daha iyi olduğu görülmektedir.

Gerçekleştirilen modelin bu fonksiyonlar üzerindeki performansı Paralel TA ve Paralel KKO algoritması dışındaki diğer algoritmalara göre çok daha yüksektir. Özellikle GA ve karınca tabanlı olan TKKO algoritmasına ait sonuçlarla karşılaştırıldığında önerilen algoritmanın performansının açık bir şekilde yüksek olduğu görülmektedir.

5. BÖLÜM

KONTROL SİSTEMLERİ

5.1. Giriş

Kontrol sistemi belirli bir amaca yönelik olarak kendisini veya diğer bir sistemi kumanda etmek, yönlendirmek veya ayarlamak üzere birleştirilen fiziksel organlar kümesidir [2].

Kontrol uygulamaları ilk olarak şamandıra düzenleyici mekanizmaları geliştirme aşamasında, M.Ö. 300 yıllarında Yunanlılar tarafından kullanılmaya başlanmıştır. Ktesibious'un su saati bu mekanizmayı kullanmaktadır. İskenderiyeli Heron tapınak kapılarının otomatik açılması (M.S. 1.yy), Watt'ın buhar motorlarının hızını kontrol etmek amacı ile geliştirdiği *flyball governor* düzeneği (1820) ve 20.yy ortalarında uçak kontrol sistemleri, doğadaki canlıları taklit eden robotlar gibi kontrol uygulamaları gelişmiştir.

Kontrol sistemleri uygulamaları genelde elektrik, mekanik ve kimyasal bölümlerden oluşmaktadır; fakat politik, sosyal ve iş hayatının dinamiklerinin anlaşılabilirliğinin artması ile birlikte kontrol mühendisliğinin sınırları oldukça genişlemiştir. Modern kontrol sistemlerinde, üretim sürecini geliştirmek, enerjinin mümkün olan en verimli şekilde kullanılmasını sağlamak ve tüm mühendislik alanlarına yönelik kontrol tasarım stratejileri geliştirmek amaçlanmaktadır.

5.2. Kontrol Sistemleri ve Türleri

Amaca bağlı olarak kontrol sistemleri çok farklı şekillerde sınıflandırılabilir. Örneğin analiz ve tasarım yöntemleri açısından kontrol sistemleri doğrusal ve doğrusal olmayan, zamanla değişen veya zamanla değişmeyen sistemler olarak sınıflandırılabilir. Tüm

fiziksel sistemler belirli bir ölçünün ötesinde doğrusal olmadığından uygulamada neredeyse doğrusal sistem yoktur. Doğrusal kontrol sistemleri, sadece analiz ve tasarım basitliği nedeniyle analizciler tarafından yaratılmış ideal bir modeldir. Bir kontrol sistemindeki işaretlerin genliği, sistem elemanlarının doğrusal davranış göstereceği, belirli sınırlar içerisinde tutulursa sistem ana hatlarıyla doğrusaldır. Ancak işaretlerin genlikleri doğrusal çalışma bölgesinin dışına taşarsa, doğrusal olmamanın derecesine bağlı olarak sistem artık doğrusal kabul edilemez. Bir kontrol sisteminin parametreleri, sistem çalışırken zamandan bağımsız olarak değişmez ise, o sisteme zamanla değişmeyen sistem denir. Uygulamada fiziksel sistemlerin çoğunda zamanla değeri kayan ya da değişen elemanlar bulunur.

Sistemler kullanılan işaret türleri yönünden sürekli verili ve ayrık verili sistemler olarak ayrılabilir. Bir sürekli verili sistem, sistemin farklı kısımlarındaki işaretlerin tümü sürekli t zaman değişkeninin fonksiyonu olan bir sistemdir. Ayrık verili kontrol sistemleri sürekli verili kontrol sistemlerinden sistemin bir veya daha çok noktasından işaretlerin bir darbe dizisi ya da sayısal kodlanmış biçimde olmasıyla ayrılır.

Sistemlerin bir başka sınıflandırılma şeklide doğal kontrol sistemleri ve endüstride kullanılan insan yapısı kontrol sistemlerdir. Örneğin insan vücudu ile ilgili fizyolojik işlemler, vücut sıcaklığının denetimi, kan şekerinin ayarlanması, gözün ışığın şiddetine göre kısılması gibi işlemler doğal kontrol sistemleri olarak bilinirken, bir sıcaklık kontrol sistemi ayarlandıktan sonra bizim müdahalemiz olmaksızın ortamın sıcaklığının otomatik olarak ayarlanarak belli bir değerde tutulması endüstriyel bir kontrol sistemidir.

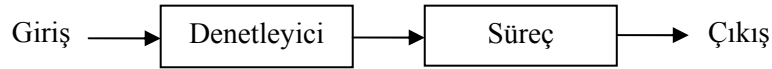
Endüstriyel anlamda bir kontrol sistemi; (i) kontrol edilen sistem ve (ii) kontrol elemanları olmak üzere iki ana kısımdan oluşur. Kontrol edilen sistem; özel bir niceliğin denetlendiği tesisat, süreç veya makinedir. Kontrol edilen sisteme uygulanacak uygun bir denetim sinyali sağlayan eleman ise kontrol elemanıdır. Sistem analizlerinin temelinde, sistem içindeki bileşenlerin sebep-sonuç ilişkilerinin belirlenmesinde doğrusal sistem teorisi sıkça kullanılmaktadır. Şekil 5.1’de herhangi bir kontrol edilen sürecin blok yapısı gösterilmiştir. Buradaki giriş çıkış ilişkileri, sürecin sebep-sonuç ilişkilerini temsil etmektedir.



Şekil 5.1. Sistem blok şeması

Kontrol sistemleri denetim etkisi açısından açık çevrimli kontrol sistemleri (geribeslemesiz sistemler) ve kapalı çevrimli kontrol sistemleri (geribeslemeli sistemler) olmak üzere iki ana sınıfa ayrılır.

Bir açık çevrimli kontrol sistemi, istenilen bir sistem tepkisini elde etmek amacı ile bir kontrol edici kullanan geribesleme içermeyen sistemdir. Açık çevrimli denetim sistemlerinde denetim eylemi sistem çıkışından bağımsızdır. Uygulamada açık çevrimli denetim sistemleri giriş çıkış bağıntıları önceden belli olan ve iç veya dış bozuculara maruz kalmayan sistemlerde kullanılır. Açık çevrimli denetleyici sistem blok şeması Şekil 5.2’de gösterilmiştir.



Şekil 5.2. Açık çevrimli sistem blok şeması

Kapalı çevrimli sistemler istenilen çıkış ile sistemin gözlemlenen çıkışı arasında kıyaslama yapabilmeye imkan tanıyacak ek ölçüm imkanları sunmaktadır. Ölçülen çıkış değeri geribesleme sinyali olarak adlandırılmaktadır. Geribeslemeli kontrol sistemi, tanımlanmış bir sistem değişkeninin bir başkası ile karşılaştırılması ve bunların farkının değerlendirilerek kontrol edilmesine dayanmaktadır. Kapalı çevrimli denetleyici sistem blok şeması Şekil 5.3’de gösterilmiştir.



Şekil 5.3. Kapalı çevrimli sistem blok şeması

Geribeslemeli kontrol sistemi genelde, süreci kontrol edebilmek için, çıkış ile daha önceden belirlenmiş olan sistemin referans girişi arasında ilişkilendirilmiş fonksiyon kullanılır. Kontrol edilecek süreç içinde genelde referans girişi ile sistemin çıkışı arasındaki fark yüksektir ve bu fark sürekli bir şekilde kontrol süreci içinde azaltılmak istenir. Açık çevrimli sistem ile kapalı çevrimli sistemi birbirinden ayırt eden en önemli unsur geribesleme etkisidir. Geribesleme etkisi negatif ve pozitif geribesleme olmak üzere iki şekilde olur. Negatif geribesleme de çıkıştaki değişimler girişe ters yönde etki ederken, pozitif geribesleme de çıkış girişi aynı yönde etki eder. Negatif geribesleme endüstriyel süreçlerin en önemli özelliğidir ve daima hatayı en küçük değerde tutmaya veya sıfır yapmaya çalışır. Pozitif geribesleme ise iç döngüler hariç bir kapalı çevrimli denetim sisteminde kullanılmaz. Kontrol sistemlerinde geribesleme kullanmanın anlamı sadece sistem çıkışı ve referans girişi arasındaki hataları azaltmak değildir. Geribesleme ilerleyen bölümlerde deyinilecek kararlılık, band genişliği, toplam kazanç, bozucu ve duyarlılık gibi sistem davranış karakteristikleri üzerinde de etkilidir.

Açık çevrimli sistem için; bir elektrikli tost makinesi, tam otomatik çamaşır makinesi, trafik ışıkları, kapalı çevrimli sistem için ise yolda ilerleyen bir arabayı değişimleri de göz önünde bulundurarak gerekli ayarlamaları yapan bir insanın kullanması örnek olarak verilebilir.

Genel olarak kontrol sistemlerini özel niteliklerine göre sınıflandırmanın birçok farklı yolu vardır. Önemli olan yaygın sınıflandırma yöntemlerinin bilinmesi ve kontrol sistemlerinin analiz ve tasarıma başlamadan önce uygun yaklaşımın belirlenmiş olmasıdır.

5.3. Kontrol Sistemlerinin Tasarım İlkeleri

Bir kontrol sisteminden istenen denetim büyüklüğünü girişteki arzu edilen değere uygun tutmaktır. Giriş değerinden uzaklaşan denetim büyüklüğünün yeniden giriş değerine erişmesi sağlanır. Doğrusal kontrol sistemlerinin tasarımı zaman ya da frekans tanım bölgesinde gerçekleştirilebilir. Kontrol sistemlerinde zaman genellikle bağımsız bir değişken görevini üstlenir ve bu nedenle durum ve çıkış büyüklüklerinin zamana göre değişimi ya da zaman yanıtları genel ilgi alanını oluşturur. Analiz problemlerinde sistemlere birim basamak, rampa ya da parabolik gibi referans giriş işaretleri uygulanır

ve bu işaretlere verilen yanıtlar incelenerek sistemlerin davranışı değerlendirilmeye çalışılır. Bir kontrol sisteminde çıkış işaretinin giriş işaretini belirli koşullar altında takip etmesi isteniyor ise, giriş ve çıkış işareti zaman fonksiyonu olarak karşılaştırılır. Uygulamada bir kontrol sisteminin davranışı, en gerçekçi ve doğru olarak ancak zaman tanım bölgesi kriterleri ile belirlenir [56]. Haberleşme sistemlerinde işlenen işaretlerin sinüzoidal olması ya da sinüzoidal birleşenlerden oluşması nedeniyle sistemlerin analiz ve tasarımında frekans cevabı çok daha fazla önem taşır. Özellikle yüksek mertebeden kontrol sistemlerinde zaman tanım bölgesi yanıtlarına ilişkin analitik ifadeler çok zor elde edilirken, frekans tanım bölgesinde düşük mertebeden sistemlerle sınırlı kalmayan, çok sayıda grafiksel yöntem mevcuttur. Frekans tanım bölgesi, sistem gürültü ve parametre değişimlerine karşı duyarlılık ölçümleri için daha uygundur. Ayrıca frekans tanım bölgesi bize sistemin zaman tanım bölgesi davranışını da tasarlamaya olanak kılar.

Bir kontrol sisteminin zaman cevabı genellikle “geçici durum” ve “kalıcı durum” olmak üzere iki tür davranış gösterir. Geçici durum davranışı, sistemin belli bir dış uyarı karşısında bir başlangıç değerinden bir nihai duruma kadar zaman değişimine bağlı olarak gösterdiği davranıştır. Kalıcı durum davranışı ise sistemin geçici durum davranışı tamamlandıktan sonra zaman sonsuza uzanıyorken koruduğu davranıştır. Bir sistemin kalıcı durum haline en küçük hata veya sıfır hata ile ulaşması istenir. Denetim sistemlerinde geribesleme kullanılmak suretiyle kalıcı durum hatası en küçük değere veya sıfıra düşürülebilir.

Kontrol sistemlerinin çözümlerinde geçici durum ve kalıcı durum çalışmaları ile ilgili olarak sistemin aşağıdaki özellikleri sağlayıp sağlamadığı araştırılır.

Kararlı çalışma: Bir sistemin kararlılığı denilince sistemin sınırlı bir giriş değerine karşılık sınırlı bir çıkış değeri vermesi anlaşılmalıdır. Bir kontrol sisteminin en önemli özelliği kararlı olmasıdır. Kararlı bir çalışma sağlamayan kontrol sistemi hiçbir işe yaramaz. Bir sistemin sadece kararlı olup olmadığını bilmek çoğu durumda yeterli olmayıp genellikle sistemin kararsız olmaya ne kadar yakın olduğunun da bilinmesi gerekir. Yani sistemin *bağıl kararlılığını* bulmak gerekir. Bir kontrol sisteminin kararlılığı sistemin kendi yapısına bağlı olup, başvuru girişinden ve bozucu girişlerden bağımsızdır.

Geçici durum çalışması halinde hızlı cevap: Bir kontrol sisteminin geçici durum çalışmasında denetim büyüklüğünün kalıcı durum değerine mümkün olan en kısa zamanda erişmesi, diğer bir deyişle uyarılara hızlı cevap vermesi istenir. Cevap hızı sistemin kazancı ile artar.

Kalıcı durum davranışı: Bir kontrol sisteminin kalıcı durum çalışmasında hataların sıfır veya kabul edilebilir değerlerde tutulması istenir. Bu durum kontrol sisteminin duyarlılığı ile ilgili olup sistemde hata sıfır ise ve kalıcı durum çalışmasında sıfır hata korunabiliyorsa “duyarlılık” yüksektir denir. Ancak hata değerini sıfırda tutmak pek kolay ve ekonomik bir iş değildir, bunun yerine çoğunlukla hata artı eksi bir tolerans sınırı içinde tutulmaya çalışılır. Bu tolerans sınırı ne kadar dar ise duyarlılık o kadar yüksek olur. Duyarlılık sistem kazancı ile artar.

İdeal olarak bir kontrol sisteminin çok kararlı, çok hızlı ve çok duyarlı olması istenir. Diğer taraftan çok hızlı ve duyarlı sistemler kararsız çalışmaya eğilimlidirler. Şöyle ki kalıcı durum hatasını en aza indirmek ve sistemin cevap hızını artırmak için sistemin duyarlılığını artırmak gerekir ki buda diğer taraftan sistemin kararlılığını azaltmaktadır. Kontrol sistem tasarımında bu iki temel özellik arasında uygun bir uzlaşma sağlanmaya çalışılır.

5.4. Geçici Durum Parametreleri

Şekil 5.4’de bir sistemin geçici durum zaman alanı cevabı ve ilgili parametreler gösterilmiştir. Aşağıda kısaca tanımlanan bu parametreler, tasarlanacak herhangi bir denetleyicinin başarımını değerlendirmede büyük öneme sahiptir.

Gecikme zamanı, t_g : Gecikme zamanı, cevabın nihai değerinin yarısına ilk defa ulaşması için geçen zamandır.

Yükselme zamanı, t_y : Yükselme zamanı, cevabın nihai değerinin %10’dan %90’ına, %5 den %95 ine veya %0 dan %100 üne kadar geçen zamandır. Genellikle aşırı sönümlü birinci dereceden sistemler için %0-100 yükselme zamanı kullanılırken titreşimli sönümlü sistemlerde %10-90 yükselme zamanı kullanılmaktadır.

modelin elde edilebilmesi için sistem değişkenleri arasındaki ilişkinin analiz edilmesi ve sistem modelinin çıkarılması gerekir. Bir sisteme ait matematiksel model doğrusal zamanla değişmiyor ise klasik kontrol teorisinde analiz ve tasarım için laplace dönüşümü, fourier dönüşümü ve z dönüşümü yaygın olarak kullanılır. Sırası ile s bölgesi, ω bölgesi ve z bölgesi yöntemleri adı altında klasik kontrol teorisinde köklerin yer eğrisi, blok diyagramları, işaret akış diyagramları ve bode diyagramları sistem tasarımı için kullanılır. Sistemlerin kararlılık analizi için genellikle Routh-Hurwitz ve Nyquist yöntemleri kullanılır. Eğer sistem doğrusal olmayan (non-linear) ise belli çalışma noktaları etrafında doğrusallaştırma yapılarak yine klasik kontrol teorisinde kullanılan yöntemlerle analiz ve tasarım gerçekleştirilir.

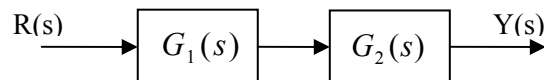
Laplace dönüşümleri kullanılarak sinüzoidal fonksiyonlar, sönümlü sinüzoidal fonksiyonlar ve üstel fonksiyonlar gibi pek çok fonksiyon karmaşık değişkenli cebrik fonksiyonlar haline dönüştürülür. Dönüşümleri yapılan bu fonksiyonların sistem çözümlemeleri çok kolay bir şekilde yapılabilmektedir.

Laplace yönteminin bir avantajı sistemin diferansiyel denklemlerini çözmeden sistem performansının kestirimi için grafiksel teknikler kullanılmasına olanak sağlamasıdır. Laplace dönüşüm yönteminin diğer bir avantajı da diferansiyel denklem çözümlerinde çözümün hem geçici durum bileşeninin hem de kalıcı durum bileşeninin aynı anda elde edilebilmesidir.

Aşağıda çok değişkenli sistemlerin transfer fonksiyonları bulurken kullanılan iki kural verilmiştir.

1. Ardışık (Seri) dizi kuralı: Şekil 5.5'deki gibi ardışık bağlanmış $G_1(s)$ ve $G_2(s)$ gibi iki sistemin transfer fonksiyonu aşağıdaki gibi tanımlanır:

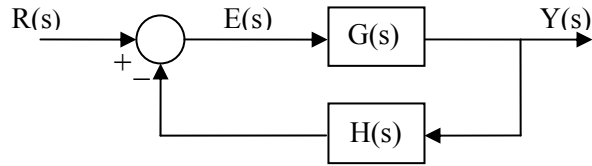
$$G(s) = G_1(s)G_2(s)$$



Şekil 5.5. Ardışık kontrol sisteminin temel blok diyagramı

2. Geribesleme kuralı: Şekil 5.6'daki gibi doğrusal geribeslemeli bir kontrol sisteminin blok diyagramı görülmektedir. Sistemin kapalı çevrimli transfer fonksiyonu $M(s)$ aşağıdaki şekilde ifade edilebilir:

$$M(s) = \frac{Y(s)}{R(s)} = \frac{G(s)}{1 \pm H(s)G(s)}$$



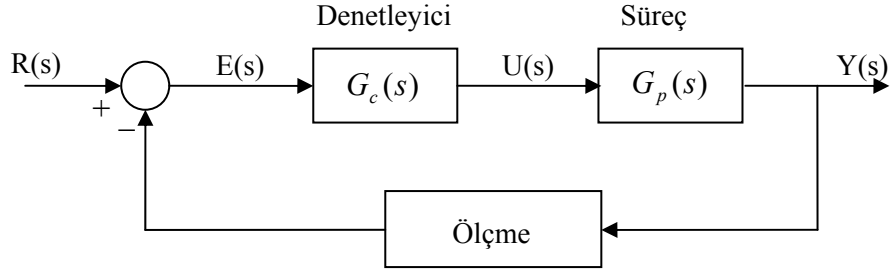
Şekil 5.6. Geribeslemeli bağlantı

5.6. Temel Denetim Organları

Kontrol sisteminde denetim organının görevi, geribeslenen çıkış büyüklüğünü giriş büyüklüğü ile karşılaştırarak ortaya çıkabilecek hata değerinin yapısına ve kendi denetim etkisine bağlı olarak uygun bir kontrol sinyali üretmektir. Kontrol organlarında kullanılan belli başlı dört temel denetim etkisi vardır [2]. Bunlar:

1. İkili veya aç-kapa (on-off) denetim etkisi
2. Orantı denetim etkisi (P etkisi)
3. İntegral denetim etkisi (I etkisi)
4. Türev denetim etkisi (D etkisi)

Bu temel denetim etkilerinin bir veya birkaçının bir arada uygun şekilde kullanılmasıyla değişik kontrol etkilerinde çalışan PI, PD ve PID gibi denetim organları oluşturulur. Şekil 5.7'de kontrol edilen bir sistemin blok şeması gösterilmiştir.



Şekil 5.7. Kontrol edilen bir sistemin blok diyagramı

5.6.1. İkili veya Aç-Kapa Denetim Organı

Aç-kapa denetim etkisi kullanan denetim organları oldukça basit yapıdadır. İkili denetim organı sadece iki belirli konumda bulunur. Hata değerine bağlı olarak denetim organı ve buna bağlı olarak motor eleman devrede veya devrede değildir.

İkili denetim organı özellikle çok hassas denetim gerektirmeyen yerlerde yaygın olarak kullanılırken, hassas denetim gerektiren süreç endüstrilerinde nadiren kullanılır.

5.6.2. Orantı Denetim Etkisi (P-Denetim)

Orantı etkide, denetim organı çıkışı bir oransal sabit yoluyla denetim girişine oranlanır. Denetim organı çıkışı $u(t)$ ile girişi hata sinyali $e(t)$ arasındaki bağıntı,

$$u(t) = K_p e(t) \quad (5.1)$$

şeklinde verilir. Burada, K_p orantı kazancı olup bu etkinin transfer fonksiyonu,

$$\frac{U(s)}{E(s)} = K_p \quad (5.2)$$

şeklinde orantı kazancına eşit bir sabittir.

Orantı etkide, herhangi bir anda denetim organı çıkışı $u(t)$, hatanın büyüklüğüne bağlıdır ve o anda hata ne kadar büyük olursa düzeltici denetim sinyali $u(t)$, o oranda büyük olur. Hata çok küçük olduğunda ise denetim organı yeteri kadar etkili düzeltici sinyal üretemez. Bu nedenle orantı etki kalıcı durum hatası verir. Kazanç katsayısı

K_p 'nin artırılması ile kalıcı durum hatasını azaltmak mümkündür. Orantı etkinin en önemli özelliği yapısının basitliğidir. Basit bir kuvvetlendirici yardımıyla orantı etkide çalışan denetim organı gerçekleştirmek mümkündür.

Orantı Bandı: Genel olarak süreç denetim sistemlerinde, denetleyici kazancı, denetleyici orantı bandı (PB) veya yüzdesi %PB cinsinden ifade edilir. Orantı bandı denetim etkisinin %100 lük değişim meydana getiren kısmı sapmayı gösterir. Orantı bandı ne kadar dar olursa o kadar hassas (kazancı yüksek) bir denetim elde edilir. Orantı bandı ne kadar genişletilecek olursa denetici hassasiyeti de o oranda düşer.

5.6.3. Integral Denetim Etkisi ve Orantı Artı İntegral Denetim Organı (PI-Denetim)

Orantı etkide ortaya çıkan kalıcı durum hatasını gidermenin yolu, denetim organına hatanın integrali ile orantılı bir denetim etkisi ilave etmektir. İntegral denetim yasası,

$$u(t) = K_i \int_0^t e(t)dt \quad \text{veya} \quad u(t) = \frac{1}{T_i} \int_0^t e(t)dt \quad (5.3)$$

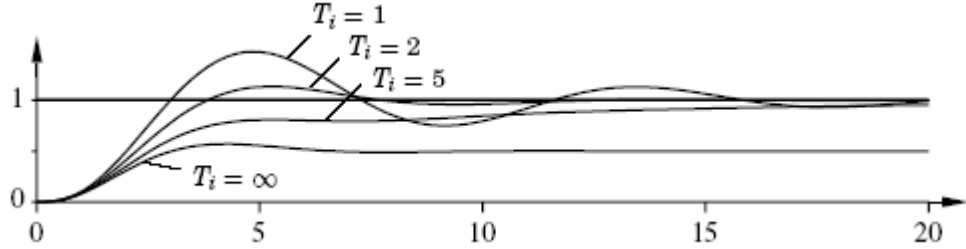
şeklinde ifade edilir ve denetim organı çıkışı $u(t)$, hata $e(t)$ 'nin zaman integrali ile orantılıdır. Burada K_i integral etki kazancı, T_i integral zaman sabiti adını alır. Bu etkinin transfer fonksiyonu,

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad \text{veya} \quad \frac{U(s)}{E(s)} = \frac{1}{T_i s} \quad (5.4)$$

şeklinde ifade edilir. İntegral etkinin çıkışı geçmişte meydana gelen hatanın birikimi ile orantılıdır ve herhangi bir anda hatanın integrali büyük olursa, büyük bir düzeltme etkisi sağlanır.

İntegral etki zamanı T_i çok yüksek değerde olduğu zaman hatanın sıfırlama işlemi çok uzun zaman almaktadır. Buna karşılık T_i çok küçük değerde olduğu zaman cevabın olması gerektiği değere ilk defa ulaşması çok kısa zamanda olmakla beraber, eğri ulaşması gereken değere ulaşmadan önce biraz salınım yapmaktadır. Hem az salınımlılık hem de hızlı cevap göz önünde bulundurularak integral etki zamanı için

uygun bir deęer seilmelidir. T_i ayar deęerlerinin sistem cevabına etkisini daha iyi anlamak amacıyla Şekil 5.8’de $G_p(s) = 1/(s+1)^3$ sistemi için T_i deęişimleri gösterilmiştir ($K = 1$).



Şekil 5.8. T_i ayar deęerinin sistem cevabı üzerindeki etkileri

Teorik olarak integral etki tipi denetim organının tek başına kullanılması mümkün ise de uygulamalarda integral etki daha çok oransal etki ile beraber kullanılır. Denetim organına bir integral alıcı ilavesi hata sıfır olana kadar deęişimi sürdüren bir denetim etkisi kazandırmaktadır.

5.6.4. Türev Denetim Etkisi ve Orantı Artı Türev Denetim Organı (PD-Denetim)

Türev etki geribeslemeli bir sistemin cevap hızını ve sönümlenmesini (geçici durum davranışını) iyileştirir. Türev denetim etki çıkışı $u(t)$, hata $e(t)$ ’nin zamana baęlı türev ile orantılıdır. İdeal olarak,

$$u(t) = K_d \frac{de(t)}{dt} \text{ veya } u(t) = T_d \frac{de(t)}{dt} \quad (5.5)$$

şeklinde ifade edilir. Transfer fonksiyonu

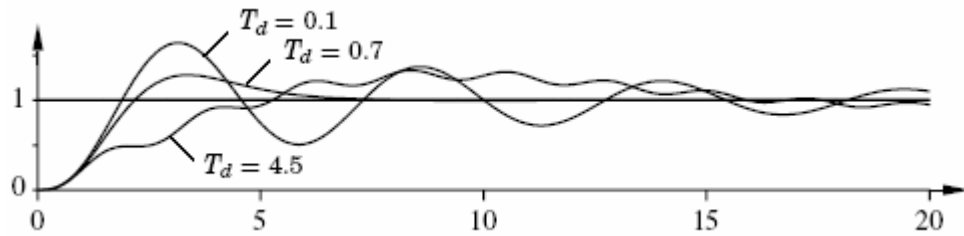
$$\frac{U(s)}{E(s)} = K_d s \text{ veya } \frac{U(s)}{E(s)} = T_d s \quad (5.6)$$

şeklinde elde edilir. Burada K_d türev denetim organı kazancı, T_d ise türev etki zamanı olarak adlandırılmaktadır.

Bir sabitin türevi sıfır olduğundan türev etkinin zamanla deęişmeyen hata üzerinde

etkisi yoktur. Türev etkinin en önemli üstünlüğü, hatanın büyümesini önceden kestirmesi ve büyük bir hata ortaya çıkmadan bir düzetme etkisi sağlamasıdır. Türev etki daha hata değişmeye başlar başlamaz harekete geçtiğinden “*önceden seziş*” etkisi olarak ta bilinir.

Orantı etki hatadaki değişimlere hızlı bir tepki göstermekle beraber hatanın değişim hızına duyarsızdır. Bu durumda sisteme hatanın değişim hızına duyarlı olan türev denetim etkisi ilavesi uygun olmaktadır. T_d ayar değerinin sistem cevabına etkisi Şekil 5.9’da $G_p(s) = 1/(s+1)^3$ sistemi için gösterilmiştir ($K = 3, T_i = 2$).



Şekil 5.9. T_d ayar değerinin sistem cevabı üzerindeki etkileri

5.6.5. Orantı Artı Integral Artı Türev Denetim Organı (PID-Denetim)

PID denetim üç temel denetim etkisinin (P, I, D) birleşiminden meydana gelmiştir. PID denetim, üç temel denetim etkisinin üstünlüklerini tek bir birim içinde birleştiren bir denetim etkisidir. Integral etki sistemde çıkabilecek kalıcı durum hatasını sıfırlarken türev etkide yalnızca PI denetim etkisi kullanılması haline göre sistemin aynı bağıl karalılığı için cevap hızını artırır. Buna göre PID denetim organı sistemde sıfır kalıcı durum hatası olan hızlı bir cevap sağlar.

PID denetim organı P, PI, PD denetim organlarına göre daha karmaşık yapıda ve maliyeti de diğerlerine göre daha pahalıdır. PID denetimin sağlayacağı üstün özelliklerden yararlanmak için K_p, T_i, T_d sırasıyla orantı kazancı, integral zaman sabiti ve türev zaman sabiti kontrol parametrelerinin uygun bir şekilde ayarlanması gereklidir.

Bu parametrelerin belirlenmesine yönelik detaylı bilgi ilerleyen konularda verilecektir.

5.7. Sistemlerin Kararlılığı

Tasarımda kullanılan pek çok davranış kriteri arasında, en önemli koşul sistemin kararlı olmasıdır. Doğrusal, doğrusal olmayan, zamanla değişmeyen ve zamanla değişen tüm sistemler göz önünde bulundurulduğunda kararlılık tanımı çok farklı şekillerde verilebilir. Genel olarak kararlı bir sistem, sınırlı bir cevaba sahip sistem olarak tanımlanabilir. Kararlı sistemler bozucu girişler karşısında geçici durum davranışı gösterdikten sonra tekrar denge konumuna geçerler. Kararlı bir sistemde, sistem cevabı düzgün veya küçülen genlikli titreşim şeklinde azalır. Doğrusal denetim sistemlerinin kararlılığı, kapalı çevrimli transfer fonksiyonunun kutuplarından yararlanılarak belirlenir. Bir sistemin kararlı olup olmadığı o sistemin kendisiyle ilgili olup sistemin girişinden bağımsızdır. Giriş fonksiyonu kutupları sistemin kararlılığını etkilemez. Analiz ve tasarım amacıyla kararlılık kavramı *mutlak kararlılık* ve *bağıl kararlılık* olarak sınıflandırılır. Mutlak kararlılıkta sistemin kararlı veya kararsız olma sorusu bir evet ya da hayır ile cevaplandırılabilir. Sistem kararlı bulunmuş ise sistemin ne kadar kararlı olduğunun ölçüsü bağıl kararlılık ile belirlenir [56].

Bir sistemin kararlı olup olmadığı farklı yöntemlerle incelenebilir. Yaygın olarak kullanılan yöntemler, (i) zaman-alanı yaklaşımı, (ii) karmaşık s -düzlemi yaklaşımı ve (iii) frekans alanı yaklaşımlarıdır.

Zaman alanı yaklaşımında, çeşitli giriş fonksiyonlarına bağlı olarak bulunan cevap fonksiyonlarında zaman sonsuza giderken sistemin cevabı sonlu bir değerde kalıyorsa sistem kararlıdır, aksi takdirde sistem kararsızdır denir. Karmaşık s -düzlemi yaklaşımında sistemin transfer fonksiyonu kutuplarının karmaşık düzlemdeki yerleri sistemin kararlılığı hakkında bilgi verir. Geribeslemeli bir sistemin kararlı olabilmesi için gerek ve yeter şart sistem transfer fonksiyonu kutuplarının negatif gerçek kısımlara sahip olmasıdır. Transfer fonksiyonu kutupları sistemin kararlılığı yanında sistemin dinamik davranışı hakkında da bilgi verir. Transfer fonksiyonunun kutupları bulunurken sistemin paydası olan özyapısal denklemin köklerinin belirlenmesi gerekir. Özyapısal denklemin derecesi küçük ise kökler kolaylıkla bulunurken, derecesi büyüdükçe köklerin bulunması zorlaşır. Karmaşık sayı düzleminde özyapısal denklemin köklerinin bulunmasına gerek kalmadan Routh-Hurwitz kararlılık ölçütü ile sistemin kararlılığı hakkında bilgi elde edilebilir. Frekans alanı yaklaşımında Nyquist yöntemi gibi

yöntemler uygulanmakta olup, burada kararlılık ölçütü olarak kazanç payı ve faz payı tanımlanmıştır.

5.7.1. Routh-Hurwitz Kararlılık Ölçütü

Routh-Hurwitz ölçütü bir polinom denklemin pozitif gerçel kısmı köklerinin olup olmadığını, denklem çözülmeden anlaşılmasını sağlar. Bu yöntem özellikle yüksek dereceden polinomlarda köklerin incelenmesinde önemli kolaylıklar sağlar. Routh-Hurwitz ölçümünün zayıf tarafı sadece mutlak kararlılık hakkında bilgi vermesidir. Özyapısal denklemin kökleri belli olmadığı için bağıl kararlılık hakkında bilgi vermez. Aşağıdaki gerçel katsayılı özyapısal bir denklemin genel ifadesi verilmiştir:

$$a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_2 s^2 + a_1 s + a_0 = 0$$

Routh-Hurwitz ölçütü bir doğrusal sistemin kararlılığı için aşağıda izah edilen gerek ve yeter şartlar ortaya koyar.

Gerek şartlar: Özyapısal denklemin tüm a_i , $i = 0, 1, 2, \dots, n-1$, katsayılarının pozitif ve sıfırdan farklı olması gerekir. Aksi takdirde sistem kararsızdır yada sınırda kararlıdır.

Yeter şartlar: Pozitif değerli katsayılarla sahip bir sistemin kararlılığının kesin olarak belirlenmesi için Routh tarafından ileri sürülen aşağıdaki tablodan yararlanılır:

s^n	a_n	a_{n-2}	a_{n-4}	a_{n-6}	L
s^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	a_{n-7}	L
s^{n-2}	b_1	b_2	b_3	b_4	
s^{n-3}	c_1	c_2	c_3	c_4	
s^3	d_1	d_2	0		
s^2	e_1	e_2	0		
s^1	f_1	0			
s^0	g_1				

Routh tablosunda ilk iki satıra doğrudan polinomun katsayıları yazılmaktadır. s 'in kuvvetleri indekslemek amacı ile kullanılmıştır. Üçüncü ve onu izleyen satırlar aşağıdaki hesaplama yöntemi ile elde edilmektedir.

$$b_1 = \frac{-\begin{vmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-2} - a_n a_{n-3}}{a_{n-1}}$$

$$b_2 = \frac{-\begin{vmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-4} - a_n a_{n-5}}{a_{n-1}}$$

$$b_3 = \frac{-\begin{vmatrix} a_n & a_{n-6} \\ a_{n-1} & a_{n-7} \end{vmatrix}}{a_{n-1}} = \frac{a_{n-1}a_{n-6} - a_n a_{n-7}}{a_{n-1}}$$

Benzer şekilde $b_4, b_5, \dots$ katsayıları bulunarak üçüncü satır tamamlanmış olur.

Dördüncü satırı oluşturmak içinde benzer bir yol izlenerek,

$$c_1 = \frac{-\begin{vmatrix} a_{n-1} & a_{n-3} \\ b_1 & b_2 \end{vmatrix}}{b_1} = \frac{b_1 a_{n-3} - b_2 a_{n-1}}{b_1}$$

$$c_2 = \frac{-\begin{vmatrix} a_{n-1} & a_{n-5} \\ b_1 & b_3 \end{vmatrix}}{b_1} = \frac{b_1 a_{n-5} - b_3 a_{n-1}}{b_1}$$

katsayıları bulunur. Benzer yoldan gidilerek tablonun diğer tüm elemanları da belirlenebilir. Tablonun 1. sütunundaki elemanlar sistemin kararlılığının belirlenmesinde kullanılır. Özyapısal denklemin pozitif gerçel kısmı köklerinin sayısı tablonun 1. sütunundaki elemanların işaret değiştirme sayısına eşittir. Gereklilik şartını sağlayan bir sistemin kararlı olabilmesi için Routh tablosunun 1. sütunundaki elemanlardan hiç birinin işaret değiştirmemesi gerekir. Yöntemin uygulanışı aşağıda verilen örnek üzerinde gösterilmiştir.

Örnek 5.1: Transfer fonksiyonu,

$$G_p(s) = \frac{1}{s^4 + 6s^3 + 11s^2 + 6s}$$

olarak verilen sistemin özyapısal denklemi $s^4 + 6s^3 + 11s^2 + 6s = 0$ dır. Görüldüğü gibi denklem gereklik şartını sağlamaktadır. Sistemin kesin olarak kararlı olduğunu

söyleyebilmek için Routh tablosu çıkarılarak yeterlilik şartını sağlayıp sağlamadığına bakılmalıdır.

$$\begin{array}{r} s^4 \quad 1 \quad 11 \quad 0 \\ s^3 \quad 6 \quad 6 \quad 0 \\ s^2 \quad 10 \quad 0 \\ s^1 \quad 6 \\ s^0 \quad 0 \end{array}$$

Routh tablosunun birinci sütunu incelendiği zaman işaret değişimi olmadığı görülmektedir. Böylece hem gereklilik şartını hem de yeterlilik şartı sağlandığından dolayı dikkate alınan sistemin kararlı olduğu sonucuna varılabilir.

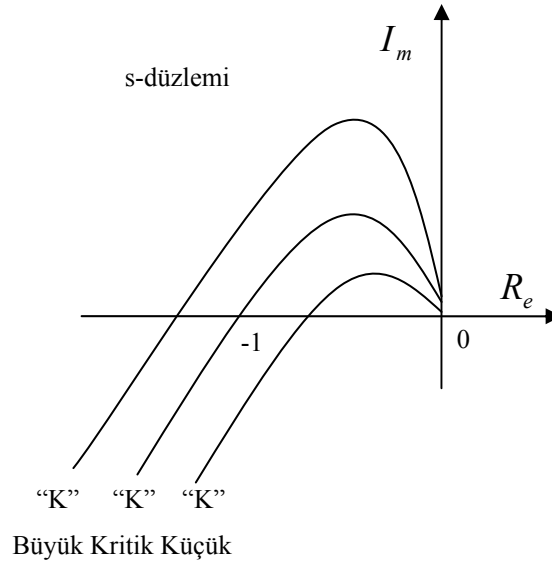
5.7.2. Nyquist Kararlılık Ölçütü

Nyquist kararlılık ölçütü, açık çevrimli transfer fonksiyonu $G(j\omega)H(j\omega)$ 'nın frekans cevabı eğrilerinden hareketle sistemin kapalı çevrimli halinde kararlılığının belirlenmesini sağlayan grafiksel bir yöntemdir. Kapalı çevrimli transfer fonksiyonu,

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)}$$

olan sistemin kararlı olabilmesi için $1 + G(s)H(s) = 0$ tüm köklerin sol yarı s -düzleminde olması gerekir. Nyquist kararlılık ölçütü açık çevrimli frekans cevabı ile $1 + G(s)H(s)$ in sağ yarı s -düzleminde yer alan sıfırlarının sayısı arasında bir bağlantı kurar. Bu ölçüt ilk defa H. Nyquist tarafından bulunmuş olup kapalı çevrimli sistemin mutlak kararlılığını, kapalı çevrimli kutuplarının bulunmasına gerek kalmadan grafiksel olarak açık çevrimli frekans cevabı eğrilerinden bulunmasını sağlar.

Nyquist ölçütü bir sistemin bağıl kararlılığı hakkında da bilgi edinilmesini sağlar. Şekil 5.10'da üç farklı kazanç katsayısı K için çizilen $G(j\omega)H(j\omega)$ kutupsal eğrileri ele alındığında K nın çok büyük değerleri için sistem kararsızdır. K nın belli bir değere kadar küçültülmesi sonucu eğri $(-1+j0)$ noktasından geçer. Bu noktada sistem kararsızlık eşiğindedir ve dinamik olarak sönümsüz titreşimli bir davranış gösterir. K nın küçük değerleri için ise sistem kararlıdır.



Şekil 5.10. Kazanç katsayısına bağlı olarak kutupsal eğri değişimi

Kontrol sistemlerinin analiz ve tasarımında, Nyquist kriterinin aşağıdaki özellikleri, yaklaşımı alternatif bir yöntem olarak çekici kılar [56]:

1. Nyquist kriteri, Routh-Hurwitz kriterinde olduğu gibi, sistemin mutlak kararlılığı, kararlı bir sistemin bağıl kararlılığı ve kararsız sistemin kararsızlığı konusunda bilgi sağlar.
2. $G(j\omega)H(j\omega)$ Nyquist yer eğrisi, özellikle bilgisayar yardımıyla çok kolay oluşturulabilir.
3. $G(j\omega)H(j\omega)$ Nyquist yer eğrisi ile diğer frekans tanım bölgesi karakteristikleri ile ilgili kolay bilgi edinilebilir.
4. Nyquist yer eğrileri ile Routh-Hurwitz kriteriyle incelenemeyen ve kök yer eğrileri güçlüklerle işlenebilen ölü zamanlı sistemler kolaylıkla değerlendirilebilir.

5.8. Performans İndisleri

PID denetleyici parametrelerinin bulunmasında, tasarlanan sistemin performansının değerlendirilmesinde farklı hata fonksiyonları kullanılmaktadır. Denetleyici tasarımında kullanılan hata fonksiyonlarına göre ölçülen geçici durum ve kalıcı durum parametreleri farklılık gösterebilmektedir. Aşağıda yaygın olarak kullanılan hata fonksiyonları hakkında bilgi verilmiştir.

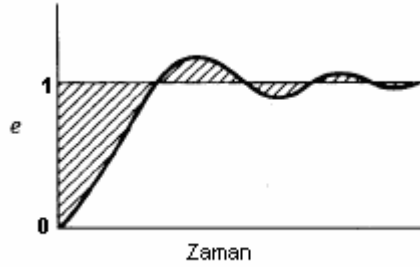
Mutlak Hatanın İntegrali (Integral of the Absolute Error, IAE):

IAE hata değeri sistemin zaman alanı cevabının geçici durum çalışması halinde meydana gelen hata değerlerinin toplamı ile ifade edilir. Şekil 5.11’de grafiği verilen IAE hata değeri Denklem 5.7 ile hesaplanmaktadır.

$$IAE = \int_0^{\infty} |e(t)| dt$$

$$e(t) = 1 - y(i)$$

$$IAE = \sum_i abs(1 - y(i)) \quad (5.7)$$



Şekil 5.11. IAE hata fonksiyonu grafiği

Karesel Hatanın İntegrali (Integral of the Square Error, ISE):

ISE hata değeri, sistemin zaman alanı cevabının geçici durum çalışması halinde meydana gelen hata değerlerinin kareleri toplamı ile ifade edilir. ISE hata kriterinde hatanın karesi alındığından dolayı büyük hatanın etkisi, küçük hatanın etkisine göre çok daha fazla olmaktadır. Sistem zaman alanı cevabında büyük taşmalar olduğunda toplam hata çok yüksek çıkarken sürekli küçük salım gösteren sistemlerde toplam hata daha az olmaktadır. ISE hata değeri Denklem 5.8 ile hesaplanmaktadır.

$$ISE = \int_0^{\infty} [e(t)]^2 dt$$

$$e(t) = 1 - y(i)$$

$$ISE = \sum_i (1 - y(i))^2 \quad (5.8)$$

Zaman Ağırlıklı Mutlak Hatanın İntegrali (Integral of the Time-weighted Absolute Error, ITAE):

ITAE hata değeri sistemin geçici durum çalışması halinde meydana gelen hata değerlerinin hatanın meydana geldiği zamanla çarpılıp toplanması ile belirlenir. ITAE hata kriterinde hatanın zamanla çarpılmasından dolayı başlangıçta meydana gelen hatanın toplam hataya etkisi az olurken ilerleyen zamanlarda meydana gelen hataların toplam hataya etkisi fazla olur. Böylece başlangıçta büyük olan hata ile ilerleyen zamanlardaki düşük olan hata değerlerinin toplam hata üzerindeki etkileri dengelenmiş olur. Sürekli salınım gösteren sistemlerde toplam hata çok yüksek değerler alır. ITAE hata değeri Denklem 5.9 kullanılarak hesaplanmaktadır.

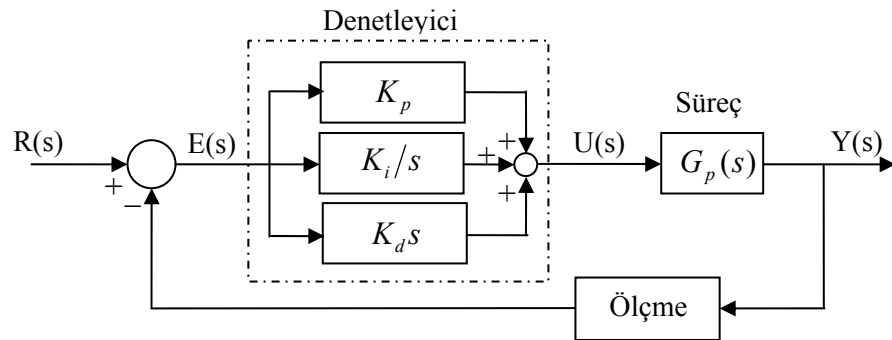
$$ITAE = \int_0^{\infty} t|e(t)|dt$$

$$e(t) = 1 - y(i)$$

$$ITAE = \sum_i (abs(1 - y(i))) * t(i) \quad (5.9)$$

5.9. PID Kontrol Ediciler

Hâlihazırda üç terimli kontrol edici olan PID kontrol ediciler şu an için endüstride kullanılan otomatik geribeslemeli kontrol cihazlarının %80'inden fazlasında kullanılmaktadır. Geçmişte bunlar frekans analiz metotları yardımı ile ayarlanırken günümüzde modern yaklaşımda PID kontrol edicinin ayarları süreç modeline bağlı olarak oluşturulmaktadır. Şekil 5.12'de verilen PID denetleyicili sistemin denetleyici transfer fonksiyonu Denklem 5.10 formundadır.



Şekil 5.12. PID Denetleyicili Sistem

$$G_c(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s \quad (5.10)$$

Buradaki K_p, K_i, K_d , sırasıyla ile oransal, integral ve türev kazançlarıdır. $U(s)$ kontrol edicinin çıkışı, $E(s)$ ise hata sinyali ve kontrol edicinin girişidir. Yukarıdaki eşitlik bir başka formda da:

$$\frac{U(s)}{E(s)} = K_c \left[1 + sT_d + \frac{1}{T_i s} \right] \quad (5.11)$$

ile de ifade edilebilir. Burada $T_i = K_p / K_i$ ve $T_d = K_d / K_p$ olup, bu parametreler sırasıyla integral ve türev zaman sabitleri olarak isimlendirilir ve $K_p \approx K_c$ ise kontrol edici kazancıdır. Bazı kontrol ediciler kazanç yerine $K_c = 100 / PB$ gibi oransal band ayarını kullanmaktadır. Burada PB oransal durumdur ve kontrol edici çıkışı %100 değiştiği zamanki yüzde hata miktarı olarak tanımlanır. PB orantı bandı ne kadar dar olursa o kadar hassas (kazancı yüksek) bir denetim elde edilirken, geniş olduğu zaman da denetleyici hassasiyeti o oranda düşmektedir. Türevsel terimden dolayı frekans artırımı ile kontrol edici kazancı da artmaktadır. Bu yüzden pratik PID kontrol ediciler bu yüksek frekanslardan kaynaklanan kazançları birinci dereceden bir ön eleman (alçak geçiren süzgeç) kullanarak sınırlar. Bu filtrenin zaman sabiti ε genelde türev terimi kazancının onda biri mertebesinde seçilir. Pratikte kullanılan PID kontrol edicinin transfer fonksiyonu,

$$\frac{U(s)}{E(s)} = \frac{K_p s^2 + K_d s + 1}{s(\varepsilon K_d s + 1)} \quad (5.12)$$

veya

$$\frac{U(s)}{E(s)} = K_c \left[1 + \frac{1}{T_i s} \right] \left[\frac{1 + T_d s}{1 + \varepsilon s} \right] \quad (5.13)$$

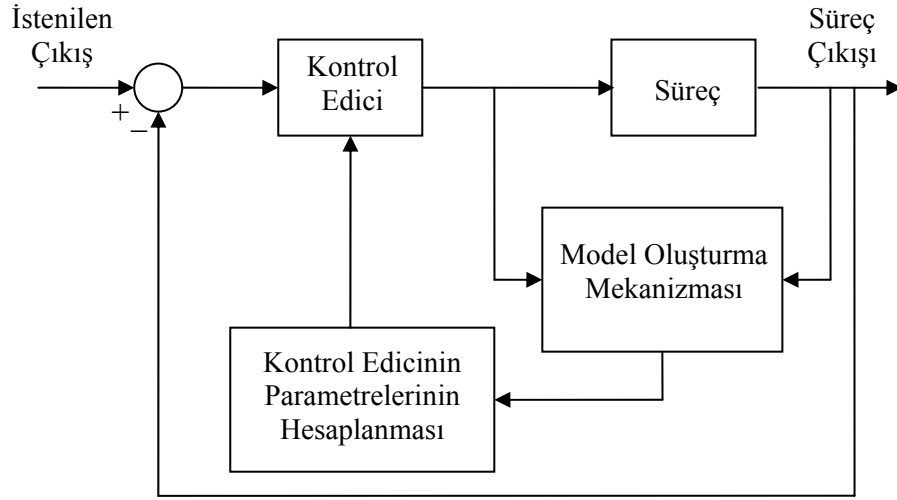
ile ifade edilir.

PID kontrol edici tasarımının temel hedefi K_p, K_d, K_i veya K_c, T_i, T_d katsayılarının bulunması ve verilen kapalı çevrimli sistemin performans koşullarını sağlayacak şekilde ayarlanmasıdır. Bu kazançlar sistemin geçici başlangıç tepkisi, gürültü yok edimi ve durum uzayı hata özellikleri yardımı ile bulunur. Pratikte bu üç kavramın hepsini aynı anda kullanmak çok mümkün değildir. Örnek olarak eğer kontrol edici ayarları daha iyi bir geçici sistem tepkisi vermek üzere ayarlanmış ise (minimum taşma ve minimum oturma zamanı) genelde gürültü değişimlerine daha ağır tepki verir.

PID Kontrol edici tasarımında izlenecek temel basamaklar aşağıda verilmiştir:

1. Sistemin açık çevrimli transfer fonksiyonunun elde edilmesi ve nelerin geliştirilmesi gerektiğine karar verilmesi,
2. Sisteme yükselme zamanını iyileştirmek için oransal denetleyici eklenmesi,
3. Sistemin ani tepkisini iyileştirmek için türevsel denetleyici eklenmesi,
4. Kararlı durum hatasını yok etmek için integral denetleyicisinin sisteme eklenmesi,
5. Bütün K_p, K_d, K_i 'lerin istenen genel tepkiyi verecek şekilde ayarlanması.

PI kontrol edici birinci dereceden doğrusal zamanla değişmeyen sistemler için optimum bir çözüm sunmaktadır. Benzer olarak da PID kontrol edicide ikinci dereceden doğrusal zamanla değişmeyen sistemler için optimum çözüm sunabilmektedir. Bununla beraber gerçek hayattaki sistemler çoğu zaman doğrusal değildir ve zamanla süreç karakteristikleri değişiklik göstermektedir. Dolayısı ile başlangıç kontrol edici tasarımı için doğrusal bir model seçimi, sistemin zaman ile değişerek farklı bir bölgede işlemeye başladığı durumlarda uygulanabilirlikten çıkacaktır. Buna bir çözüm birden fazla kontrol edici tanımlamalarının daha önceden hafızaya alınması veya bir yerde tutulması ve süreç çalışma bölgesini değiştirdikçe buna uygun kontrol edicinin devreye girmesini sağlamaktır. Bu strateji parametrik veya kazanç tarifli kontrol olarak adlandırılır. Daha hassas bir teknik ise kontrol ediciyi uyarlanır olacak şekilde ayarlamaktır. Bu şekilde doğrusal modelin parametreleri o anki sürecin karakteristiklerine bağlı olarak sürekli güncellenecektir. Bu gibi yapılar genelde otomatik-ayarlı/ uyarlanır / kendinden-ayarlı gibi tanımlamalarla nitelendirmektedir. Uyarlanır kontrol edicinin blok şeması Şekil 5.13'de gösterilmiştir.



Şekil 5.13. Uyarlanırlı kontrol edicinin basit gösterimi

PID denetleyici çeşitli avantajlara sahip olmakla birlikte, tüm sistemlerin kullanılması zorunluluğu yoktur. Örneğin bir sistem için PI denetleyici iyi sonuç veriyor ise türevsel denetleyiciye gerek olmayabilir.

Tablo 5.1’de orantı, integral ve türev kazançlarının zaman alanı cevabı parametreleri üzerindeki etkisi özetlenmiştir [57].

Tablo 5.1. Orantı, integral ve türev kazançlarının etkileri

Kazanç	Yükselme zamanı	Sistem ani tepkisi	Oturma zamanı	Kalıcı durum hatası
K_p	Azalır	Artar	Ufak değişim gösterir	Azalır
K_i	Azalır	Artar	Artar	Yok eder
K_d	Ufak değişim gösterir	Azalır	Azalır	Ufak değişim gösterir

5.9.1. Tek Girişli ve Tek Çıkışlı Sistem Tasarım Metotları

Tek girişli Tek çıkışlı sistemler (Single Input Single Output, SISO) için PID kontrol edici tasarım yöntemlerinden bazıları Ziegler-Nichols, Cohen ve Coon, İç Model

Kontrolü ve Ho-Hang-Cao yöntemleridir [58]. Aşağıda bu yöntemlerden Ziegler-Nichols yöntemi tanıtılmıştır.

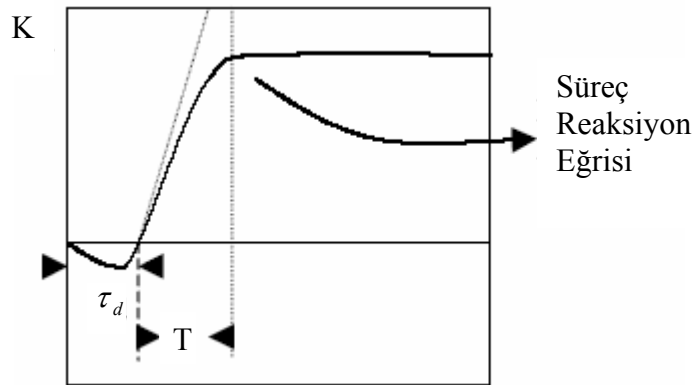
5.9.1.1. Ziegler-Nichols Metodu

Ziegler-Nichols (Z-N) tasarım metodu PID kontrol edicilerin parametrelerinin belirlenmesinde kullanılan en yaygın yöntemlerden birisidir. Z-N'ın önerdiği yöntemlerden birincisi sürekli döngü metodu olarak adlandırılır ve bu metot da kontrol edici kazancı, bastırılmış salınım K_u kazancında sağlanana kadar arttırılır. Eğer tekabül eden salınım periyodu T_u ise o zaman PID kontrol edicinin parametreleri Z-N metoduna göre Tablo 5.2'deki gibi seçilir.

Tablo 5.2. Sürekli döngü için Z-N ayar kuralları

Kontrol Edici	Parametreler		
	K_c	T_i	T_d
P	$0.5K_u$	-	-
PI	$0.45K_u$	$(1/1.2)T_u$	-
PID	$0.6K_u$	$0.5T_u$	$(1/8)T_u$

Z-N in önermiş olduğu ikinci bir yaklaşımda, genelde süreç reaksiyon eğrisi metodu olarak bilinir. Bu metot da sistemin açık çevrimli birim basamak tepkisi incelenir ve genelde Şekil 5.14'deki gibi analiz edilir.



Şekil 5.14. Açık çevrimli basamak tepkisi

Sistem tepkisi τ_d , T ve K katsayıları ile verilen yaklaşık olarak düzgün bir doğru ile ifade ediliyor. Bu tepki birinci dereceden zaman gecikmeli (First Order Plus Time Delay, FOPTD) Denklem 5.14'deki gibi bir sistemin basamak tepkisini vermektedir.

$$G(s) = \frac{Ke^{\tau_d s}}{Ts + 1} \quad (5.14)$$

Bu yaklaşıma ait Z-N kuralları Tablo 5.3'de verilmiştir. Kontrol edicinin parametreleri bu tabloya uygun bir şekilde seçilebilir.

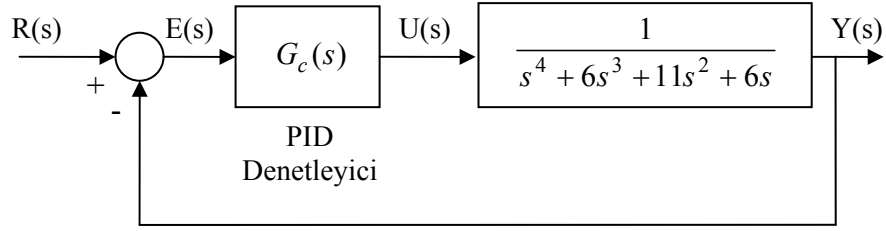
Tablo 5.3. Süreç eğrisi metodu için Z-N ayar kuralları

Kontrol Edici	Parametre		
	K_c	T_i	T_d
P	$\frac{T}{K\tau_d}$	-	-
PI	$\frac{0.9T}{K\tau_d}$	$\frac{\tau_d}{0.3}$	-
PID	$\frac{1.2T}{K\tau_d}$	$2\tau_d$	$0.5\tau_d$

Z-N metodunun iki temel dezavantajı vardır. Çoğunlukla tasarlanan kapalı çevrimli sistem istenenden daha çok salınıma sahiptir. Bir diğer dezavantajı ise dengelenmemiş sistemin kazanç ve frekansına bağlı olarak, oransal geribesleme döngüsünde salınıma sebep olmasıdır. Z-N yönteminin uygulanışına ilişkin bir örnek aşağıda verilmiştir.

Örnek 5.2: Şekil 5.15'de blok şeması verilen denetim sisteminde denetim organı olarak aşağıda verilen PID denetleyicisi kullanılmaktadır. Ziegler-Nichols titreşim yöntemine göre sınırlı kararlı çalışmasına karşılık gelen kazanç değeri ve bu kazanca karşılık gelen sürekli titreşim periyodunun bulunup, denetim organı parametrelerinin Tablo 5.2 kullanılarak ayarlanması:

$$\frac{U(s)}{E(s)} = K_p \left[1 + sT_d + \frac{1}{T_i s} \right]$$



Şekil 5.15. PID denetimli bir sistem

Çözüm: PID denetimli sistemin kapalı çevrimli transfer fonksiyonu $T_i = \infty, T_d = 0$ yapılarak,

$$\frac{Y(s)}{R(s)} = \frac{K_p}{s^4 + 6s^3 + 11s^2 + 6s + K_p}$$

olarak elde edilir. Buradan sistemi sınırdaki kararlı, sürekli titreşimli hale getiren kazanç değeri K_p Routh kararlılık ölçütü yolu ile tespit edilebilir. Sistemin özyapısal denklemi $s^4 + 6s^3 + 11s^2 + 6s + K_p = 0$ olduğuna göre Routh tablosu aşağıdaki şekilde düzenlenebilir.

s^4	1	11	K_p
s^3	6	6	
s^2	10	K_p	
s^1	$\frac{60 - 6K_p}{10}$		
s^0	K_p		

Routh tablosundaki birinci sütundaki elemanları incelenirse sistemin sınırlı kararlı ve sürekli titreşim halinde çalışmasını sağlayacak $K_{p\max} = K_u = 10$ olarak bulunur. Bu kazanç değerine karşılık özyapısal denklemin alacağı şekil $s^4 + 6s^3 + 11s^2 + 6s + 10 = 0$ olur. Sistemin sınırlı çalışması halinde köklerin sanal eksen üzerinde olması gerektiğinden $s = j\omega$ alınır ve Routh tablosunun 2. satırından $10(j\omega)^2 + 10 = 0$ ve bu denklemin çözümünden sürekli titreşim frekansı $\omega = 1$ rad/s olarak bulunur. Buna göre sürekli titreşim periyodu,

$$T_u = \frac{2\pi}{\omega} = \frac{2\pi}{1} = 6.28 \text{ sn}$$

olarak bulur. Tablo 5.2'ye göre K_p, T_i, T_d değerleri sırasıyla orantı kazancı $K_p = 6$, integral zaman sabiti $T_i = 3.14$ ve türev zaman zabiti $T_d = 0.79$ elde edilir. $K_d = K_p T_d$ ve $K_i = K_p / T_i$ eşitlikleri kullanılarak integral kazancı $K_i = 1.91$, türev kazancı ise $K_d = 4.74$ bulunur.

Örnek 5.3: Transfer fonksiyonu aşağıdaki şekilde verilen sistemin Ziegler-Nichols yöntemi kullanılarak parametre değerlerinin bulunması.

$$G_p(s) = \frac{e^{-s}}{s+1}$$

Çözüm: İlk olarak sürekli titreşimli frekansı aşağıdaki gibi hesaplanabilir ($s = j\omega$)

$$\begin{aligned} G_p(j\omega) &= \frac{e^{-j\omega}}{j\omega+1} \\ &= \frac{\cos \omega - j \sin \omega}{j\omega+1} \\ &= \frac{(\cos \omega - j \sin \omega)(-j\omega+1)}{\omega^2+1} \\ &= \frac{1}{\omega^2+1} (-j\omega \cos \omega - \omega \sin \omega + \cos \omega - j \sin \omega) \\ &= \frac{1}{\omega^2+1} (\cos \omega - \omega \sin \omega - j(\omega \cos \omega + \sin \omega)) \end{aligned}$$

Sistemin sınırlı çalışması halinde köklerin sanal eksen üzerinde olması gerektiğinden,

$$\omega \cos \omega + \sin \omega = 0$$

$$\omega = -\tan \omega$$

olur. En küçük pozitif sayı için bu denklemi çözdüğümüzde sürekli titreşim frekansı $\omega = 2.0288$ rad/s bulunur. Buna göre sürekli titreşim periyodu,

$$T_u = \frac{2\pi}{\omega} = \frac{2\pi}{1} = 3.097 \text{ sn}$$

dir. Sistemin sınırlı kararlı ve sürekli titreşim halinde çalışmasını sağlayacak kazanç değeri

$$K_u G_p(j\omega) = -1$$

$$K_u = \frac{1}{G_p(j\omega)}$$

$$K_u = 2.2619$$

bulunur. Tablo 5.2 kullanılarak K_p, T_i, T_d değerleri sırasıyla orantı kazancı $K_p = 1.3571$, integral zaman sabiti $T_i = 1.548$ ve türev zaman sabiti $T_d = 0.3871$ olarak elde edilir.

5.9.2. Çok Girişli ve Çok Çıkışlı Sistem Tasarım Metotları

Çok girişli Çok çıkışlı sistemler (Multi Input Multi Output, MIMO) için PID kontrol edici tasarımı bu tarz sistemlerin karmaşık yapısı yüzünden daha zordur. MIMO PID kontrol edici tasarımı sistem içinde tekil döngüler üzerine uygulanarak veya tüm sistemi bir arada düşünerek gerçekleştirilebilir. MIMO için PID kontrol edici tasarım metodlarından bazıları Ziegler-Nichols, Zhuang ve Atherton Optimizasyonu ve Seraji metodlarıdır [58].

Sonuç olarak, PID kontrol ediciler endüstriyel süreçlerin %80'inden fazlasında kullanılmaktadır ve 1940'lardan bu yana kontrol edicilerin tasarımı ve ayarlanması ile ilgili olarak oldukça fazla sayıda metod geliştirilmiş ve uygulamaya konulmuştur.

6. BÖLÜM

MEMETİK ALGORİTMA KULLANILARAK DENETLEYİCİ TASARIMI

6.1. Giriş

Elektrik, mekanik ve kimyasal bölümleri içeren ilk kontrol uygulamaları günümüzde diğer bilim dallarının anlaşılabilirliğinin artması ile birçok uygulamada geniş yer bulmuştur. Kontrol sistemleri uygulamalarının bu kadar geniş bir kullanım alanına sahip olması kontrol mühendisliğinin sınırlarını oldukça genişletmiştir.

PID denetleyici ve türevleri (P, PI, PD), yapısının basit olması ve çalışma şartlarının geniş bir aralığında oldukça kararlı olması nedeniyle endüstriyel süreç kontrolünde yaygın olarak kullanılmaktadır [59-61]. Bu tür kontrol sistemlerinin tasarımında, denetleyici parametrelerinin uygun şekilde belirlenmesi son derece önemlidir. Kontrol edilecek sistemin yapısı (doğrusal olmaması, derecesi v.b.) nedeniyle bu parametrelerin uygun şekilde belirlenmesi oldukça zor olabilmektedir. Dikkate alınması gereken diğer husus ise uyarlanırlı denetleyici tasarımında denetleyici parametrelerinin belirlenmesi için harcanacak sürenin sınırlı olmasıdır. Bu parametrelerin belirlenmesi amacıyla geliştirilmiş birçok klasik teknik bulunmaktadır [2,59,62,63]. Bu klasik teknikler çoğunlukla karmaşık kontrol sistem tasarımlarında etkin sonuçlar vermemekle birlikte yoğun matematiksel işlemlere ihtiyaç duymaktadırlar. Ayrıca hedeflenen tasarım makul bir sürenin ötesinde zaman gerektirebilmektedir. Bu gibi nedenlerle kontrol alanında çalışan araştırmacıların sezgisel algoritmalar ve yapay sinir ağları kullanımına olan ilgisi gittikçe artmaktadır. Karmaşık kontrol sistemlerini tasarlamak için GA, TA, BT ve YSA'lar üzerine dayalı yaklaşımların kullanılmasına ilişkin literatürde çeşitli sayıda çalışma bulunmaktadır.

1996 yılında Karaboğa ve Kalınlı PID denetleyici tasarımında tabu araştırma algoritması üzerine dayalı yeni bir yaklaşım önermiştir. Önerilen yaklaşım performansı,

bilinen bazı geleneksel yöntemlerle karşılaştırılmış ve TA algoritmasının bu amaçla başarılı bir şekilde kullanılabileceği gösterilmiştir [64]. 2004 yılında ise Puangdownreong ve arkadaşları uyarlanır bir TA algoritmasını PID denetleyici tasarımında başarıyla kullanmışlardır [65]. Shen ise örnek aldığı kontrol sistemlerinde genetik algoritma kullanarak PID tasarımı gerçekleştirmiştir [66]. Shen yine bir başka çalışmada bu defa kontrol parametrelerinin belirlenmesinde bulanık sinir ağlarını kullanmıştır [67]. Zhou ve Birdwell ise bir başka sezgisel yaklaşım olan benzetilmiş tavlama algoritması kullanarak denetleyici tasarımı gerçekleştirmiştir [68].

Granada üniversitesinde yapılan bir çalışmada Chiaberge ve arkadaşları genetik algoritma, benzetilmiş tavlama algoritması ve yapay sinir ağları yaklaşımını gerçek zamanlı bir kontrol sistemine uygulamışlardır. Aynı zamanda genetik algoritma ile benzetilmiş tavlama algoritmasını birleştirerek oluşturdukları hibrid algoritmada bu iki algoritmanın üstün özelliklerinden nasıl faydalandıklarını ve algoritmaların bağımsız kullanımlarına göre avantajlarını ortaya koymuşlardır [69].

Alli ve Kaya denetleyici parametrelerinin bulunmasında genetik algoritma kullanmışlar, ayrıca bu çalışmada tasarımda kullandıkları performans indislerini karşılaştırarak, en uygun hata fonksiyonunun ITAE olduğunu belirtmişlerdir [70].

Griffin 2003 yılında yapmış olduğu çalışmada genetik algoritma kullanarak farklı hata fonksiyonları ile gerçekleştirilen tasarımların performansını incelemiştir [71].

Dandil ve arkadaşlarının yapmış oldukları çalışmada doğrusal ve doğrusal olmayan çeşitli sistemler için genetik algoritma ile model referans PID denetleyici tasarımı gerçekleştirmiştir. Çalışmada doğrusal ve doğrusal olmayan çeşitli sistemler için PID denetleyici tasarımları gerçekleştirilmiştir. Önerdikleri GA-PID referans modeli ile klasik yöntemlerin aksine karmaşık tasarım işlemlerine gerek duymadan etkin sonuçlar elde edileceğini göstermişlerdir [72].

Bununla beraber, yapılan detaylı literatür taramasında memetik algoritmanın denetleyici tasarımında kullanılmasına ilişkin iki çalışmaya rastlanmıştır. Shyr ve arkadaşları genetik algoritma ve tepe tırmanma algoritmasını bir arada kullanarak oluşturdukları memetik algoritma modelini örnek bir PID optimizasyonu için test ederek elde ettikleri neticeleri genetik algoritma ile karşılaştırmışlardır [73]. Coelho ve arkadaşları 2002

yılında memetik algoritma kullanarak çok döngülü (multiloop) denetleyici tasarımı gerçekleştirmiştir [74].

6.2. Memetik Algoritma ile Denetleyici Tasarımı

Bu bölümde, Bölüm 3.4’de gerçekleştirilen memetik algoritma modeline dayalı olarak PID denetleyici tasarımı izah edilmiştir. Önerilen yaklaşımın performansı farklı özelliklere sahip 6 adet sistem üzerinde test edilmiştir. Benzetim çalışmalarından elde edilen sonuçlar klasik yöntemlerden olan Ziegler-Nichols metodu ve literatürde yer alan diğer bazı denetleyici tasarım metotlarıyla kıyaslanmıştır.

Literatürde yer alan çeşitli çalışmalarda performans fonksiyonu seçiminin tasarlanacak sistemin geçici durum ve kalıcı durum parametreleri üzerinde etkili olduğu gösterilmiştir [70,71]. Bu nedenle memetik algoritma ile gerçekleştirilecek bazı tasarımlarda kullanılacak hata fonksiyonlarının (ITAE, ISE, IAE) performans üzerindeki etkileri de incelenmiştir.

Elde edilen sonuçları karşılaştırmak için sistemin geçici durum parametrelerinden yükselme zamanı, maksimum taşma ve oturma zamanı dikkate alınmıştır. Yükselme zamanı olarak, sistem cevabının nihai değerinin %5 inden %95 ine kadar ulaşması için geçen süre dikkate alınırken, oturma zamanı için başlangıçtan itibaren sistem cevabındaki salınımın nihai değerin %2’lik aşma aralığına düşmesine kadar geçen süre dikkate alınmıştır.

Gerçekleştirilen memetik algoritma modelinde birbirinden bağımsız paralel çalışan BT algoritması sayısı 10, BT algoritması ile her bir kademedeki gerçekleştirilecek iterasyon sayısı 100 olarak alınırken, başlangıç sıcaklık değeri 450-550 aralığında, sıcaklık azaltma katsayısı ise 0.7-0.9 aralığında her denemede rasgele olarak belirlenmiştir. Genetik algoritma için, çaprazlama oranı 0.9 olarak seçilirken, mutasyon oranının 0.01-0.1 aralığında her bir kademedeki uyarlanırlar olarak değişmesi sağlanmıştır. Her bir katmandaki genetik algoritma için değerlendirme sayısı ise 50 olarak seçilmiştir. Genetik algoritmada değeri optimize edilecek her parametre 12 bit ile temsil edilmiş ve benzetilmiş tavlama ve genetik algoritma arasında bilgi alışverişi sırasında ikili-gerçel, gerçel-ikili sayı dönüşümleri gerçekleştirilmiştir.

Sistem 1:

Denetleyici tasarımı gerçekleştirilecek ilk sistem transfer fonksiyonu Denklem 6.1’de verilen dördüncü dereceden bir sistemdir.

$$G_p(s) = \frac{1}{s^4 + 6s^3 + 11s^2 + 6s} \quad (6.1)$$

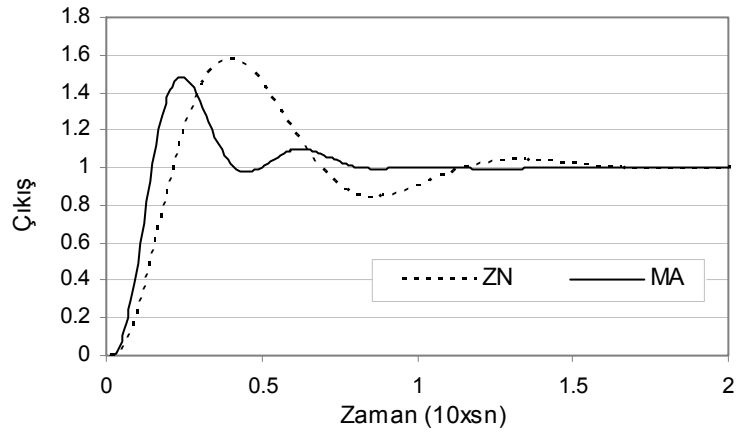
Sistemin klasik denetleyici tasarım yöntemi olan Z-N metodu ile çözümü için, sistemi sürekli titreşimli hale getiren sınırlı kararlı kazanç değeri Routh-Hurwitz kararlılık ölçütü ile $K_u = 10$ ve $T_u = 6.28$ olarak bulunur. Tablo 5.2 Ziegler-Nichols ayar kuralları kullanılarak sırasıyla orantı kazancı $K_p = 6$, integral zaman sabiti $T_i = 3.14$, türev zaman sabiti $T_d = 0.79$ bulunur.

Geliştirilen memetik algoritma yaklaşımı ile optimum değerleri bulunması hedeflenen K_c, T_i, T_d parametreleri $0 < K_c < K_u$, $0 < T_i < T_u$, $0 < T_d < T_u$ değer aralığında araştırılmıştır.

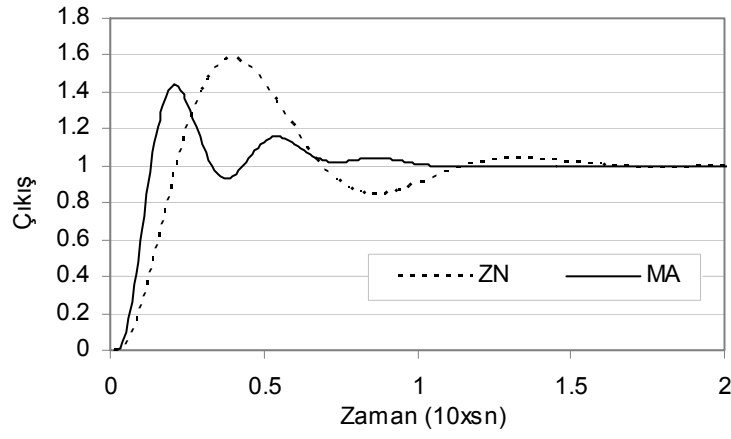
Memetik algoritma ile gerçekleştirilen tasarımda, parametre değerleri bulunurken birim basamak zaman alanı cevabı üzerinde farklı performans hata fonksiyonları için benzetimler gerçekleştirilerek elde edilen sonuçlar Z-N yöntemi ile bulunan sonuçlarla birlikte Tablo 6.1’de verilmiştir.

Tablo 6.1. Sistem 1 için Z-N ve MA ile belirlenen tasarım sonuçları

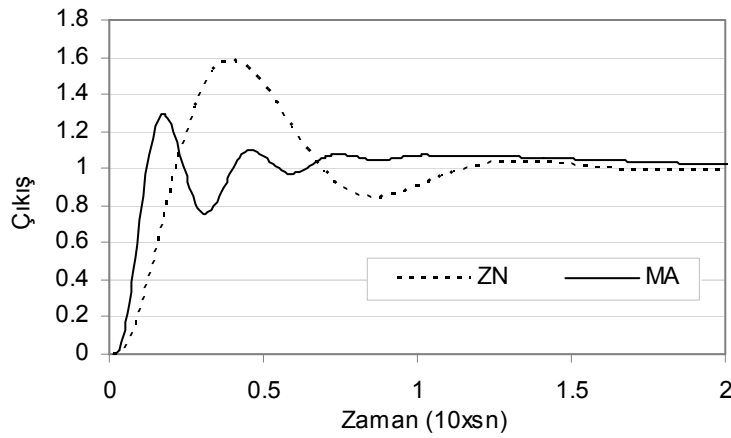
Parametre	Z-N	ITAE	IAE	ISE
K_c	6	9.9659	9.9634	5.6879
T_d	0.79	1.1841	1.5545	3.7542
T_i	3.14	3.7022	3.8553	6.2785
Yükselme zamanı (sn)	2.1	1.3	1.16	1.07
Maksimum taşma (%)	58.09	48.35	43.81	29.20
Oturma zamanı (sn)	14.8	7.7	9.2	19.9



(a) ITAE



(b) IAE



(c) ISE

Şekil 6.1. Sistem 1 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.

Tablo 6.1'den de görüldüğü gibi denetleyici tasarımında kullanılan performans indislerinden ISE de küçük yükselme zamanı, en küçük taşma miktarı ve en büyük oturma zamanı gerçekleşmiştir. Memetik algoritma ile gerçekleştirilen denetleyici tasarımlarında performans indislerinin tamamında yükselme zamanı ve maksimum taşma oranı Z-N'ye göre daha iyi sonuçlar elde edilmiştir. Yaklaşımların oturma zamanı performansları incelendiğinde memetik algoritma ile ITAE, IAE hata fonksiyonları dikkate alınarak gerçekleştirilen tasarımlarda Z-N'ye göre daha iyi sonuçlar sağladığı görülmektedir. Memetik algoritma ve Z-N ile denetleyici tasarımı gerçekleştirilen sistemlerin basamak cevapları Şekil 6.1'de verilmiştir.

Önerilen yaklaşımın performansını değerlendirmek amacıyla memetik algoritma ile elde edilen sonuçlar, Griffin'in genetik algoritma kullanarak bulduğu sonuçlarla karşılaştırılmıştır [71]. Sonuçlar Tablo 6.2'de sunulmuştur.

Tablo 6.2. Z-N, GA, MA için geçici durum parametreleri

Metot		Yükselme Zamanı (sn)	Maksimum taşma (%)	Oturma Zamanı (sn)
Z-N		2.10	58.15	14.80
GA [71]	ITAE	1.30	57.19	8.20
	IAE	1.20	44.97	9.30
	ISE	1.20	28.80	20.40
MA	ITAE	1.30	48.35	7.70
	IAE	1.16	43.81	9.20
	ISE	1.07	29.20	19.90

Tablodaki değerler incelendiğinde memetik algoritma ile tasarlanan sistemin yükselme zamanı, maksimum taşma ve oturma zamanı parametrelerinin GA ile elde edilen sonuçlara göre daha iyi olduğu görülmektedir.

Sistem 2:

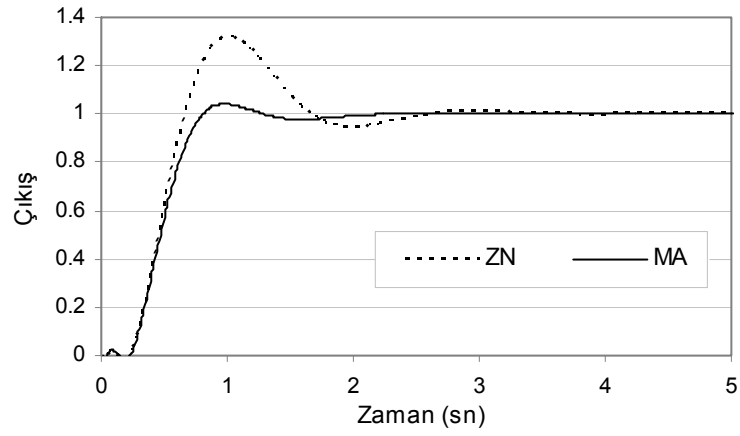
Dikkate alınan ikinci sistem transfer fonksiyonu Denklem 6.2’de verilen zaman gecikmeli ve ikinci dereceden bir sistemdir. Zaman gecikmesi içeren sistemler giriş işaretine belirli bir zaman geçmeden yanıt veremez. Zaman gecikmeli sistemlerin transfer fonksiyonları rasyonel olmadığı için sistemlerin işlenmesi daha zordur. Ölü zaman e^{-ts} içeren sistemlerin transfer fonksiyonu rasyonel fonksiyonlarla yaklaşık ifade etmenin birçok yolu vardır. Örnek sistemimizde Pade açılımı kullanılarak ölü zamanlı sistem rasyonel fonksiyonlarla ifade edilmiştir.

$$G_p(s) = \frac{e^{-0.5s}}{(s+1)^2} \quad (6.2)$$

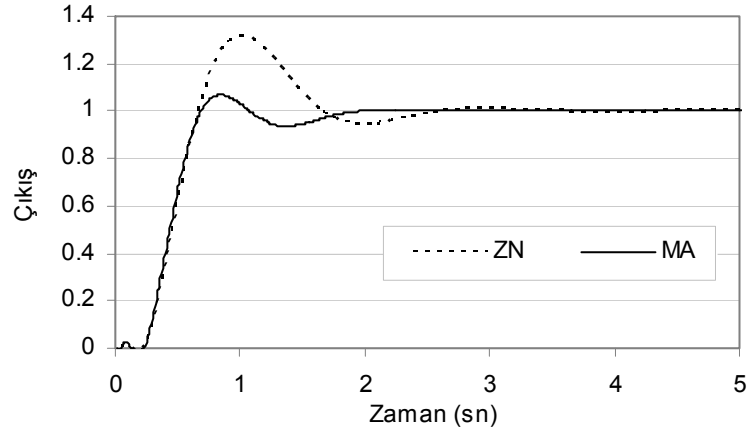
Gerçekleştirilen memetik algoritma modeli ile optimum değerleri bulunması hedeflenen K_p, K_d, K_i parametreleri $[K_{p \min}, K_{p \max}]$, $[K_{d \min}, K_{d \max}]$, $[K_{i \min}, K_{i \max}]$ değer aralığında araştırılmıştır. Bu aralıkların oluşturulması için aşağıdaki denklemlerden faydalanılmıştır [75]. Tablo 6.3’de memetik algoritma ile farklı hata fonksiyonları dikkate alınarak elde edilen sonuçlar ve Z-N yöntemi ile bulunan sonuçlar verilmiştir.

$$\begin{aligned} K_{p \min} &= 0.32K_u & K_{p \max} &= 0.6K_u \\ K_{d \min} &= 0.08K_u T_u & K_{d \max} &= 0.15K_u T_u \\ K_{i \min} &= K_{p \min}^2 / 3K_{d \min} & K_{i \max} &= K_{p \max}^2 / 3K_{d \min} \end{aligned}$$

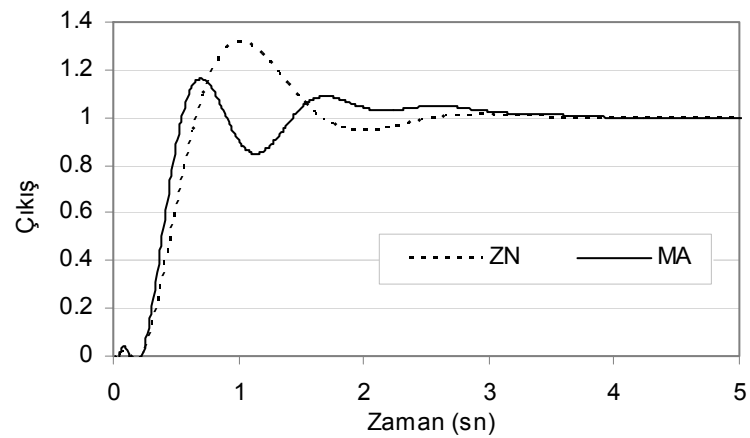
Tablo 6.3’den de görüldüğü gibi ikinci örnek sistem tasarımında Ziegler-Nichols ve diğer performans indisleri yükselme zamanı için yakın değerler vermiştir. Maksimum taşma ve oturma zamanına dikkate alındığında ITAE performans indisi kullanılarak gerçekleştirilen tasarım diğer tasarımlara göre çok daha iyi sonuç vermiştir. MA ve Z-N ile denetleyici tasarımı gerçekleştirilen sistemlerin basamak cevapları Şekil 6.2’de verilmiştir.



(a) ITAE



(b) IAE



(c) ISE

Şekil 6.2. Sistem 2 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.

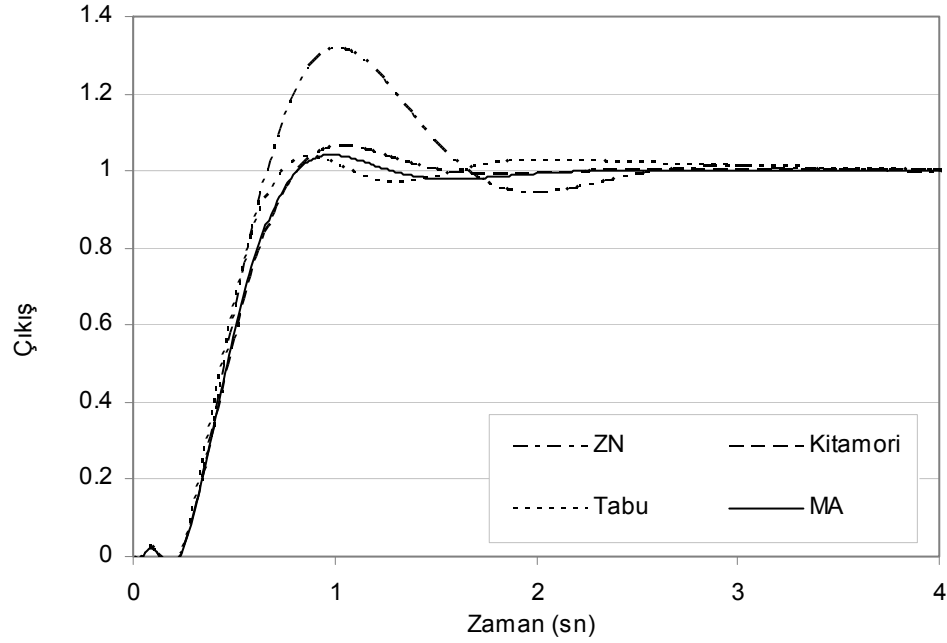
Tablo 6.3. Sistem 2 için Z-N ve MA ile belirlenen tasarım sonuçları

Parametre	Z-N	ITAE	IAE	ISE
K_p	2.808	2.2091	2.3985	2.4027
K_d	1.151	1.2280	1.4965	2.1718
K_i	1.7122	1.0590	1.1233	1.5805
Yükselme zamanı (sn)	0.74	0.82	0.42	0.54
Maksimum taşma (%)	31.72	4.01	6.9	16.32
Oturma zamanı (sn)	4.78	3.36	3.44	6.2

Önerilen yaklaşımın performansını değerlendirmek amacıyla memetik algoritma ile elde edilen sonuçlar, Karaboğa ve Kalınlı'nın tabu araştırma algoritması kullanarak bulduğu sonuçlarla karşılaştırılmıştır [64]. Sistem 2 için oturma zamanı için müsaade edilebilir tolerans değeri nihai değerin %2'lik ve %5'lik aşma değerleri için ayrı ayrı değerlendirilmiştir. Sonuçlar Tablo 6.4'de sunulmuş ve bu kazanç değerlerine ait birim basamak cevapları Şekil 6.3'de verilmiştir.

Tablo 6.4. Sistem 2 için tasarım sonuçlarının karşılaştırılması

Parametreler	Z-N	Kitamori [64]	Tabu [64]	MA
K_p	2.808	2.212	2.194	2.2091
K_d	1.151	1.148	1.465	1.2280
K_i	1.7122	1.0858	1.209	1.0590
ITAE	79.79	33.25	44.04	31.70
Yükselme zamanı (sn)	0.74	0.96	0.82	0.92
Maksimum Taşma (%)	31.72	6.36	3.62	4.01
Oturma Zamanı %5 için (sn)	4.24	2.34	1.34	1.46
Oturma Zamanı %2 için (sn)	4.78	2.72	5.04	3.34



Şekil 6.3. Farklı denetleyiciler için Sistem 2'nin basamak cevapları

Yapılan denemelerde elde edilen sonuçlar karşılaştırıldığında memetik algoritmasının ITAE hata değerlendirmesinde en küçük değeri elde ettiği görülmüştür. Maksimum taşma yüzdesi dikkate alındığında memetik algoritma, tabu araştırma algoritmasının ardından ikinci sırada optimum çözümü vermiştir.

Sistem 3:

Denetleyici tasarımı gerçekleştirilecek üçüncü sistem Denklem 6.3'de verilen üçüncü dereceden zaman gecikmesi içermeyen bir sistemdir. Hata değerleri hesaplanırken örnekleme periyodu 0.02sn, değerlendirme zaman aralığı [0-10] sn alınmıştır. K_p , K_i , K_d parametre değerleri için araştırma uzayı olarak [0, 30.32] değer aralığı kullanılmıştır. Tablo 6.5'de memetik algoritma ile farklı hata fonksiyonları dikkate alınarak elde edilen sonuçlar ve Z-N yöntemi ile bulunan sonuçlar verilmiştir.

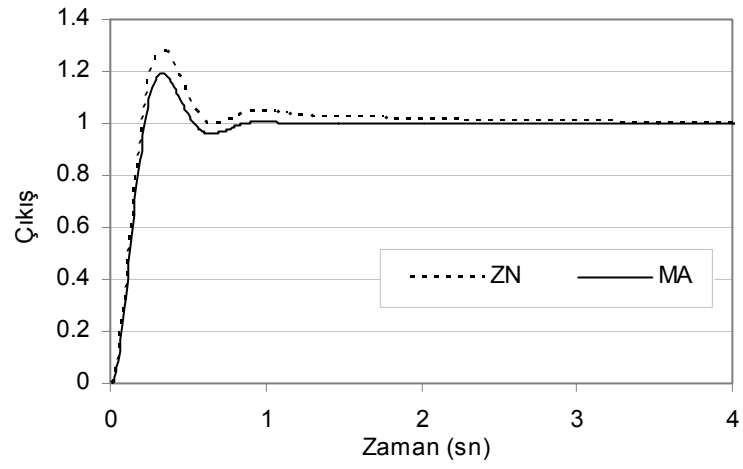
$$G_p(s) = \frac{1}{s(s+1)(s+5)} \quad (6.3)$$

Tablo 6.5. Sistem 3 için Z-N ve MA ile belirlenen tasarım sonuçları

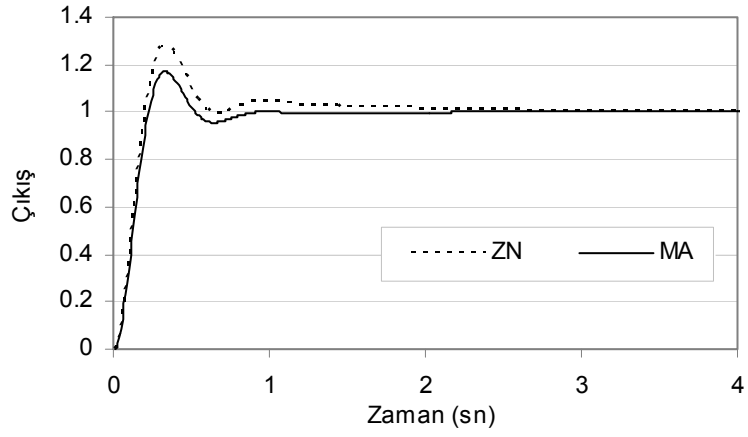
Parametreler	Geliştirilmiş Z-N	ITAE	IAE	ISE
K_p	39.42	28.8208	25.5031	19.9497
K_i	12.81	0.001	0.0083	0.3563
K_d	30.32	29.978	29.9927	30.32
Yükselme zamanı (sn)	0.30	0.30	0.34	0.32
Maksimum taşma (%)	28.03	19.19	17.05	13
Oturma zamanı (sn)	4.78	1.6	1.64	1.8

Tablo 6.5’den de görüldüğü gibi üçüncü örnek sistem tasarımında geliştirilmiş Ziegler-Nichols ve diğer performans indisleri yükselme zamanı için yakın değerler vermiştir. Maksimum taşma zamanına dikkate alındığında ISE performans indisi kullanılarak gerçekleştirilen tasarım diğer tasarımlara göre çok daha iyi sonuç vermiştir. Oturma zamanı dikkate alındığında ise ITAE ve IAE performans indisleri en iyi değerleri vermiştir. MA ve Z-N ile denetleyici tasarımı gerçekleştirilen sistemlerin basamak cevapları Şekil 6.4’de verilmiştir.

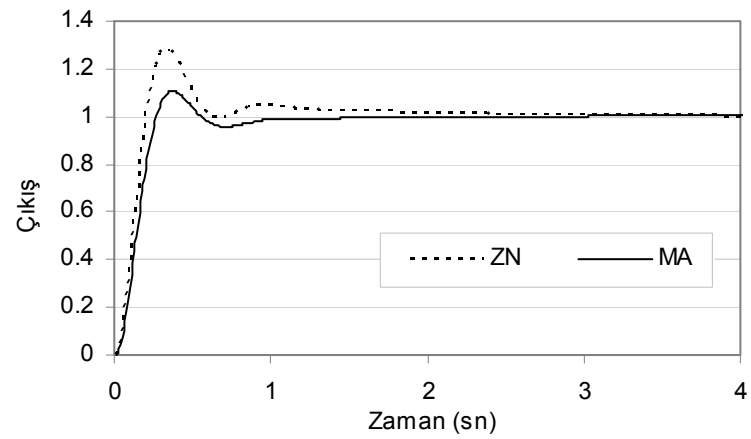
Önerilen yaklaşımın performansını değerlendirmek amacıyla memetik algoritma ile elde edilen sonuçlar, Amaral ve arkadaşlarının yapay bağışıklık algoritmasının bir türü olan Clonal Selection (CS) algoritması ile ISE hata indisini dikkate alarak gerçekleştirdikleri tasarıma ait sonuçlarla karşılaştırılmıştır [76]. Sonuçlar Tablo 6.5’de sunulmuş ve bu kazanç değerlerine ait birim basamak cevapları Şekil 6.5’de verilmiştir.



(a) ITAE



(b) IAE

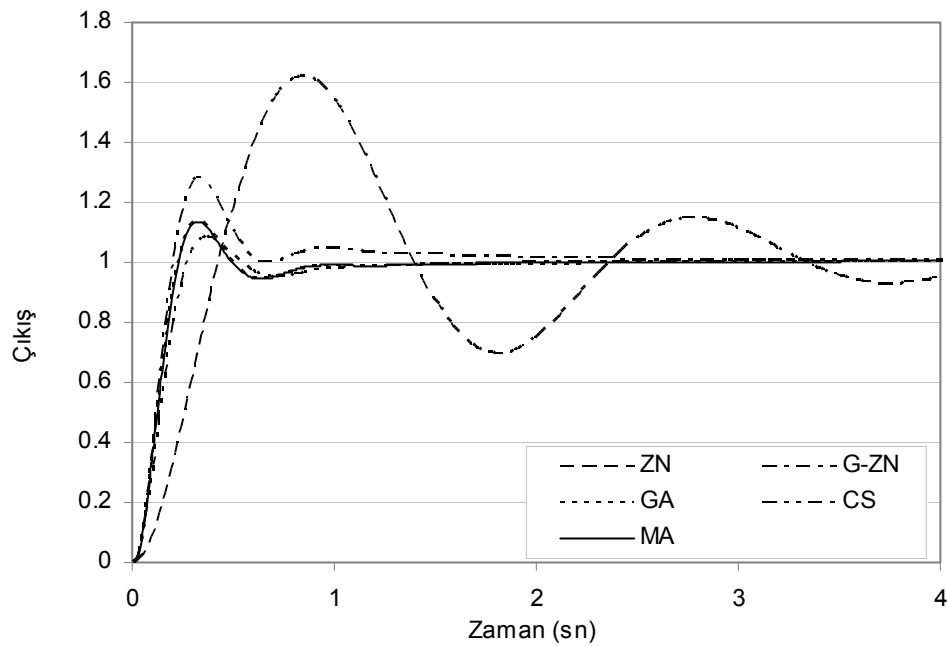


(c) ISE

Şekil 6.4. Sistem 3 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.

Tablo 6.6. Sistem 3 için tasarım sonuçlarının karşılaştırılması

Parametre	Z-N	Geliştirilmiş Z-N [76]	GA [76]	CS [76]	MA
K_p	18	39.42	20.17	15.57	19.95
K_i	12.81	12.81	0	0.33	0.3563
K_d	6.32	30.32	30.32	24.82	30.32
ISE	43.05	10.32	9.46	10.50	9.45
Yükselme Zamanı (sn)	0.68	0.30	0.36	0.40	0.32
Maksimum Taşma (%)	61.83	28.03	13.6	8.36	13.5
Oturma Zamanı %1 için (sn)	>10	4.78	2.78	2.96	2.52
Oturma Zamanı %2 için (sn)	>10	3.44	1.68	1.98	1.8



Şekil 6.5. Farklı denetleyiciler için Sistem 3'ün basamak cevapları

Sistem 3 için sonuçlar karşılaştırıldığında GA, CS ve MA ile elde edilen sonuçların Ziegler-Nichols ve geliştirilmiş Ziegler-Nichols'dan daha iyi sonuç verdiği

görülmektedir. Maksimum taşma ve %1'lik hata toleransı ile hesaplanan oturma zamanında memetik algoritma ile gerçekleştirilen tasarımın diğer yöntemlere göre daha iyi performans verdiği görülmektedir. Genel olarak dikkate alınan bu sistemin nispeten kolay olması nedeniyle GA, CS ve MA ile elde edilen sonuçlar bir birine yakın olmakla beraber, MA ile sağlanan ISE hata değeri daha düşük gerçekleşmiştir.

Sistem 4:

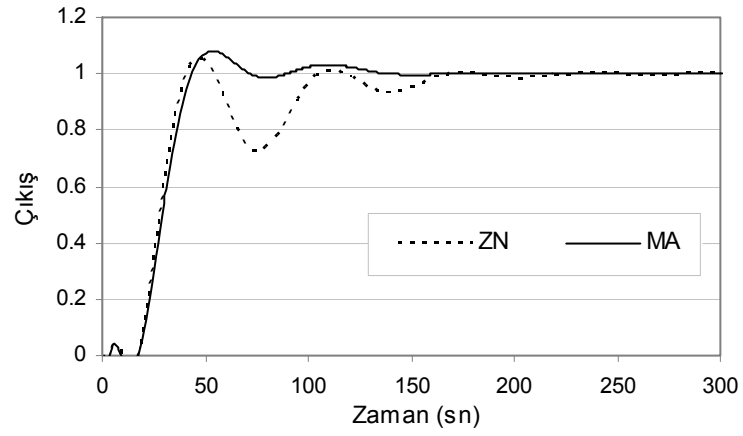
Dördüncü sistem örneği transfer fonksiyonu Denklem 6.4'te verilen ikinci dereceden geniş zaman gecikmesi içeren bir sistemdir. Tablo 6.7'de memetik algoritma ile farklı hata fonksiyonları dikkate alınarak elde edilen sonuçlar ve Z-N yöntemi ile bulunan sonuçlar verilmiştir.

$$G_p(s) = \frac{0.33e^{-18.25s}}{134s^2 + 18.5s + 1} \quad (6.4)$$

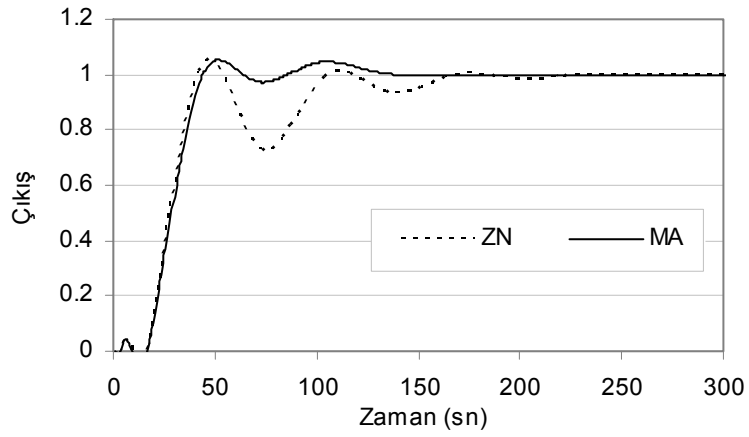
Tablo 6.7'den de görüldüğü gibi dördüncü örnek sistem tasarımında yükselme zamanı ve oturma zamanı için en iyi performans ISE hata indisiyle gerçekleştirilen tasarımda gerçekleşirken, en iyi taşma yüzdesi IAE hata indisiyle yapılan tasarımda elde edilmiştir. MA ve Z-N ile denetleyici tasarımı gerçekleştirilen sistemlerin basamak cevapları Şekil 6.6'da verilmiştir.

Tablo 6.7. Sistem 4 için Z-N ve MA ile belirlenen tasarım sonuçları

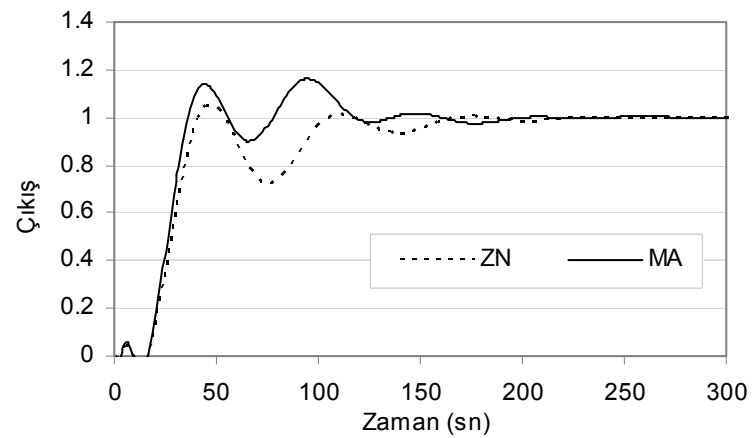
Parametre	Z-N	ITAE	IAE	ISE
K_p	2.901	2.3815	2.3405	2.4786
K_i	0.080	0.1142	0.1132	0.1410
K_d	26.43	25.0527	26.7864	39.5942
Yükselme zamanı (sn)	21	23	23	16
Maksimum taşma (%)	5.29	8.11	5.22	16.2
Oturma zamanı (sn)	159	121	122	114



(a) ITAE



(b) IAE



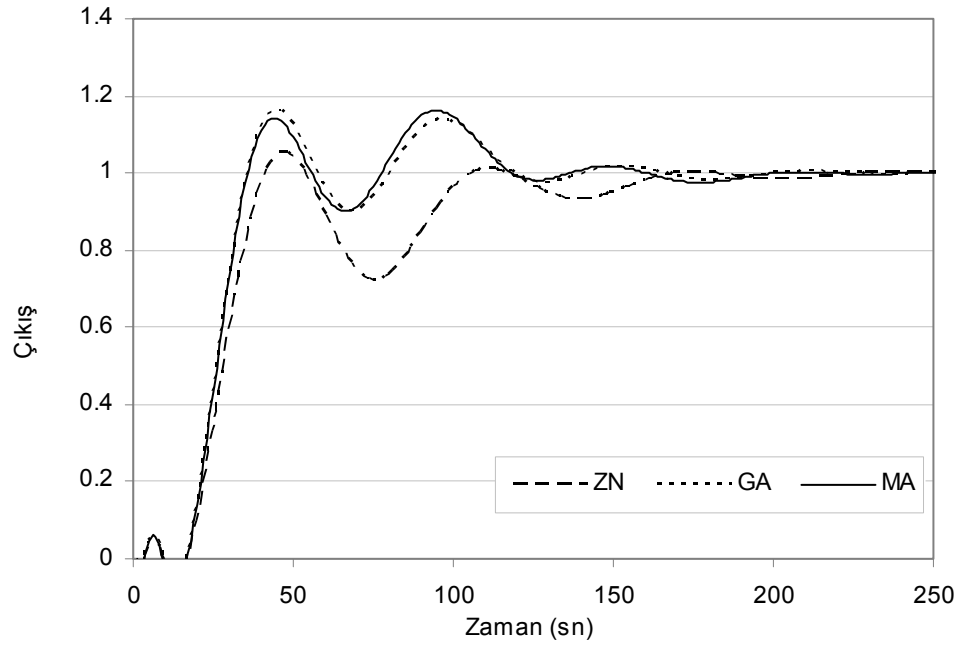
(c) ISE

Şekil 6.6. Sistem 4 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.

Önerilen yaklaşımın performansını değerlendirmek amacıyla memetik algoritma ile elde edilen sonuçlar, Wang ve Kwork'un geliştirdikleri GA yaklaşımıyla elde ettikleri sonuçlarla karşılaştırılmıştır [77]. Sonuçlar Tablo 6.8'de sunulmuş ve bu kazanç değerlerine ait birim basamak cevapları Şekil 6.7'de verilmiştir.

Tablo 6.8. Sistem 4 için tasarım sonuçlarının karşılaştırılması

Parametre	Z-N	GA [77]	MA
K_p	2.901	2.612	2.4786
K_i	0.080	0.139	0.1410
K_d	26.43	38.23	39.5942
Yükselme Zamanı (sn)	21	17	16
Maksimum Taşma (%)	5.29	16.4	16.2
Oturma Zamanı (sn)	159	134	114



Şekil 6.7. Farklı denetleyiciler için Sistem 4'ün basamak cevapları

Yapılan denemelere ait sonuçlar karşılaştırıldığında, en iyi yükselme zamanı, oturma zamanı ve maksimum taşma yüzdesi değerlerinin MA ile elde edildiği görülmektedir.

Sistem 5:

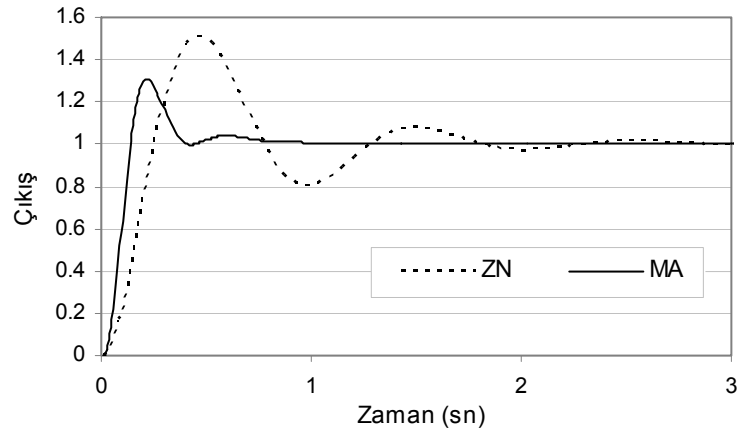
Denetleyici tasarımı gerçekleştirilecek beşinci sistem transfer fonksiyonu Denklem 6.5’de verilen dördüncü dereceden zaman gecikmesi içermeyen bir sistemdir. Tablo 6.9’da memetik algoritma ile farklı hata fonksiyonları dikkate alınarak elde edilen sonuçlar ve Z-N yöntemi ile bulunan sonuçlar verilmiştir.

$$G_p(s) = \frac{10000}{s^4 + 126s^3 + 2725s^2 + 12600s + 10000} \quad (6.5)$$

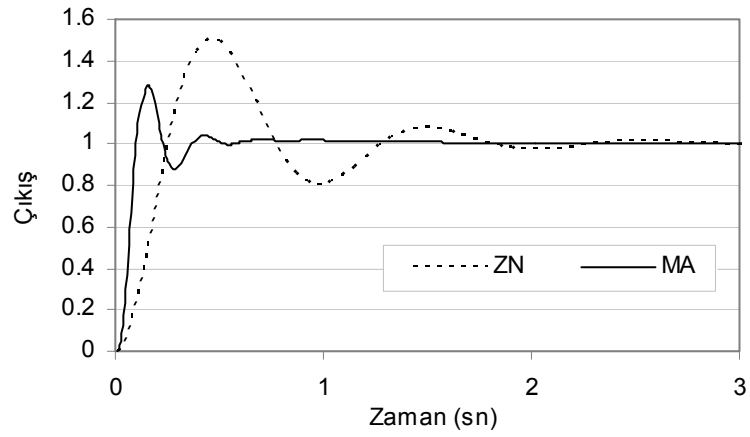
Tablo 6.9’dan da görüldüğü gibi beşinci örnek sistem tasarımında yükselme zamanı en iyi performans ISE hata indisiyle gerçekleştirilen tasarımla gerçekleşirken maksimum taşma yüzdesi IAE hata indisiyle yapılan tasarımda elde edilmiştir. Tüm tasarımlar incelendiğinde ITAE, IAE, ISE hata indisleriyle elde edilen sonuçlar yükselme zamanı, maksimum taşma ve oturma zamanında Ziegler-Nichols’dan daha etkin olduğu görülmüştür. MA ve Z-N ile denetleyici tasarımı gerçekleştirilen sistemlerin basamak cevapları Şekil 6.8’de verilmiştir.

Tablo 6.9. Sistem 5 için Z-N ve MA ile belirlenen tasarım sonuçları

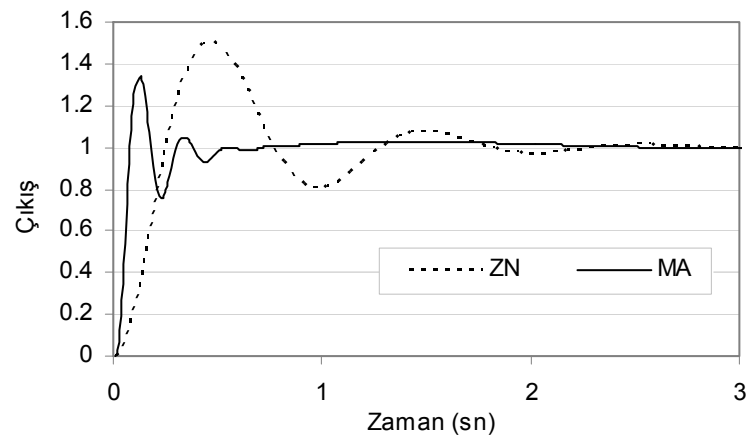
Parametre	Z-N	ITAE	IAE	ISE
K_c	11.1524	24.1541	23.1219	15.8671
T_d	0.0811	0.1409	0.2659	0.5982
T_i	0.3244	0.4924	0.6398	0.4350
Yükselme zamanı (sn)	0.18	0.10	0.08	0.06
Maksimum taşma (%)	50.43	30.61	27.92	33.77
Oturma zamanı (sn)	2.16	0.72	0.72	1.84



(a) ITAE



(b) IAE



(c) ISE

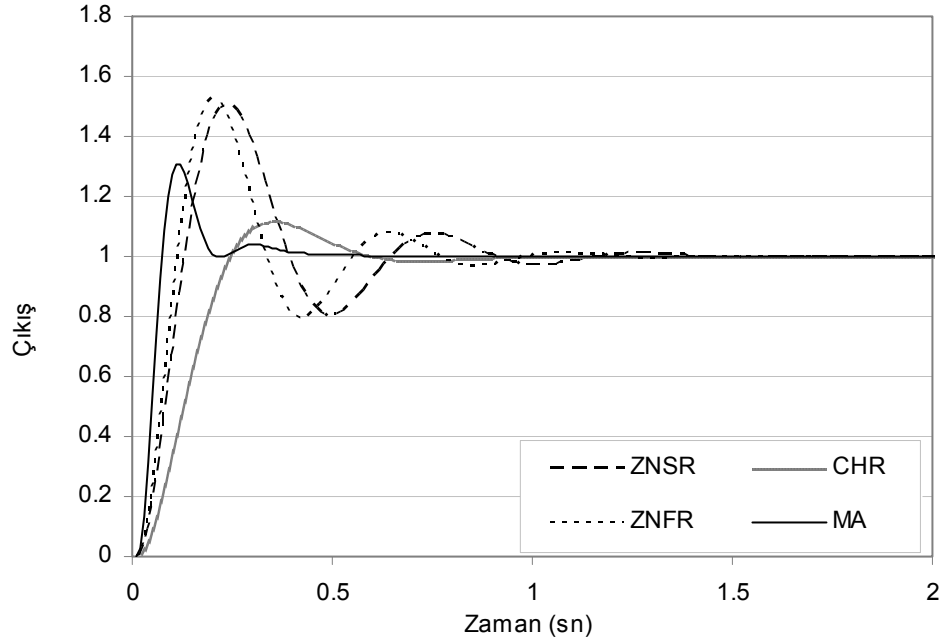
Şekil 6.8. Sistem 5 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.

Gerçekleştirilen memetik algoritma modelinin performansını değerlendirmek amacıyla ITAE hata fonksiyonu dikkate alınarak memetik algoritma ile elde edilen neticeler, kaynak [78]'deki sonuçlarla karşılaştırılmıştır. İlgili kaynakta belirtilen çalışmada aynı sistem için Ziegler-Nichols birim basamak cevabı (ZNSR), Ziegler-Nichols frekans alanı cevabı (ZNFR) ve Chein-Hrones-Reswick (CHR) ile elde edilen sonuçlara yer verilmiştir. Sonuçlar Tablo 6.10'da sunulmuş ve bu kazanç değerlerine ait birim basamak cevapları Şekil 6.9'da verilmiştir.

Tablo 6.10. Sistem 5 için tasarım sonuçlarının karşılaştırılması

Parametre	ZNSR [78]	CHR [78]	ZNFR [78]	MA
K_c	11.1524	5.5762	14.496	24.154
T_d	0.0811	0.0811	0.08	0.1409
T_i	0.3244	1.0978	0.32	0.4924
ITAE	11.32	4.12	8.51	1.12
Yükselme Zamanı (sn)	0.18	0.36	0.16	0.10
Maksimum Taşma (%)	50.43	11.8	52.65	30.61
Oturma Zamanı (sn)	2.16	1.1	1.84	0.72

Sonuçlar diğer metotlarla kıyaslandığında yükselme zamanı ve oturma zamanı için en iyi değerler gerçekleştirilen memetik algoritma modeliyle elde edilmiştir. CHR modeli ile düşük maksimum taşma yüzdesi sağlanmış olmasına rağmen, bu metot ile sağlanan çözümün yükselme zamanı ve oturma zamanı değerleri oldukça yüksektir.



Şekil 6.9. Farklı denetleyiciler için Sistem 5'in basamak cevapları

Sistem 6:

Denetleyici tasarımı gerçekleştirilecek son örnek transfer fonksiyonu Denklem 6.6'da verilen dördüncü dereceden zaman gecikmesi içermeyen bir sistemdir. Tablo 6.11'de memetik algoritma ile farklı hata fonksiyonları dikkate alınarak elde edilen sonuçlar ve Z-N yöntemi ile bulunan sonuçlar verilmiştir.

$$G_p(s) = \frac{1}{(1+s)(1+s/6)^3} \quad (6.6)$$

Tablo 6.11'den de görüldüğü gibi sistem tasarımında yükselme zamanı için en iyi performans ISE hata indisiyle gerçekleştirilen tasarımla gerçekleşirken, optimum taşıma yüzdesi ITAE hata indisiyle yapılan tasarımda elde edilmiştir. Tüm tasarımlar incelendiğinde MA ile ITAE, IAE, ISE hata fonksiyonları dikkate alınarak elde edilen sonuçların Ziegler-Nichols'dan daha başarılı olduğu görülmektedir. MA ve Z-N ile denetleyici tasarımı gerçekleştirilen sistemlerin basamak cevapları Şekil 6.10'da verilmiştir.

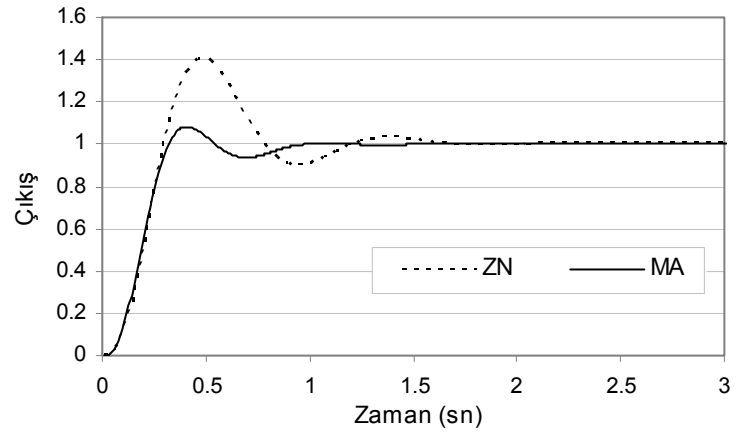
Tablo 6.11. Sistem 6 için Z-N ve MA ile belirlenen tasarım sonuçları

Parametre	Z-N	ITAE	IAE	ISE
K_p	4.605	3.8954	4.2568	4.2631
K_i	5.904	2.7870	1.6692	3.3077
K_d	0.847	1.1993	3.1577	6.7596
Yükselme zamanı (sn)	0.50	0.44	0.36	0.22
Maksimum taşma (%)	40.47	8.08	10.09	23.44
Oturma zamanı (sn)	3	1.74	1.54	3.34

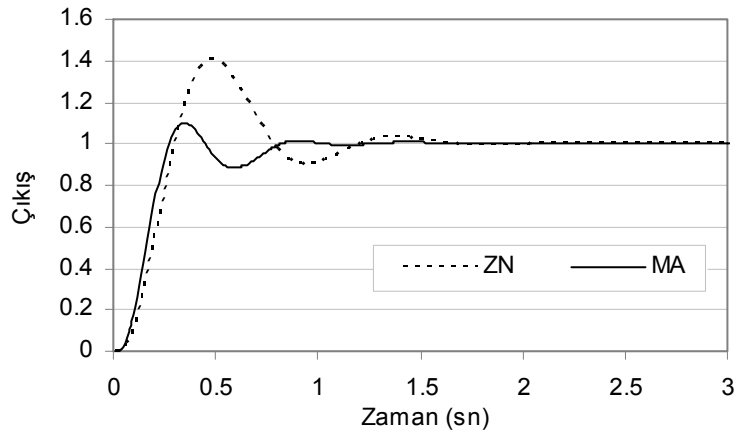
Önerilen yaklaşımın performansını değerlendirmek amacıyla memetik algoritma ile elde edilen sonuçlar, Wang ve Kwork'un geliştirdikleri GA yaklaşımıyla elde ettikleri sonuçlarla karşılaştırılmıştır [77]. Sonuçlar Tablo 6.12'de sunulmuş ve bu kazanç değerlerine ait birim basamak cevapları Şekil 6.11'de verilmiştir.

Tablo 6.12. Sistem 6 için tasarım sonuçlarının karşılaştırılması

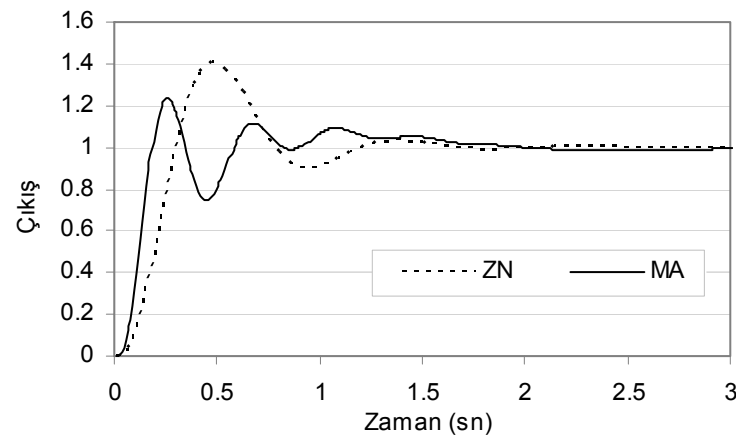
Parametre	Z-N	GA [77]	MA
K_p	4.605	4.244	4.2631
K_i	5.904	6.576	6.7596
K_d	0.847	3.201	3.3077
ISE	18.93	10.90	10.89
Yükselme Zamanı (sn)	0.5	0.24	0.22
Maksimum Taşma (%)	40.47	22.42	23.44
Oturma Zamanı (sn)	3	3.36	3.34



(a) ITAE

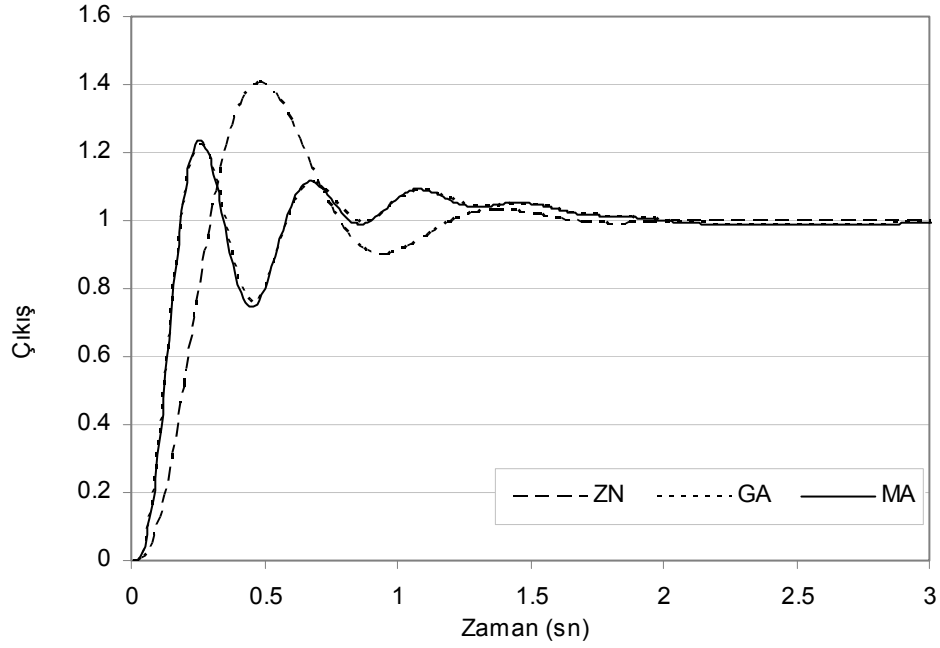


(b) IAE



(c) ISE

Şekil 6.10. Sistem 6 için Z-N ve MA ile gerçekleştirilen tasarımların (a) ITAE, (b) IAE ve (c) ISE hata fonksiyonları için birim basamak cevapları.



Şekil 6.11. Farklı denetleyiciler için Sistem 6'nın basamak cevapları

Yapılan benzetim çalışmalarından elde edilen sonuçları incelendiğinde bu sistem için de memetik algoritma ile sağlanan çözümlerin diğer yöntemlere göre daha başarılı olduğu görülmektedir.

Benzetim çalışmalarında dikkate alınan 6 sistem için elde edilen sonuçlar incelendiğinde tasarlanan denetleyici sistemlerin performanslarının kullanılan hata fonksiyonlarına göre değişiklik gösterdiği görülmektedir. Örneğin, ISE hata değeri dikkate alınarak gerçekleştirilen bir tasarımda en iyi yükselme zamanı değeri sağlanırken, bu sistemin maksimum taşma yüzdesi değeri büyük olabilmektedir. Genel olarak, tasarım parametrelerinin tamamında en başarılı sonuçları sağlayan bir hata fonksiyonunun olmadığı söylenebilir. Dolayısıyla, tasarımcının denetleyici tasarlanacak sistemin yapısına uygun hata fonksiyonunu belirlemesi gerekmektedir. Literatürde yaygın olarak kullanılan hata indislerinin dışında tasarlanmak istenen sistemin özelliğine göre farklı hata fonksiyonları da oluşturulabilir. Benzetim çalışmalarında elde edilen sonuçlardan sezgisel yaklaşımlarla elde edilen tasarımların klasik metotlarla elde edilenlere göre başarımlarının daha yüksek olduğu söylenebilir. Bu çalışmada, önerilen memetik algoritma yaklaşımı ile sağlanan sonuçların hem klasik metotlara hem de diğer dikkate alınan sezgisel metotlara göre daha başarılı olduğu görülmüştür.

7. BÖLÜM

SONUÇLAR

Bu çalışmada ilk olarak, tabii seleksiyon ile canlılarda bulunan genetik gelişimi simüle eden ve evrim kurallarına dayanarak geliştirilen genetik algoritma ile katıların fiziksel tavlama işlemini simüle eden benzetilmiş tavlama algoritmasının birlikte kullanılmasıyla bir memetik algoritma modeli gerçekleştirilmiştir. Gerçekleştirilen memetik algoritma modelinin sürekli test fonksiyonlarının optimizasyonundaki başarımının incelenmesi amacıyla literatürde yaygın olarak kullanılan yedi adet test fonksiyonu dikkate alınarak benzetim çalışmaları gerçekleştirilmiştir. Benzetim çalışmalarından elde edilen sonuçlar, literatürde yer alan genetik algoritma, tabu araştırma algoritması, paralel tabu araştırma algoritması, turlayan karınca koloni optimizasyon algoritması ve paralel karınca koloni optimizasyon algoritmalarına ait sonuçlarla karşılaştırılmıştır. Elde edilen sonuçlar, gerçekleştirilen memetik algoritma modelinin sürekli test fonksiyonlarının optimizasyonundaki başarımının oldukça yüksek olduğunu göstermiştir.

Tez çalışmasının ikinci aşamasında, endüstriyel süreçlerin denetiminde yaygın olarak kullanılan PID denetleyicilerin tasarımı amacıyla memetik algoritmaya dayalı bir yaklaşım önerilmiştir. Önerilen yaklaşımla değişik özelliklere sahip çeşitli sistemler için PID denetleyici tasarımları gerçekleştirilerek, elde edilen sonuçlar literatürde yer alan klasik ve sezgisel algoritmalara dayalı diğer bazı metotlara ait sonuçlarla karşılaştırılmıştır.

Karmaşık ve nispeten zor olan sistemlerin tasarımında Ziegler-Nichols gibi klasik yaklaşımların sağladığı tasarımların başarımı çoğunlukla düşük olabilmekte ve bu tür yöntemler genellikle karmaşık matematiksel işlemlerin gerçekleştirilmesini gerektirmektedir. Ayrıca klasik tekniklerin uygulanması çoğunlukla uzun işlem süreleri

gerektirebilmektedir. Benzetim çalışmalarından elde edilen sonuçlardan da görülebileceği gibi, memetik algoritma yaklaşımı daha hızlı ve daha başarılı tasarım sonuçları sağlamaktadır. Ayrıca, Ziegler-Nichols gibi klasik yöntemler çoğunlukla tasarım parametreleri için yaklaşık değerler sağlamakta ve uygulamada parametrelerin manüel olarak ayarlanması gerekebilmektedir. PID tasarımında memetik algoritmanın kullanılması, hızlı bir şekilde oldukça yüksek doğrulukta parametre değerleri sağlamanın yanı sıra bu tür manüel ayarlamalar yapılmasına da ihtiyaç duymayarak tam bir tasarım otomasyonu sağlamaktadır. Elde edilen sonuçların literatürde yer alan diğer bazı sezgisel metotlara ait sonuçlarla karşılaştırılmasından, önerilen memetik algoritma yaklaşımına genel olarak dikkate alınan diğer yöntemlere göre daha başarılı neticeler sağladığı da görülmüştür.

Memetik algoritmaların iteratif ve popülasyon tabanlı algoritmaların avantajlarını birleştirmesi araştırma uzayının etkin bir şekilde taranması ve daha başarılı neticelerin elde edilmesinde bir avantaj olarak ortaya çıkarken, algoritmanın çok sayıda kontrol parametresi içermesi bir dezavantaj olarak gözükmektedir. Uygulamada bu parametrelerin uygun değerlerde belirlenmesi algoritmanın performansı üzerinde son derece etkili olmaktadır. Tez çalışmasında yapılan birçok deneme neticesinde uygun değerler belirlenmeye çalışılmıştır. Memetik algoritmaya ait parametrelerin uygun şekilde belirlenmesi, algoritmanın performansının artırılmasına yönelik yeni stratejilerin algoritmaya entegre edilmesi, farklı şekillerde oluşturulan memetik algoritma modellerinin PID denetleyici tasarımındaki performanslarının incelenmesi ve memetik algoritmanın diğer bazı bilgisayar mühendisliği problemlerinin çözümüne uygulanması gelecekte yapılabilecek çalışmalar olarak düşünülebilecektir.

KAYNAKLAR

1. Harrel, D., Algorithmics, The Spirit of Computing, Addison-Wesley, 1987.
2. Ogata, K. Modern Control Engineering (second ed.). Prentice-Hall, Inc., USA, 1990.
3. Pham, D.T., and Karaboga, D., Intelligent Optimisation Techniques, Genetic Algorithms, Tabu Search, Simulated Annealing And Neural Networks, Advanced Manufacturing Series, Springer-Verlag, London, 2000.
4. Shang, Y., Global Search Methods for Solving Nonlinear Optimization Problems, University of Illinois at Urbana Champaign, 1997.
5. Spaulding, K.A., Natural Metaphoric Optimization Algorithms, Master of science Engineering, The University of Texas at Austin, December 1998.
6. Waner, S., Costenoble, S.R., On-Line Tutorials for Finite Mathematics Applied Calculus Finite Mathematics & Applied Calculus, http://people.hofstra.edu/faculty/Stefan_Waner/RealWorld/tutindex.html, 2001.
7. Poyla, G., How to Solve it, Princeten University Pres, Princeton, 1948.
8. Osman, I.O., Heuristics for combinatorial optimization problems: Development and new directions, Proceedings of the first seminar on information technology and applications, Marfield Conf. Center, Leicester, U.K., 29 Eylül, 1-23, 1991.
9. Holland, J.H., Adaption in Natural and Artificial Systems, MAMIT Press, Cambridge, 1975.
10. Goldberg, D.E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Readin, MA, 1989.
11. Glover, F., Tabu Search – Part I, ORSA Journal on Computing, 1 (3), 190-206, 1989.
12. Glover, F., Tabu Search – Part II, ORSA Journal on Computing, 2 (1), 4-32, 1990.
13. Kirckpatric, S., Gelatt, C.D., and Vecchi, J.M.P., Optimization by Simulated Annealing, Science, 220, 671-680, 1983.
14. Dorigo, M., Maniezzo, V. and Colorni, A., Positive Feedback as a Search Strategy. Teknik Rapor No. 91-016, Politecnico di Milano, Italy, 1991.

15. Colorni, A., Dorigo, M., Maniezzo, V., Distributed Optimization by Ant Colonies, Proceedings of the first European Conference on Artificial Life, Paris, France, F.Varela and P.Bourgine (Eds.), Elsevier Publishing, 134-142, 1991.
16. Hirayasu, T., Miki, M., Ono, Y., Ant Colony Optimization for Continuous Problems Doshisha University, Dept. of Knowledge Engineering and Computer Science 1-3 Tatara Miyakodani, Kyotanabe, Kyoto 610-0321, Japan, 2000.
17. Kalınlı, A., Karaboga, N., Karaboga, D., A Modified Touring Ant Colony Optimisation Algorithm for Continuous Functions, The Sixteenth International Symposium on Computer and Information Sciences (ISCISXVI), Işık Üniversitesi, Antalya, Kasım 5-7, 2001, 94-102, 2001.
18. Sarıkoç F., Paralel Karınca Kolonisi Optimizasyon Algoritması ve Test Problemlerindeki Performansının İncelenmesi, Yüksek Lisans Tezi, Erciyes Üniversitesi, Kayseri, 2004.
19. Kartalopoulos, S.V., Understanding Neural Networks and Fuzzy Logic, IEEE Press, New York, 1996.
20. Sagirolu, S., Artificial Neural Networks in Robotic Applications, International Journal of Mathematical and Computational Applications, 3(2), 67-81, 1998.
21. Penrose, R., Kralın Yeni Usı I, Bilgisayar ve Zeka, Tübitak Popüler Bilim Kitapları, Çevirmen Tekin Dereli, Ankara, 1997.
22. An Introduction to Computational Complexity. <http://users.forthnet.gr/ath/kimon/index.htm>, 2002.
23. Complexity, P and NP. <http://cgm.cs.mcgill.ca/~cwu25/proj507/pandnp.htm>, 2003.
24. Greenberg H. J., Mathematical Programming Glossary. <http://carbon.cudenver.edu/~hgreenbe/glossary/>, 2004.
25. Corne, D., Dorigo, M., Glover, F., New Ideas in Optimization, McGraw-Hill, 1999.
26. Norman, M.G., Moscato, P., A Competitive and Cooperative Approach to Complex Combinatorial Search, Teknik Rapor, Caltech Concurrent Computation Program, Rapor 790, California, Institute of Technology, Pasadena, California, USA, 1989.

27. Cohoon, J.P. et.al., Punctuated Equilibria: A Parallel Genetic Algorithms, Genetic Algorithms and Their Applications, J.J. Grefenstette (editor), Lawrence Erlbaum Associated, 1987.
28. Cohoon, J.P. et.al., Floorplan Design Using Distributed Genetic Algorithms, Proceeding of the IEEE International Conference on Computer Aided Design, 1988.
29. Hedge, S.U., Efficacy of Parallel Genetic Algorithms, Master Thesis, University of Virginia, Charlottesville VA, Augustos, 1988.
30. Pettey, C.B. et.al. A Parallel Genetic Algorithm, Genetic Algorithms and Their Applications, J.J. Grefenstette, editor, Lawrence Erlbaum Associated, 1987.
31. Tanese, R., Distributed Genetic Algorithms, Proceedings of the Third International Conference of Genetic Algorithms, Fairfax, VA, ed. By J.D. Schaffer, Morgan Kaufmann, San Mateo CA, 434-439, 1989.
32. Burke, E.K., Newall, J.P., and Weare, R.F., A Memetic Algorithm for University Exam Timetabling, in Lecture Notes in Computer Science 1153, 241-250, 1996.
33. Cheng, R., Gen, M., Parallel Machine Scheduling Problems Using Memetic Algorithms, Computers & Industrial Engineering, 33(3-4), 761-764, 1997.
34. Merz, P. and Freisleben, B., A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem, to Appear in Proceedings of the International Congress of Evolutionary Computation (CEC'99), 1999.
35. Merz, P. and Freisleben, B., Memetic Algorithms and the Fitness Landscape of the Graph Bi-partitioning Problem, in: A-E. Eiben, T. Back, M. Schoenauer and H.-P. Schwefel (eds.), Proceedings of the 5th International Conference on Parallel Problem Solving From Nature, Lecture Notes in Computer Science, 1498, Springer, 765-774, 1998.
36. Goldberg, D.E., Zen and the Art of Genetic Algorithms, Proceedings of the Third International Conference of Genetic Algorithms, Fairfax, VA, ed. By J.D. Schaffer, Morgan Kaufmann, San Mateo CA, 80-85, 1989.
37. Goldberg, D.E., Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley, 1989.

38. Syswerda, G., Uniform Crossover in Genetic Algorithms. Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann, 2-9, 1989.
39. Greenwell, R.N., Angus, J.E., and Finck, M., Optimal Mutation Probability for Genetic Algorithms, Mathematical and Computer Modelling, 21, 1-11, 1995.
40. De Jong, K.A. An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral Thesis, Department of Computer and Communication Sciences. University of Michigan, Ann Arbor, 1975.
41. Schaffer, J.J., ve ark., A Study of Control Parameters Affecting On-line Performance of Genetic Algorithms for Function Optimisation, Proc. 3rd Int. Conf. on Genetic Algorithms and Their Applications, George Mason University, 51-61, 1989.
42. Grefenstette, J.J., Optimization of Control Parameters for Genetic Algorithms. IEEE Trans on Systems, Man and Cybernetics SMC-16, 122-128, 1986.
43. Mansfield, R.A., Genetic Algorithms, University of Wales College of Cardiff, 1990.
44. Goldberg, D.E., Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, Mass, 1989.
45. Baker, J.E Adaptive Selection Methods for Genetic Algorithms, Proc. 1st Int. Conf. Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates, Hillsdale, NJ, 101-111, 1985.
46. Mansfield, R.A., Genetic Algorithms, University of Wales College of Cardiff, 1990.
47. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E., Equation Of The State Calculation By Fast Computing Machines, Journal of Chemical Physics, 21, 1087-1092, 1953.
48. Jhonson, S., Aragon, C., Mccgeoch, L., and Schevon, C., Optimization By Simulated Annealing: An Experimental Evaluation, Part-I, Graph Partitioning, Operations Research, 37, 865-892, 1989.
49. Karaboga, D., Yapay Zeka Optimizasyon Algoritmaları, Atlas-Nobel Yayınları, 2003.
50. Dowsland, K.A., Simulated Annealing. In Modern Heuristic Techniques for Combinatorial Problems, (Editor, Reeves, C.R.), McGraw-Hill, 1995.

51. Wong, K.P., Steven, R.M., Feng, D. and Fulham M.J., Estimation of Input Function and Kinetic Parameters Using Simulated Annealing: Application in a Flow Model, *IEEE Transactions on Nuclear Science*, 49(3), 707-713, 2002.
52. Kalınlı, A., Elman Ağıının Benzetilmiş Tavlama Algoritması Kullanarak Eğitilmesi, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 19(1-2), 28-37, 2003.
53. Ciftlikli, C., Kalinli, A. ve Develi, I., An Improved Formula Obtained by Simulated Annealing for the Computation of Maximum SINR for DS-CDMA Receiver with Exponentially Weighted Despreading Function, *European Transactions on Telecommunications*, (basılacak), 2005.
54. Brown, D., Huntley, C.L. and Spillance, A., A Parallel Genetic Heuristic for the Quadratic Assignment Problem, *Proceedings of the Third International Conference of Genetic Algorithms*, Fairfax, VA, ed. By J.D. Schaffer, Morgan Kaufmann, San Mateo CA, 406-415, 1989.
55. Kalınlı, A., Karaboga, D., Trainingn Recurrent Neural Networks by Using Paralel Tabu Search Algorithm based on Crossover Operation, *Engineering Applications of Artificial Inteligence*, 17(3), 301-309, 2004.
56. Kuo, B.C., *Otomatik Kontrol Sistemleri*, Çevirmen Prof. Dr. Atill Bir, 2002.
57. Yüksel, İ, *Otomatik Kontrol Sistem Dinamiği ve Denetim Sistemleri*, Uludağ Üniversitesi Yayını, Yayın no 21, Bursa, 1997.
58. Ceylan, A.M., Çok Değişkenli Sistemlerde İleri Kontrol Algoritmalarının İncelenmesi, Yüksek Lisans Tezi, Ankara Üniversitesi, Ankara, 2001.
59. Astrom, K.J. and Hagglund, T., *PID Controllers: Theory, Design and Tuning*, (Second Edition), Instrument Society of America, 1995.
60. Sung, S.W., Lee, I.B. and Lee, B.K., On-line Process Identification and Automatic Tuning Method for PID Controllers, *Chemical Engineering Science*, 53(10), 1847-1859, 1998.
61. Chung, K.J., Kwak, H.J., Sung, S.W., Lee, I.B. and Park, J.Y., A Direct Estimation Method of Second Order Plus Time Delay Model Parameters for PID Controller Autotuning, *Journal of Chemical Engineering of Japan*, 32(3), 288-294, 1999.
62. O'Dwyer, A. and Ringwood, J.V., Convergence of Parameter and Delay Estimates Using Open Loop Time Domain Gradient Methods *Proceedings of the*

- Irish Signals and Systems Conference, Dublin, Ireland, Haziran 29-30, 434-441, 2000.
63. Ziegler, J.G. and Nichols, N.B., Optimum Setting for Automatic Controllers, Trans. ASME, 64, 759-768, 1942.
 64. Karaboga, D. and Kalinli, A., Tuning PID Controller Parameters Using Tabu Search Algorithm. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 134-136, 1996.
 65. Puangdownreong, D, Kulworawanichpong, T., Sujitorm, S., Input Weighting Optimization for PID Controllers Based on the Adaptive Tabu Search, Int. Conf. of IEEE Tencon2004, 2004.
 66. Shen, J.C., New Tuning Method for PID Controller, ISA Trans., 41(4), 473-84, Ekim, 2002.
 67. Shen, J.C., Fuzzy Neural Networks for Tuning PID Controller for Plants with Underdamped Responses, IEEE Transactions on Fuzzy Systems, 9(2), Nisan, 2001.
 68. Zhou, G. and Birdwell, J.D., Fuzzy Logic-Based PID Autotuner Design Using Simulated Annealing, Proc. of the 1994IEEE/IFAC Joint Symposium on Computer-Aided Control System Design, 67-72, Mart 7-9, 1994.
 69. Chiaberge, M., Merelo, J.J., Reyneri, L.M., Prieto, A., Zocca, L., A Comparison of Neural Networks, Linear Controllers, Genetic Algorithms and Simulated Annealing for Real Time Control, in Proc. of ESANN 94, European Symposium on Artificial Neural Networks, Brussels (B), Nisan, 1994.
 70. Alli, H., Kaya, M., Genetik Algoritma Kullanılarak PID Kontrol Parametrelerinin Bulunması, F.Ü. Fen ve Müh. Bilimleri Dergisi, 13(2), 1-8, 2001.
 71. Griffin, I., Online PID Controller Tuning Using Genetic Algorithms, M. Eng. in Electronic Systems, Thesis, Dublin City University, School of Electronic Engineering, 2003.
 72. Dandıl, B, Gökbulut, M., Ata, F., Genetik Algoritma ile Model Referans PID Denetleyici Tasarımı, F.Ü. Fen ve Müh. Bilimleri Dergisi, 14(2), 33-43, 2002.
 73. Shyr, W.J., Wang, B.W., Yeh, Y.Y., Su, T.J., Design of Optimal PID Controllers using Memetic Algorithm, Proceedings of the American Control Conference, Anchorage, Mayıs 8-10, 2002.

74. Coelho, L. Dos S., Krohling, R.A., Parameters Tuning of Multivariable Controllers Based on Memetic Algorithm, IEEE International Symposium on Intelligent Control, Canada, 2002.
75. Zhao, Z.Y., Masayoshi, T. and Isaka, S., Fuzzy Gain Scheduling of PID Controllers, IEEE Trans. on SMC, 23(5), 1392- 1398, Eylül/Ekim, 1993.
76. Amaral, J.L.M., Amaral, J.F.M., Transcheit, R., Pacheco, M., Mesquite A. Tuning Evolvable PID Controllers through a Clonal Selection Algorithm Proceedings of the NASA/DoD Conference of Evolution Hardware, 2005.
77. Wang, P., Kwork, D.P., Auto Tuning of Classical PID Controllers using an Advanced Genetic Algorithm, Proceedings of the 1992 International Conference on Industrial Electronics, Control, and Instrumentation, 3, 1224-1229, San Diego, 9.-13. Kasım 1992.
78. Paz, A.P., The Design of the PID Controller, Klipsch School of Electrical and Computer Engineering, June, 2001.

ÖZGEÇMİŞ

Rüştü AKAY 1979 yılının Mayıs ayında Kayseri’de doğdu. İlk, orta ve lise öğrenimini Kayseri’de tamamladı. 1997 yılında Erciyes Üniversitesi’nde bir yıl yabancı dil eğitimi aldıktan sonra aynı üniversitenin Mühendislik Fakültesi Kontrol ve Bilgisayar Mühendisliği bölümünden 2002 yılında mezun oldu. Rüştü AKAY 2001 yılında öğrenci iken çalışmaya başladığı Şahin Yazılım A.Ş. firmasında halen Bilgisayar Mühendisi olarak görev yapmaktadır.

İletişim Bilgileri :

Adres: Şahin Yazılım, KAYSERİ

Tel. : (90)(352) 231 49 32

e-posta: rustuakay@mailcity.com