

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**SEZGİSEL OPTİMİZASYON ALGORİTMALARI
KULLANILARAK DİNAMİK AĞIRLIK ÖLÇME
SİSTEMİNİN KİMLİKLENDİRİLMESİ**

**Hazırlayan
Ahmet Emin BAKTİR**

**Danışman
Dr. Öğr. Üyesi Mustafa DANACI**

Yüksek Lisans Tezi

**Temmuz 2018
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**SEZGİSEL OPTİMİZASYON ALGORİTMALARI
KULLANILARAK DİNAMİK AĞIRLIK ÖLÇME
SİSTEMİNİN KİMLİKLENDİRİLMESİ**

(Yüksek Lisans Tezi)

**Hazırlayan
Ahmet Emin BAKTİR**

**Danışman
Dr. Öğr. Üyesi Mustafa DANACI**

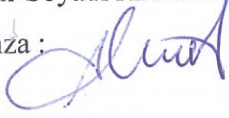
**Temmuz 2018
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Adı-Soyadı Ahmet Emin BAKTIR

İmza :



YÖNERGEYE UYGUNLUK

“Sezgisel Optimizasyon Algoritmaları Kullanılarak Dinamik Ağırlık Ölçme Sisteminin Kimliklendirilmesi” Yüksek Lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.



Tezi Hazırlayan

Ahmet Emin BAKTIR



Danışman

Dr. Öğr. Üyesi Mustafa DANACI



Prof. Dr. Veysel ASLANTAŞ

Bilgisayar Mühendisliği ABD Başkanı

KABUL VE ONAY

Dr. Öğr. Üyesi Mustafa DANACI danışmanlığında Ahmet Emin BAKTİR tarafından hazırlanan “*Sezgisel Optimizasyon Algoritmaları Kullanılarak Dinamik Ağırlık Ölçme Sisteminin Kimliklendirilmesi*” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

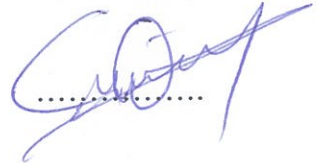
13 / 07 / 2018

JÜRİ:

Danışman : Dr. Öğr. Üyesi Mustafa DANACI



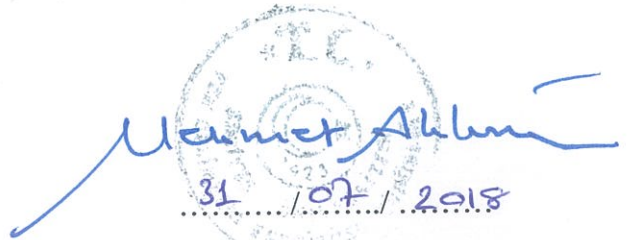
Üye : Doç. Dr. Celal ÖZTÜRK



Üye : Dr. Öğr. Üyesi Serdar KOÇKANAT

**ONAY:**

Bu tezin kabulü Enstitü Yönetim Kurulunun 31/07/2018.. tarih ve 2018./33-15... sayılı kararı ile onaylanmıştır.



31 / 07 / 2018

Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

TEŞEKKÜR

Çalışmalarım boyunca farklı bakış açıları ve bilimsel katkılarıyla beni aydınlatan, yakın ilgi ve yardımlarını esirgemeyen ve bu günlere gelmemde en büyük katkı sahibi sayın hocam Dr. Öğr. Üyesi Mustafa DANACI' ya teşekkürü bir borç bilirim.

Ayrıca; çalışmalarım süresince sabır göstererek beni daima destekleyen eşime ve aileme en içten teşekkürlerimi sunarım.

Ahmet Emin BAKTIR



SEZGİSEL OPTİMİZASYON ALGORİTMALARI KULLANILARAK DİNAMİK AĞIRLIK ÖLÇME SİSTEMİNİN KİMLİKLENDİRİLMESİ

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Temmuz 2018
Danışman: Dr. Öğr. Üyesi Mustafa DANACI

ÖZET

Zamanın etkili kullanımı, modern dünyada gittikçe önemli hale gelen bir problemdir. Endüstriyel alanlarda ürünlerin paketlenmesi ve taşınması gibi ağırlıklarının ölçülmesi işleminin de hızlı bir şekilde yapılması gerekmektedir. Dinamik ağırlık ölçme sistemi kalite kontrolü, aşırı yük tespiti, canlı ağırlıklarının belirlenmesi, ürünlerin doldurulması ve sıralanması, vb. alanlarda yaygın olarak kullanılmaktadır. Bu sistem, birbirinden farklı işlemler için özel çözüm tekniklerinin kullanılmasını gerektirmektedir. Ağırlık ölçme sisteminin amacı, kısa süre içerisinde uygulanan kütleyi doğru bir şekilde belirlemektir. Çalışmada sezgisel optimizasyon algoritmalarıyla sistemin kimliklendirilmesi ve daha az veri kullanılarak kütlenin ön tahmin işleminin iyi bir şekilde yapılması amaçlanmıştır.

Bu çalışmada kullanılan dinamik ağırlık ölçme sistemi ikinci dereceden diferansiyel denklem ile tanımlanmış olup M. Danacı tarafından yapılan doktora tez çalışmasında, sisteme ait denklemlerin homojen ve özel çözümleri yapılarak az sönümlü, aşırı sönümlü, kritik sönümlü modelleri geliştirilmiştir.

Nümerik bir çözüm elde etmek için öncelikle sisteme ait kütle, yay ve sönümleme katsayılarının doğru belirlenmesi amacıyla sistemi temsil eden parametrelerin doğru kimliklendirilmesi gerekmektedir. Sistemin uygun modelinin seçimi ve modele bağlı parametrelerin doğru belirlenmesi kimliklendirmenin en önemli parçası olup bu tez çalışmasında az sönümlü model kullanılmaktadır. Başlangıç şartları sıfır kabul edilerek dinamik ağırlık ölçme sisteminden gelen yer değiştirme bilgisi optimizasyon algoritmaları vasıtasıyla eğri uydurma işlemi için kullanılmaktadır.

Daha önceden yapılan dinamik ağırlık ölçme sisteminin kimliklendirilmesi ve kütlenin ön tahmin ile belirlenmesi işlemlerinde eğri uydurma metotlarından regresyon teknikleri, adaptif filtreleme ve yapay sinir ağı yöntemleri kullanılmıştır. Son zamanlarda sezgisel optimizasyon yöntemlerinden sürü zekası, sosyal ve evrimsel

tabanlı algoritmalar dinamik sistemlerin kimliklendirilmesinde kullanılan başarılı yöntemlerden birkaçıdır.

Bu çalışmada, dinamik ağırlık ölçme sisteminin kimliklendirilmesinde sezgisel optimizasyon algoritmalarından Emperyalist Yarışmacı, Ateş Böceği ve Parçacık Sürü algoritmaları kullanılmıştır. Gürültüsüz ve gürültülü sistem cevapları kullanılarak yapılan simülasyon çalışmalarında, önerilen optimizasyon algoritmaları ile ağırlık ölçme sistem parametrelerinin kimliklendirilmesi ve az sayıda veri kullanılarak kütlenin kısa sürede önceden tahmin edilmesi işlemlerinde oldukça başarılı sonuçlar elde edilmiştir. Çalışma offline olarak yapılmıştır. Algoritmaların online çalışmalarda kullanılabilmesi için cevap sürelerinin iyileştirilmesi gerekmektedir. Sonraki çalışmalarda algoritmaların hızlandırılması için paralel çözüm yöntemleri kullanılabilir.

Anahtar Kelimeler: Kütle Yay Sönümleme Sistemi, Dinamik Ağırlık Ölçme Sistemi, Sistem Kimliklendirme, Parametre Tahmini, Sezgisel Optimizasyon Algoritmaları, Parçacık Sürü Optimizasyon, Ateş Böceği, Emperyalist Yarışmacı Algoritma.

SYSTEM IDENTIFICATION OF DYNAMIC WEIGHING SYSTEM WITH HEURISTIC ALGORITHM

Ahmet Emin BAKTIR

**Erciyes University, Computer Engineering
M.Sc. Thesis, July 2018
Supervisor: Dr. Öğr. Üyesi Mustafa DANACI**

ABSTRACT

Effective use of time is an increasingly important problem in the modern world. Measuring the weight of products also needs to be done quickly like packing and transporting in industrial. Dynamic weighing system is widely used in quality control, overload detection, weighting of live beings, filling and sorting of products, etc. This system requires the use of special solution techniques for different processes. The aim of the weighing system is to accurately predict the applied mass within a short period of time. The aim of this study is to identify the system with the heuristic optimization algorithms and to make the prediction of the mass.

The dynamic weighing system used in this study is defined by the second order differential equation. In the PhD. study carried out by M. Danaci, homogeneous and special solutions of the system equations have been made and underdamped, overdamped, critical damped models have been developed.

In order to obtain a numerical solution, firstly the parameters representing the system must be correctly identified in order to determine the spring, damper coefficients and mass of the system correctly. The selection of the appropriate model of the system and the correct identification of the model dependent parameters are the most important parts of the identification. An underdamped model is used in this study. The initial conditions are assumed to be zero and displacement information from dynamic weighing system is used for curve fitting.

Regression techniques, adaptive filtering and artificial neural networks have been used as the methods of curve fitting in the identification of the dynamic weighing system and the prediction of the mass. Recently, swarm, social and evolutionary algorithms based

on heuristic optimization methods are some of the successful methods used in the identification of dynamic systems.

In this study, the Imperialist Competitive, Firefly and Particle Swarm Optimization algorithms were used as the heuristic optimization algorithms in the identification of the dynamic weighing system. In simulation studies using noiseless and noisy system responses, very successful results have been obtained with the proposed optimization algorithms, identification weighing system parameters and prediction of applied mass. The study was done offline. Response times must be improved so that algorithms can be used online. Parallelization can be used to accelerate the algorithms in future studies.

Keywords: Mass-Spring-Damper System, Dynamic Weighing System, System Identification, Parameter Estimation, Heuristic Algorithms, Particle Swarm Optimization, Firefly Algorithm, Imperialist Competitive Algorithm.

İÇİNDEKİLER

SEZGİSEL OPTİMİZASYON ALGORİTMALARI KULLANILARAK DİNAMİK AĞIRLIK ÖLÇME SİSTEMİNİN KİMLİKLENDİRİLMESİ

BİLİMSEL ETİĞE UYGUNLUK	i
YÖNERGEYE UYGUNLUK.....	ii
TEŞEKKÜR.....	iv
ÖZET.....	v
ABSTRACT.....	vii
İÇİNDEKİLER	ix
KISALTMALAR VE SİMGELER.....	xii
TABLolar LİSTESİ.....	xv
ŞEKİLLER LİSTESİ	xv

BÖLÜM 1

GİRİŞ

1.1. Literatür Taraması.....	2
------------------------------	---

BÖLÜM 2

GENEL BİLGİLER

2.1. Sistem Kimliklendirme Modelleri.....	6
2.2. Sistem Kimliklendirme	7
2.3. Kütle Yay Sönümlleme Sistemi	8
2.3.1. Az Sönümlü	12
2.3.2. Kritik Sönümlü	14
2.3.3. Aşırı Sönümlü.....	15

BÖLÜM 3

GEREÇ VE YÖNTEMLER

3.1. Optimizasyon	18
3.2. Parçacık Sürü Optimizasyon Algoritması	19
3.2.1. Giriş.....	19

3.2.2. Parçacık Sürü Optimizasyonu	19
3.2.2.1. PSO parametreleri	20
3.2.3. PSO Adımları ve Akış Diyagramı	22
3.2.4. PSO Örnek Çözüm	24
3.3. Ateş Böceği Algoritması	27
3.3.1. Giriş.....	27
3.3.2. Ateş Böceği Algoritması	27
3.3.3. Ateş Böceği Algoritması Adımları ve Şeması	30
3.3.4. Ateş Böceği Örnek Çözüm	31
3.4. Emperyalist Yarışmacı Algoritma	36
3.4.1. Giriş.....	36
3.4.2. Emperyalist Yarışmacı Algoritma	36
3.4.3. Emperyalist Yarışmacı Algoritma Adımları.....	39
3.4.4. Emperyalist Yarışmacı Algoritma Örnek Çözüm	40

BÖLÜM 4

SEZGİSEL OPTİMİZASYON ALGORİTMALARININ AĞIRLIK ÖLÇME SİSTEMİNDE KULLANILMASI

4.1. Verinin Hazırlanması.....	47
4.2. Algoritmaların Sistem Kimliklendirme Performanslarının Test Edilmesi....	49
4.2.1. Parçacık Sürü Optimizasyon Algoritması ile Sistem Kimliklendirme ...	49
4.2.1.1. Parçacık Sürü Optimizasyon Algoritma Değişkenleri.....	49
4.2.1.2. Gürültüsüz Sistem Cevabı Sonuçları	49
4.2.1.3. Gürültü Eklenmiş Sistem Cevabı Üzerinden Sonuçlar	50
4.2.2. Ateş Böceği Algoritması ile Sistem Kimliklendirme.....	55
4.2.2.1. Ateş Böceği Algoritmasının Parametreleri	55
4.2.2.2. Gürültüsüz Çıktı Sonuçları	55
4.2.2.3. Gürültü Eklenmiş Sistem Cevabı Üzerinden Sonuçlar	56
4.2.3. Emperyalist Yarışmacı Algoritması ile Sistem Kimliklendirme	61
4.2.3.1. Emperyalist Yarışmacı Algoritmasının Değişkenleri	61
4.2.3.2. Gürültüsüz Sistem Cevabı Sonuçları	61
4.2.3.3. Gürültü Eklenmiş Sistem Cevapları Üzerinden Sonuçlar	63
4.2.4. Farklı Veri Büyüklüklerinde Kimliklendirme	67

4.2.5. Yay Sabiti ve Sönümlleme Sabitinin Kimliklendirilmesi	70
4.2.5.1. Veri Setinin Hazırlanması	70
4.2.5.2. Yay ve Sönümlleme Sabitlerinin Kimliklendirilmesi.....	71
4.3. Ön Tahmin(Prediction) ile Kütlenin Belirlenmesi	73
4.3.1. Ön Tahmin ile Kütlenin Belirlenmesinde Bölge Seçimi	73
4.3.2. Ön Tahmin ile Kütlenin Belirlenmesi	76
4.4. Sonuçların Değerlendirilmesi	78

BÖLÜM 5

TARTIŞMA SONUÇ VE ÖNERİLER

KAYNAKÇA	84
ÖZGEÇMİŞ.....	92

KISALTMALAR VE SİMGELER

<u>Sembol</u>	<u>Anlamı</u>	<u>Birimi</u>
SSE	Sum Of Squared Error	
EYA	Emperyalist Yarışmacı Algoritma	
ABA	Ateş Böceği Algoritması	
PSO	Parçacık Sürü Optimizasyon	
NLR	Non-Linear Regresyon	
MLP	Multi-Layer Perceptron	
PID	Proportional Integral Derivative	
DHDP	Direct Heuristic Dynamic Programming	
IPIDNN	Improved PID-Neural Network	
FIR	Finite Impulse Response	
IIR	Infinite Impulse Response	
UKF	Unscented Kalman Filter	
BP	Backpropagation	
SSA	Singular Spectrum Analysis	
LVQ	Learning Vector Quantization	
SISO	Single Input Single Output	
MIMO	Multiple Input Multiple Output	
MISO	Multiple Input Single Output	
M	kütle	kg
K	yay sabiti	N/m
C	sönümleme sabiti	N/(m/s)
m_s	platform ağırlığı	kg
g	yerçekim ivmesi	m/s ²
δ	sönümleme faktörü	
w_n	doğal frekans	
w_d	sönüm frekansı	
p_i	popülasyon bireyinin en iyi çözümü	
p_{best}	en iyi popülasyon bireyi	
$v_{i,j}$	popülasyon birey hızı	
x_i	popülasyon birey konumu	

c_1, c_2	hızlanma ve öğrenme katsayıları
w	PSO arama katsayısı
w_{damp}	PSO arama katsayısı oranlama
i	birey indeksi
j	parametrenin boyutu
I	ışık yoğunluğu
r	ateş böcekleri arasındaki mesafe
β	çekicilik katsayısı
n_{pop}	popülasyon büyüklüğü
d	parametre boyutu
ub	parametre üst sınırı
lb	parametre alt sınırı
it_{max}	maksimum adım sayısı
u_i	bireyin uygunluk değeri
γ	ışık emilim katsayısı
α	mutasyon katsayısı
α_{damp}	mutasyon katsayı değiştirme
M_n	ilgili imparatorluğun normalleştirilmiş maliyeti
m_n	ilgili emperyalistin maliyeti
β	asimilasyon katsayısı
α	seçim baskısı
p_{rev}	devrim olasılığı
μ	devrim oranı
l	emperyalist ile koloni arasındaki mesafe
n_{col}, N_{col}	koloni sayısı
NM_n	ilgili emperyalistin ilgili adımda sahip olduğu koloni sayısı
n_{emp}	emperyalist sayısı
emp_i	i. emperyalist
p_{total}	toplam güç
P_n	normalleştirilmiş güç
col_i	i. emperyalistin kolonileri
$p_{col,i}$	i. emperyalistin kolonilerinin uygunluk değerleri

$p_{emp,i}$	i. emperyalistin uygunluk deęerleri
t	zaman
N	veri sayısı



TABLOLAR LİSTESİ

Tablo 4.1	PSO ile gürültüsüz referans kullanılarak bulunan parametreler.	50
Tablo 4.2	%1 gürültü referans cevabına göre PSO ile bulunan parametreler	51
Tablo 4.3	%5 gürültü referansa göre PSO ile belirlenen parametreler	52
Tablo 4.4	%10 gürültü referansa göre PSO ile bulunan parametreler	53
Tablo 4.5	%20 gürültü referansa göre belirlenen parametreler.....	54
Tablo 4.6	Gürültü eklenmemiş referansa göre ABA ile belirlenen parametreler	55
Tablo 4.7	%1 gürültü eklenmiş referansa göre ABA ile bulunan parametreler.....	57
Tablo 4.8	%5 gürültü eklenmiş referansa göre ABA ile bulunan parametreler.....	58
Tablo 4.9	%10 gürültü eklenmiş referansa göre ABA ile bulunan parametreler.....	59
Tablo 4.10	%20 gürültü eklenmiş referansa göre ABA ile bulunan parametreler.....	60
Tablo 4.11	Gürültü eklenmemiş referansa göre EYA ile bulunan parametreler.....	62
Tablo 4.12	%1 gürültü eklenmiş referansa göre EYA ile bulunan parametreler	63
Tablo 4.13	%5 gürültü eklenmiş referans çıktısına göre EYA ile bulunan parametreler	64
Tablo 4.14	%10 gürültü eklenmiş referansa göre EYA ile bulunan parametreler	65
Tablo 4.15	%20 gürültü eklenmiş referansa göre EYA ile bulunan parametreler	66
Tablo 4.16	N=100 adet veri için PSO sonuçları	67
Tablo 4.17	N=100 adet veri için ABA sonuçları	67
Tablo 4.18	N=100 adet veri için EYA sonuçları.....	67
Tablo 4.19	N=500 adet veri için PSO sonuçları	67
Tablo 4.20	N=500 adet veri için ABA sonuçları	68
Tablo 4.21	N=500 adet veri için EYA sonuçları.....	68
Tablo 4.22	N=1000 adet veri için PSO sonuçları	68
Tablo 4.23	N=1000 adet veri için ABA sonuçları	69
Tablo 4.24	N=1000 adet veri için EYA sonuçları.....	69
Tablo 4.25	N=2000 adet veri için PSO sonuçları	69
Tablo 4.26	N=2000 adet veri için ABA sonuçları	69
Tablo 4.27	430 N=2000 adet veri için EYA sonuçları.....	70
Tablo 4.28	PSO ile bulunan yay ve sönümleme sabitlerinin ortalamaları	71
Tablo 4.29	ABA ile bulunan yay ve sönümleme sabitlerinin ortalamaları.....	72
Tablo 4.30	EYA ile bulunan yay ve sönümleme sabitlerinin ortalamaları	73

Tablo 4.31	$t = 0$ anından $N=40$ veri için kütlenin bulunması	74
Tablo 4.32	$t \neq 0$ anından $N=40$ veri için kütlenin bulunması	75
Tablo 4.33	PSO ile bulunan kütle değerleri	77
Tablo 4.34	ABA ile bulunan kütle değerleri	77
Tablo 4.35	EYA ile bulunan kütle değerleri	78



ŞEKİLLER LİSTESİ

Şekil 2.1	Temel sistem şeması	8
Şekil 2.2	Ağırlık ölçme sisteminin modeli [16]	8
Şekil 2.3	Ağırlık ölçme sisteminden alınan cevap	9
Şekil 2.4	Ağırlık Ölçme Sistemi Birim Basamak Cevapları [16]	17
Şekil 3.1	Parçacığın hareketi	23
Şekil 3.2	PSO akış diyagramı	24
Şekil 3.3	Parçacıkların konumları	26
Şekil 3.4	Çözüm sonunda parçacığın güncellenen konumu	26
Şekil 3.5	Ateş Böceği Algoritması akış diyagramı	31
Şekil 3.6	Emperyalistlerin kolonilerle ilişkisi	38
Şekil 3.7	Kolonilerin Emperyalistlere doğru hareketi	38
Şekil 3.8	Emperyalist Yarışmacı Algoritma akış diyagramı	40
Şekil 4.1	Hazırlanan veriler	48
Şekil 4.2	Gürültü eklenmiş referans, bulunan parametrelerle alınan sistem cevabı	50
Şekil 4.3	%1 Gürültü eklenmiş referans, bulunan parametrelerle alınan sistem cevabı	51
Şekil 4.4	%5 gürültü eklenmiş referans, bulunan parametrelerle alınan sistem cevabı	52
Şekil 4.5	%10 gürültü eklenmiş referans, bulunan parametrelerle alınan sistem cevabı	53
Şekil 4.6	%20 gürültülü referans, bulunan parametrelerle alınan sistem cevabı	54
Şekil 4.7	Gürültü eklenmemiş referans, bulunan parametreler ile alınan sistem cevabı	56
Şekil 4.8	%1 gürültü eklenmiş referans, bulunan parametreler ile alınan sistem cevabı	57
Şekil 4.9	%5 gürültülü referans, bulunan parametreler ile alınan sistem cevabı	58
Şekil 4.10	%10 gürültülü referans, bulunan parametreler ile alınan sistem cevabı	59
Şekil 4.11	%20 gürültülü referans, bulunan parametreler ile alınan sistem cevabı	60
Şekil 4.12	Gürültüsüz referansa, bulunan parametreler ile alınan sistem cevabı	62
Şekil 4.13	%1 gürültülü referans, bulunan parametreler ile alınan sistem cevabı	63

Şekil 4.14	%5 gürültü eklenmiş referans, bulunan parametreler ile alınan sistem cevabı	64
Şekil 4.15	%10 gürültülü referans, bulunan parametreler ile alınan sistem cevabı	65
Şekil 4.16	%20 gürültülü referans, bulunan parametreler ile alınan sistem cevabı	66
Şekil 4.17	$t = 0$ anından $N = 40$ veri için sırasıyla %0, %1, %5, %10 ve %20 referanslar	74
Şekil 4.18	$t \neq 0$ anından $N = 40$ veri için sırasıyla %0, %1, %5, %10 ve %20 referanslar	75



BÖLÜM 1

GİRİŞ

Günümüzde büyük ölçekli üretim yapan endüstriler, sahip oldukları sistemlerle güvenli, verimli ve kaliteli üretim yapabilmek için sistemlerle birlikte kontrol mekanizmalarını da bulundurması gerekmektedir. Kontrol edilmesi gereken dinamik sistemin matematiksel modelinin bilinmesi zorunludur. Dinamik sistemlerin, önceki bilgilerden ve gözlemlerden faydalanarak matematiksel modelinin oluşturulması sistem modelleme olarak tanımlanır [1,2]. Dinamik bir sistemin matematiksel modelinde yer alan parametreler deneysel verilerden faydalanılarak belirlenmesi, sistem kimliklendirme sürecinin bir parçasıdır [3]. Sistemin etkin bir şekilde kontrolünü sağlayabilmek için ve çıkış değerlerinin her zaman güvenilebilir olabilmesi açısından kimliklendirilecek sistemin parametre değerleri, mümkün olan en az hata ile bulunmalıdır. Sistemin giriş ve çıkış değerleri arasında lineer olmayan bir ilişkisi varsa ve beklenmeyen girişler sistemi etkiliyorsa, parametre değerlerinin minimum hata ile belirlenmesi zorlaşmaktadır [3-9]. Dolayısıyla sistem kimliklendirme işlemi de zaman ve performans açısından verimsizleşmeye başlar.

Dinamik ağırlık ölçme sistemi birçok alanda kullanılan bir kütle belirleme sistemidir. Bu sisteme karayolları, köprüler, transfer şirketleri, yüksek hacimli üretim tesisleri vb. endüstriler tarafından ihtiyaç duyulmaktadır. Dinamik ağırlık ölçme sistemini kimliklendirme işlemi ne kadar iyi yapılırsa, geçici rejim cevabı üzerinden yapılan kütle belirlenmesi işlemi o kadar başarılı olmaktadır.

Tez çalışması, sezgisel optimizasyon algoritmalarının, dinamik ağırlık ölçme sistemin kimliklendirilmesi işleminde kullanılabilirliğini araştırmaktadır.

Sezgisel algoritmalar, belirlenen bir amacı gerçekleştirmek veya hedefe varmak için doğal oluşumlardan esinlenen algoritmalar. Bu algoritmalar, çözüm uzayında yer

alan mutlak bir çözümü işaret etmemektedir. Yani sezgisel algoritmalar çözüme yaklaşan fakat kesin çözümü garanti etmeyen algoritmalarlardır. Anlaşılabilirlik ve probleme uygulanabilirlik açılarından basit yapıda olmalarından ve optimizasyon problemlerinde gösterdikleri performanslardan dolayı optimum çözümü bulma işlemlerinde sezgisel algoritmaların kullanımına ihtiyaç duyulmaktadır[10].

1.1 Literatür Taraması

Ağırlık ölçme platformunun parametre tahminleri üzerine birçok yöntem uygulanmıştır. Çoğunlukla kullanılan yöntemler, adaptif filtre teknikleri, NLR metodu, yapay sinir ağı modelleri vb. olarak sıralanabilir.

W. J. Shi, dinamik ağırlık ölçme sistemi üzerinde adaptif filtreleme tekniğini kullanmıştır [11]. W. Q. Shu, dinamik ağırlık sisteminin sıfır olmayan başlangıç koşullarında çalışmasını yapmıştır [12]. M. Danacı ve çalışma arkadaşı, dinamik ağırlık sisteminin modeli üzerinde Non-Linear Regresyon(NLR) metodu kullanarak uygulanan ağırlığın tahmini yapmışlardır[13]. D. H. Horrock ve M. Danacı, platform dinamik ağırlık ölçme sisteminin üç farklı modelinden birinde cevap veren otomatik bir yöntem geliştirmişlerdir [14]. M. Danacı, dinamik ağırlık ölçme sistemini NLR metoduyla parametre kimliklendirmesini içeren tez çalışmasını yapmıştır [15]. D. H. Horrock ve M. Danacı, platform geçişinde ağırlık bulunmasında bir otomatik regresyon modelini önermiştir[16].

M. Danacı ve çalışma arkadaşı, bilinmeyen yükleme zamanı, başlangıç şartlarının ve ağırlık ölçme sistemin model parametrelerini erken safhada belirlemiş ve modelleme hatası yöntemiyle uygulanan ağırlık tahmin edilmiştir. Ardından ise NLR metodu ile otomatik tahmin işlemi yapılmıştır [17]. M. Danacı ve çalışma arkadaşı, gürültülü veriler bilinmeyen yükleme zamanını, başlangıç şartlarının ve ağırlık ölçme sistemin model parametrelerini erken safhada belirlemiş ve modelleme hatası yöntemiyle uygulanan ağırlık tahmin edilmiştir. Ardından ise NLR metodu ile otomatik tahmin işlemi yapılmış ve Adaptif Filtre tekniği ile karşılaştırılmıştır[18]. M. Danacı ve çalışma arkadaşları, yapay sinir ağlarının çok katmanlı mimarisini kullanarak doğru kütle tahminin yapılmasını sağlamıştır [19]. M. Danacı, NLR metoduna dayanan Gauss-Newton modeli kullanarak ağırlık ölçme sisteminden çıkan, zaman domenindeki sistem

cevabı ile eğri uydurma yöntemi kullanarak, uygulanan ağırlığının tahminini yapmıştır. [20]. M. Danacı ve çalışma arkadaşı, MLP(Multi-Layer Perceptron) yapay sinir ağı modeli kullanılarak uygulanan kütle yükleme işlemi devam ederken tahmini yapılmıştır [21]. G. Liao ve çalışma arkadaşı, kara yollarındaki köprü, asfalt vb. yapılara zarar veren araçları üzerinde analiz yaparak lineer regresyon modeline dayalı, medyan ve ortalama dinamik bir ağırlık ölçme sistemini tasarlamışlardır. Hareketli araçların kütle tahmininde iyileştirmeler yapmışlardır[22]. W. Jeridi ve çalışma arkadaşları, bu çalışmada ki öncelikli amacı filtreleme teknikleriyle bir uzman ağırlık ölçme süreci geliştirmektir. Ek olarak, SAT(Security Analysis Team)' nin SRA(Security Risk Analysis) birlikte çalışmasının iyileştirilmesini yapmaktır [23]. P. Hu ve çalışma arkadaşları, dinamik sayısal ağırlık ölçme sisteminde doğruluk ve hız arasındaki çelişkiyi çözmek için PID(Proportional Integral Derivative) kontrol teorisi ve sinir ağı teorisi kombine edilerek yeni bir akıllı kontrol stratejisi önermişlerdir. [24]. J. Sun ve çalışma arkadaşları, geliştirilmiş DHDP(Direct Heuristic Dynamic Programming) ağırlarına PID-sinir ağı olarak bilinen IPIDNN' in yapılandırılması uygulanmıştır. Bu çalışma altında, bir PID kontrolörü ile birlikte yeni bir başlatma yaklaşımı önerilmiştir. Böylece DHDP öğrenme süreci daha iyi bir başlangıç şartlarından başlamaktadır. Başlatma yaklaşımının etkinliğini el arabası modeli üzerinde denenmiştir [25]. Z. Ying ve çalışma arkadaşları, FIR(Finite Impulse Response) ve IIR(Infinite Impulse Response) filtre tasarımını içeren veri işleme yöntemi üzerine bir çalışma yapmışlardır. Kalman filtresinin ağırlık ölçme sistemi üzerine uygulanan bu çalışma daha iyi sonuçlar verdiğini gözlemlenmiştir [26]. A. D. Martin ve çalışma arkadaşları, süt tozunun torbalara doldurulması işleminde, dikey kuvvet bileşeninin simüle edilmiş ölçümlerinden askıya alınmış bir süt tozu torbasının sıralı kütle çıkarımını yapmışlardır. Kalman parçacıkları UKF(Unscented Kalman Filter) zamanın bir fonksiyonu olarak süt tozu kütlelerinin tahmin işlemi incelenmektedir. Kalman parçacıklarının daha iyi sonuçlar verdiği diğer yöntemler ile karşılaştırılması sonucu gösterilmiştir [27]. M. Niedźwiecki ve çalışma arkadaşı, ağırlık ölçme sistemi cevabının sonlu sistem cevabının modeline dayanan filtreleme şeması uygulamışlardır. Sonuçlar en son çalışmalardan dört kat daha yüksek olduğunu göstermişti [28]. M. Hamilic ve çalışma arkadaşı, dinamik ağırlık ölçme sistemine sahip ürün kütlesi hakkında daha iyi bir çözüm sunmak için Kalman Filtresi kullanmıştır [29]. M. Hamilic ve çalışma arkadaşları, dinamik ağırlık ölçme sistemi için C-clustering yöntemine dayanan bulanık tahmin mekanizması önermişlerdir.

Bu işlemlerin yanında FIR filtresi ve Hann Hammer modeli ile karşılaştırmalar yapılmıştır. Sonucunda bulanık mantık tahmincisi dijital filtreden daha iyi sonuçlar elde ettiğini göstermiştir [30]. J. Li ve çalışma arkadaşları, dinamik ağırlık ölçme sistemi ile aşırı yüklü kütlelerin tahmini için Wavelet, Genetik algoritma ve ARX ile bir çözüm hazırlamışlardır [31]. C. Xiaoyan ve çalışma arkadaşları, sistemin doğruluğunu ve hızını geliştirmeyi amaçlayan dinamik ağırlık ölçme işlemi için akıllı kontrolör üzerine çalışmışlardır. Kendi aralarında organize olma ve kendi kendine öğrenme yetenekleri olan sinir ağı ile bulanık mantığına dayanan yapay zeka tekniği kullanmışlardır. Hız ve doğruluk oranı arasındaki çelişkiyi ortadan kaldırmak için organik birleştirme tekniğini kullanmışlardır [32]. H. Gao ve çalışma arkadaşları, dinamik ağırlık ölçme sistemi üzerinde iyileştirme yapmak için sistem üzerinde ki hataları ve sebeplerini analiz etmişlerdir. Dinamik ağırlık ölçme sistemi için bir BP sinir ağı modeli içeren bir çalışma yapmışlardır [33]. Q. Wu ve çalışma arkadaşları, gürültü azaltıcı bir algoritma olan SSA(Singular Spectrum Analysis) ile kendi kendine organize olabilen yapay sinir ağı LVQ(Learning Vector Quantization) arasında hiyerarşik bir yapı kurarak dinamik ağırlık ölçme sistemi için sinyal işleme platformu hazırlamışlardır. Deneysel sonuçları kayda değer iyileşme olduğunu göstermiştir [34]. J. G. Gray ve çalışma arkadaşları, Genetik Programlama ile laboratuvar ölçeğinde eşleştirilmiş bir su deposu sistemi modellenmesi ve bir helikopter motorunun tanımlanmasına başarılı bir şekilde uygulamışlardır [35]. D. Sendrescu ve çalışma arkadaşları, Parçacık Sürü Optimizasyonu (PSO) algoritması kullanarak bir bakteri büyütme süreci (biyoproses) için offline parametrelerin tanımlanması işlemi yapılmıştır [36]. D. Sendrescu ve çalışma arkadaşı, parçacık sürüsü optimizasyonu ve genetik algoritmalar kullanarak kalibre edilen 9 bilinmeyen parametreye sahip bakteri büyütme sürecinin matematiksel modeli tanımlamıştır. Mikrobiyal büyümeyi tanımlamak için yaygın olarak kullanılan iki kinetik ifade, Monod ve Haldane denklemleri, model simülasyonlarında test edilmiştir [37]. M. Ulinowicz ve çalışma arkadaşları, genetik algoritmalarla bir gemi modelinin kimliklendirilmesi için önerilmiştir [38]. M. Kumar ve çalışma arkadaşları, Yarasa Optimizasyon algoritmasını adaptif bir IIR sisteminin tasarımında kullanılmıştır [39]. S. Ryzhikov ve çalışma arkadaşları, ters matematiksel modelleme problemini çözmek için evrimsel optimizasyon tekniğini ve performansı artırmak için algoritma yeniden başlatma tekniğini uygulamıştır [40].

Tez çalışmasının ikinci bölümünde sistem kimliklendirmenin tanımı ve neden yapıldığı sistem kimliklendirme model yapıları ve sistem kimliklendirme süreci hakkında bilgi verilmektedir. Üçüncü bölümde son yıllarda birçok alanda karşılaşılan problemlerin çözümünde sıkça başvurulmuş sezgisel yöntemlerde; Ateş Böceği (ABA), Parçacık Sürü Optimizasyonu (PSO) ve Emperyalist Yarışmacı(EYA) algoritmalarının anlatımlarına yer verilmektedir. Kullanılan algoritmalarla ilgili algoritma adımları, algoritmada kullanılan değişkenler ve değişkenlerin algoritmanın seyrini nasıl etkilediği konularından bahsedilmektedir. Dördüncü bölümde optimizasyon algoritmalarıyla yapılan simülasyon çalışması gösterilmektedir. Farklı veri seti büyüklüklerinden alınan parametrelerin karşılaştırılması ve sistem cevabının erken safhasında ön tahmin ile kütlenin belirlenmesi işlemi yapılmaktadır. ABA, PSO ve EYA yöntemleri ile yapılan çalışmanın sonuçları grafikler ve tablolar halinde sunulmaktadır.

Üçüncü bölüm iki kısımdan oluşmaktadır. İlk kısımda algoritmaların kimliklendirme performansları zaman aralığı değişmeden, 4 sn. boyunca 100, 500 1000 ve 2000 veri seti büyüklüğüne sahip sistem cevapları ile kimliklendirme işlemi yapılmaktadır Bu kısımda yapılan çalışmalar test amaçlı olup algoritmaların kimliklendirme ve ön tahmin işlemlerindeki kullanılabilirliği gösterilmektedir. Son olarak belirlenmiş sınırlar içerisinde yer alan kütle değerlerine karşılık ‘C’ ve ‘K’ parametreleri algoritmalar vasıtasıyla kimliklendirilmektedir

İkinci kısımda kimliklendirme işlemi tamamlandıktan sonra sistem farklı kütle değerleriyle ön tahmin performansları test edilmektedir.

Beşinci bölümde kullanılan algoritmaların karşılaştırmalarından, sonra ki çalışmalarda yapılabileceklerden, geliştirilebilecek alanlardan, avantaj ve dezavantajlardan bahsedilmektedir.

BÖLÜM 2

GENEL BİLGİLER

2.1. Sistem Kimliklendirme Modelleri

Sistem kimliklendirme işleminde kullanılan modeller giriş-çıkış parametre sayısına, zaman bağımlılığına, domene, doğrusallık durumuna ve bozucu etmenlere göre sınıflandırılmaktadır.

a) Giriş-çıkış değişkenlerine göre

Giriş ve çıkış parametre sayısına göre sınıflandırılan modellerdir. Bir giriş ve karşılığında bir çıkışa sahip olan modeller SISO(Single Input Single Output), birden fazla giriş ve birden fazla çıkışa sahip olan modeller MIMO(Multiple Input Multiple Output), birden fazla giriş ve bir çıkışa sahip olan modeller MISO(Multiple Input Single Output) olarak adlandırılmaktadır. Dinamik sistemlerin yaygın olarak kullanılan modeli birden fazla giriş karşılık bir çıkışa sahip olan MISO modelidir. SISO modeline göre parametrelerin belirlenmesi daha zordur [41,42].

b) Zamana göre:

Modellerin zaman bağımlılığına göre sınıflandırılmasıdır. Bazı modeller iç dinamikleri zamana bağımlıyken, bazı sistemler ise zamandan etkilenmezler dinamik sistemlerin çoğunun zaman bağımlılığı bulunmaktadır. Sistem tarafından verilen cevaplar zamana göre değişkenlik göstermektedir. Hesaplamanın kolay olması için zamana bağımsız modellerin kullanımı yaygındır [42].

c) Domene göre:

Sistem modelleri zaman ve frekans olmak üzere iki domende incelenmektedir. Zaman domeni diferansiyel ve fark denklemleri ile kimliklendirme işleminde kullanılırken,

frekans domeni modelleri spektral yoğunluk veya bode eğrisi gibi sistemlerin kimliklendirilmesinde kullanılmaktadır [42-47].

d) Doğrusallığın durumuna göre:

Giriş, çıkış parametreleri ile bozucular arasındaki matematiksel ilişkiye göre sistemlerin modellenmesidir. Sinyaller arasındaki ilişki doğrusal denklemler ile ifade edilebiliyorsa bu modeller, doğrusal modeller olarak adlandırılmaktadır. Sinyallerin arasındaki ilişki diferansiyel, üstel, logaritmik, trigonometrik vb. doğrusal olmayan denklemler ile ifade ediliyorsa, bu sistemler doğrusal olmayan modeller olarak adlandırılmaktadır [42].

e) Bozucu etkilere göre:

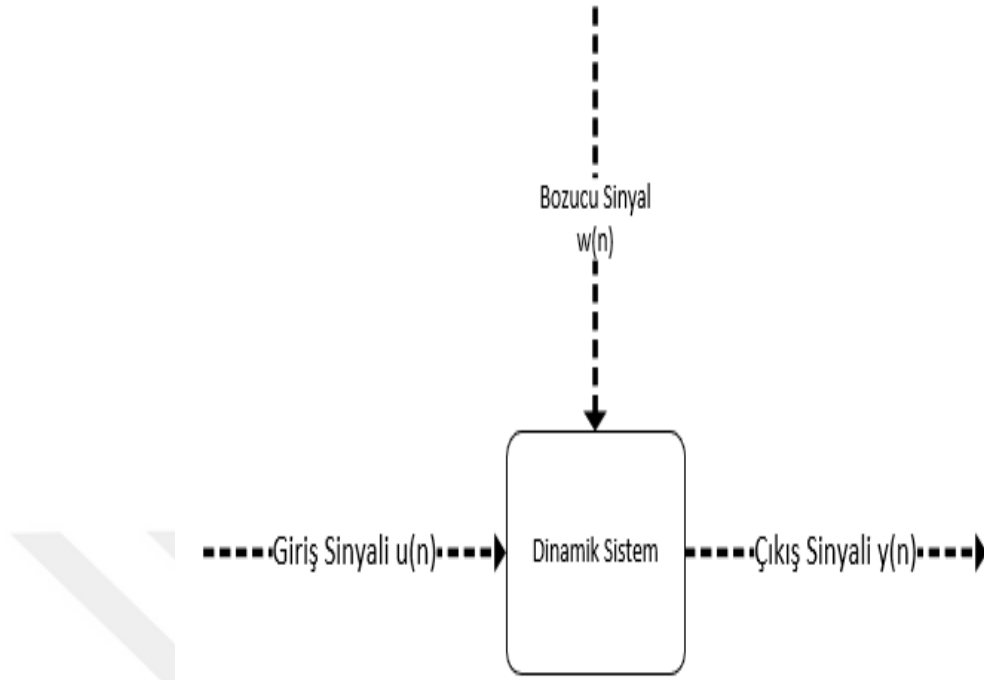
Giriş sinyalinin bilindiği ve çıkış sinyalinin tam olarak hesaplanabildiği modeller determinist model olarak adlandırılırken içyapısında dış etkilere kaynaklı hesaplanamayan rastgele değerlerin olduğu modeller stokastik model olarak adlandırılmaktadır. Birçok sistem stokastik model kullanılarak kimliklendirilmektedir [42].

2.2. Sistem Kimliklendirme

Her sistem kendi iç değişkenlere göre dış dünyadan aldığı etkileri şekillendirerek bir çıktı üretir. Sistem tarafından üretilen çıktı, sistemi etkileyen dış etmenler ve iç değişkenler arasında bir ilişkisi mevcuttur. Bir sistemin çıkış değişkenleri daha önceden aldığı girişlere göre şekilleniyorsa bu sistem dinamik bir sistemdir [41].

Sistem kimliklendirme, gerçek model üzerinden elde edilmiş giriş ve çıkış değerleri arasındaki ilişkiyi matematiksel bir tabana dayalı olarak ifade edilmektedir[1].

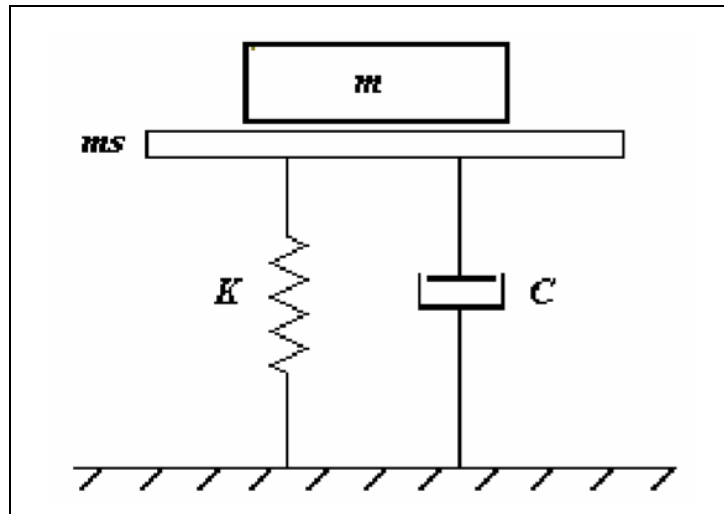
Sistem kimliklendirmenin amacı tekrardan kullanılabilecek güvenilir bir matematiksel model oluşturmaktır. Bir sistemin temel bileşenleri giriş sinyali $u(n)$, çıkış sinyali $y(n)$ ve bozucu sinyalden $w(n)$ oluşur. Dinamik sistem, giriş sinyali ve bozucu sinyallerden etkilenecek çıkış sinyalini oluşturur. Yapılan çalışmada $y(n)$ sinyali referans olarak kullanılarak sistem parametreleri tahmin edilmektedir. Dinamik sistemin temel şeması Şekil 2.1 ile gösterilmektedir.



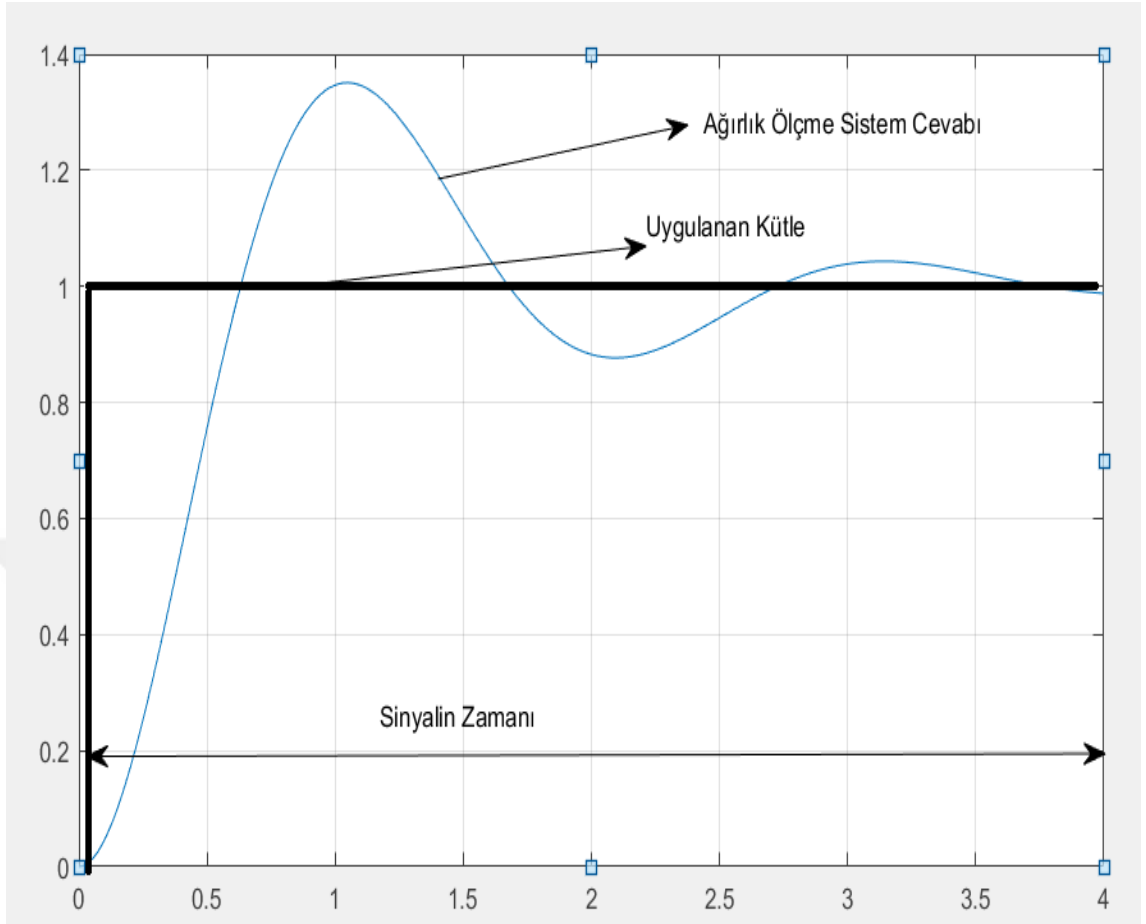
Şekil 2.1 Temel sistem şeması

2.3. Kütle Yay Sönümlleme Sistemi

Bir nesnenin ağırlık ölçme platformuna uygulanması Şekil 2.2 ile gösterilmektedir, kütlenin doğru bir şekilde belirlenebilmesi için belli bir süreyle Şekil 2.3 ile gösterilen geçici rejim cevabına ihtiyaç vardır. Modern dünyada doğru ve hızlı ağırlık ölçme işlemi önemli bir gereksinimdir [48-53].



Şekil 2.2 Ağırlık ölçme sisteminin modeli [16]



Şekil 2.3 Ağırlık ölçme sisteminden alınan cevap

Kütle yay sönümlleme sistemi, mekanik sistemlerde yaygın olarak kullanılan bir darbe sönümlleme sistemidir. Kütle yay sönümlleme sistemi kullanılarak ağırlık ölçme sistemi modellenmiştir. Kütle yay sönümlleme sistemi, ikinci dereceden diferansiyel denklemle (2.1) ifade edilir ve en genel formunda modellenmesi aşırı sönümlü, kritik sönümlü ve az sönümlü olarak yapılmaktadır [54,55].

$$(M + m_s) \frac{d^2 y(t)}{dt^2} + C \frac{dy(t)}{dt} + Ky(t) = F(t) = g * M/K \quad (2.1)$$

Denklemden gösterilen parametrelerden 'M' uygulanan kütleyi, ' m_s ' platformun ağırlığını, 'C' parametresi sönümlleme sabitini, 'K' parametresi yay sabitini ve 'g' parametresi yerçekimi ivmesini ifade etmektedir [7].

$F(t) = 0$ olduğu durumda, homojen çözüm için ikinci dereceden tanımlı denklem (2.2) ile gösterilmektedir.

$$M \frac{d^2 y(t)}{dt^2} + C \frac{dy(t)}{dt} + Ky(t) = 0 \quad (2.2)$$

Geleneksel çözümün yapılabilmesi için $y(t)$ değeri yerine (2.3) kabul edilmektedir.

$$y = e^{\lambda t} \quad (2.3)$$

Çözümler yapılarak yerlerine konulduğu zaman denklem (2.4)' teki şeklini almaktadır.

$$\left(\lambda^2 + \lambda \frac{C}{M} + \frac{K}{M} \right) e^{\lambda t} = 0 \quad (2.4)$$

$e^{\lambda t}$, üstel işlemlerin sıfır olması mümkün olmadığı için sadeleştirilme işlemi yapılarak denklem (2.5) elde edilmektedir.

$$\lambda^2 + \lambda \frac{C}{M} + \frac{K}{M} = 0 \quad (2.5)$$

Polinom köklerinin bulunması işlemi denklem (2.6) ile yapılmaktadır.

$$\lambda_{1,2} = -\frac{C}{2M} \pm \frac{1}{2} \sqrt{\left(\frac{C}{2M}\right)^2 - \frac{K}{M}} \quad (2.6)$$

Kökleri bulunan denklemin homojen çözümü denklem (2.7) ile verilmektedir.

$$y_c = A_1 e^{\lambda_1 t} + A_2 e^{\lambda_2 t} \quad (2.7)$$

A_1 ve A_2 değerleri birer sabit olup, başlangıç koşullarından türetilmektedir. Denklem dönüşüm işlemlerinin tamamlanmasıyla en son olarak denklem (2.8) deki halini almaktadır.

$$y_c = e^{-\frac{C}{2M}t} * \left[A_1 * \cos \left(\sqrt{\left(\frac{C}{2M}\right)^2 - \frac{K}{M}} \right) + A_2 * \sin \left(\sqrt{\left(\frac{C}{2M}\right)^2 - \frac{K}{M}} \right) \right] \quad (2.8)$$

Denklem kökleri değişkenlere bağlı olarak farklı değerler alabilmektedir. Bu şartlar altında üç farklı durumun öne çıkması söz konusudur.

Ağırlık ölçme sistemi parametreleri, sabitler 'K', 'C' ve değişkeni ise 'M' dir. Uygulanan kütle 'M' ve platform kütlesi, ' m_s ' sadeleştirilebilmesi için sıfır olarak kabul edilmektedir. Bunun yanında ilk konum ve hız değerleri sırasıyla $y_0 = 0$ ve $\dot{y}_0 = 0$ olarak belirlenmektedir [13,16,56]. Ağırlık ölçme sistemi uygulanan kütleye bağlı

olarak alınan üç tip tepki vardır. Bunlar az sönümlü, kritik sönümlü ve aşırı sönümlü cevaplardır [12,13,16,18,50,57-60].

$(\frac{c}{2M})^2$ değeri, $\frac{K}{M}$ değerinden büyük olduğu zaman iki tane gerçek kök vardır. Bu şartlar altında aşırı sönümlü(overdamped) olarak adlandırılan model kullanılır.

$(\frac{c}{2M})^2$ değeri, $\frac{K}{M}$ değerinden küçük olduğu zaman iki sanal kök vardır. Bu şartlar altında az sönümlü(underdamped) olarak adlandırılan model kullanılır.

$(\frac{c}{2M})^2$ değeri, $\frac{K}{M}$ değerine eşit olduğu zaman iki tane çakışık gerçek kök vardır. Bu şartlar altında kritik sönümlü(criticaldamped) olarak adlandırılan model kullanılır.

Sistemin denkleminde bulunan değişkenler denklem (2.9-2.11) arasında gösterilmektedir.

$$\delta = \frac{c}{2\sqrt{KM}}, \text{ sönümleme faktörü} \quad (2.9)$$

$$w_n = \sqrt{\frac{K}{M}}, \text{ doğal frekans} \quad (2.10)$$

$$w_d = w_n \sqrt{1 - \delta^2}, \text{ sönüm frekansı} \quad (2.11)$$

$F(t) \neq 0$ durumda homojen çözümün yanında özel çözümde yapılması gerekir. $F(t) \neq 0$ durumunda birim basamak girişi için özel çözüm denklem (2.12) ile verilmektedir [15].

$$y_p(t) = (M + m_s) \frac{g}{k} \quad (2.12)$$

Homojen ve özel çözümün birlikte verildiği genel çözüm, denklem (2.13) ile verilmektedir.

$$y(t) = y_c(t) + y_p(t) \quad (2.13)$$

Bu şartlar altında az sönümlü, kritik sönümlü ve aşırı sönümlü olmak üzere 3 durum öne çıkmaktadır.

2.3.1. Az Sönümlü

Köklerin sanal olması durumunda az sönümlü(underdamped) model cevabı kullanılmaktadır. Kökler sanal ve gerçek olmak üzere iki kısımdan oluşur. Bu köklerin gerçek kısmı denklem (2.14) ve sanal kısım denklem (2.15) ile gösterilmektedir.

$$-\delta * w_n = -\frac{c}{2M} \quad (2.14)$$

$$jw_n\sqrt{1-\delta^2} = j\sqrt{\frac{K}{M} - \left(\frac{c}{2M}\right)^2} \quad (2.15)$$

Denklemin köklerinin genel gösterimi denklem (2.16) ile gösterilmektedir.

$$\lambda_{1,2} = -\frac{c}{2M} \pm j\sqrt{\frac{K}{M} - \left(\frac{c}{2M}\right)^2} \quad (2.16)$$

Sönüm frekansını(w_d) denklem (2.17) ile hesaplanarak homojen çözüm üzerinde kullanılmaktadır.

$$w_d = w_n * \sqrt{1-\delta^2}$$

$$w_d = \sqrt{\frac{K}{M} - \left(\frac{c}{2M}\right)^2} \quad (2.17)$$

Az sönümlü sisteminin homojen çözümü denklem (2.18) ile gösterilmektedir.

$$y_c(t) = Ae^{-\frac{c}{2M}t} \sin(w_d t + \varphi) \quad (2.18)$$

Denklem içerisinde kullanılan ‘A’ değişkeni ‘ A_1 ve A_2 ’ sabitleri kullanılarak oluşturulan birer katsayıdır. bu işlem denklem (2.19) ile gösterilmektedir.

$$A = \sqrt{A_1^2 + A_2^2} \quad (2.19)$$

‘ φ ’ ile ifade edilen değişken faz açısıdır. Faz açısının hesabı denklem (2.20) ile gösterilmektedir.

$$\varphi = \tan^{-1}\left(\frac{A_1}{A_2}\right) \quad (2.20)$$

Genel çözümü elde etmek için hem özel hem homojen çözümün kullanılması gerekmektedir. Bu işlem denklem (2.21) ile gösterilmektedir.

$$y(t) = M \frac{g}{K} + y_c(t) \quad (2.21)$$

Bu denklemin çözümünde başlangıç anında sahip olunan hız ve konum değerleri sırasıyla $y(0) = y_0$, $\dot{y}(0) = \dot{y}_0$ alındığında, bağlı olan A_1 ve A_2 değerlerin hesabını içeren denklem (2.22-2.27) arasında verilmektedir.

$$y(t) = M \frac{g}{K} - e^{-\frac{c}{2M}t} [A_1 \cos(w_d t) + A_2 \sin(w_d t)] \quad (2.22)$$

$$y(0) = y_0 = M \frac{g}{K} + A_1 \quad (2.23)$$

$$A_1 = -M \frac{g}{K} - y_0 \quad (2.24)$$

$$\begin{aligned} \dot{y}(t) = & -\frac{C}{2M} e^{-\frac{c}{2M}t} [A_1 \cos(w_d t) + A_2 \sin(w_d t)] + \\ & e^{-\frac{c}{2M}t} [A_2 w_d \cos(w_d t) - A_1 w_d \sin(w_d t)] \end{aligned} \quad (2.25)$$

$$\dot{y}(0) = \dot{y}_0 = -\frac{C}{2M} A_1 + w_d A_2 \quad (2.26)$$

$$A_2 = \frac{\dot{y}_0 - \frac{C}{2M} A_1}{w_d} \quad (2.27)$$

Değerlerin yerlerine konulması işleminden sonra elde edilen genel çözüm denklem (2.28) ile gösterilmektedir.

$$y = M \frac{g}{K} - e^{-\frac{c}{2M}t} [A_1 \cos(w_d t) + A_2 \sin(w_d t)] \quad (2.28)$$

Bu denklem ile gösterilen çözüm yerine (2.29) ile gösterilen denklemi de kullanılabilir.

$$y = M \frac{g}{K} - A e^{-\frac{c}{2M}t} \sin(w_d t + \varphi) \quad (2.29)$$

Denklem içinde kullanılan değişkenlerin sadeleştirilmesi amacıyla, değişkenlere atanan denklem takımları denklem (2.30-2.34) arsında gösterilebilmektedir.

$$b_0 = M \frac{g}{K} \quad (2.30)$$

$$b_1 = A \quad (2.31)$$

$$b_2 = \frac{c}{2M} \quad (2.32)$$

$$b_3 = w_d \quad (2.33)$$

$$b_4 = \varphi \quad (2.34)$$

Sadeleştirme işleminin sonucunda elde edilen denklem (2.35) ile gösterilmektedir.

$$y_c = b_0 - b_1 e^{-b_2 t} \sin(b_3 t + b_4) \quad (2.35)$$

2.3.2. Kritik Sönümlü

Köklerin aynı olması durumudur. İki denklem çakışık olarak adlandırılır. Bu durumda sönüm katsayısı sıfır olarak işlem görmektedir. Bu şartlar altında denklem köklerinin hesaplanması işlemi denklem (2.36) ile gösterilmektedir.

$$\lambda_{1,2} = -\frac{c}{2M} = w_n \quad (2.36)$$

Sönüm katsayısına bağlı olan sönüm frekansı sıfır değerini almaktadır.

$$w_d = 0$$

Kritik sönümlü olarak adlandırılan genel çözüm denklem (2.37) ile gösterilmektedir.

$$y = M \frac{g}{K} - e^{-\frac{c}{2M} t} (A_1 + A_2 t) \quad (2.37)$$

Bu denklemin çözümünde başlangıç anında ki hız ve konum olan $y(0) = y_0$, $\dot{y}(0) = \dot{y}_0$ alındığında bağlı olan A_1 ve A_2 değerlerinin hesaplanması için kullanılan denklemler (2.38-2.42) ile gösterilmektedir.

$$y(0) = y_0 = M \frac{g}{K} + A_1 \quad (2.38)$$

$$A_1 = M \frac{g}{K} - y_0 \quad (2.39)$$

$$\dot{y} = -\frac{c}{2M} e^{-\frac{c}{2M}t} (A_1 + A_2 t) + A_2 e^{-\frac{c}{2M}t} \quad (2.40)$$

$$\dot{y}(0) = \dot{y}_0 = -\frac{c}{2M} * A_1 + A_2 \quad (2.41)$$

$$A_2 = [\frac{c}{2M} * A_1 - \dot{y}_0] \quad (2.42)$$

Hesaplanan A_1 ve A_2 değerleri yerlerine koyulduğunda elde edilen homojen çözümü denklem (2.43) ile gösterilmektedir.

$$y = M \frac{g}{K} - e^{-\frac{c}{2M}t} \left(M \frac{g}{K} - y_0 + \left[\frac{c}{2M} A_1 - \dot{y}_0 \right] t \right) \quad (2.43)$$

Genel çözümü sadeleştirmek için denklem takımları değişkenlere tanımlanarak denklem (2.44-2.47) arasında gösterilmektedir.

$$b_0 = M \frac{g}{K} \quad (2.44)$$

$$b_1 = -\frac{c}{2M} \quad (2.45)$$

$$b_2 = M \frac{g}{K} - y_0 \quad (2.46)$$

$$b_3 = \frac{c}{2M} A_1 - \dot{y}_0 \quad (2.47)$$

Denklem takımlarını değişkenlere atayarak sadeleştirilen denklem (2.48) ile gösterilmektedir.

$$y = b_0 - e^{-b_1 t} (b_2 + b_3 t) \quad (2.48)$$

2.3.3. Aşırı Sönümlü

Aşırı sönümlü cevabın genel çözümü denklem (2.49) ile gösterilmektedir.

$$y = M \frac{g}{K} + (A_1 e^{\lambda_1 t} + A_2 e^{\lambda_2 t}) \quad (2.49)$$

Aşırı sönümlü sistemin denklem kökleri denklem (2.50-2.51) ile gösterilmektedir

$$\lambda_{1,2} = -\frac{c}{2M} \pm \sqrt{\left(\frac{c}{2M}\right)^2 - \frac{K}{M}} \quad (2.50)$$

$$\lambda_{1,2} = -\frac{c}{2M} \pm w_d \quad (2.51)$$

Bu denklemin çözümünde başlangıç anında ki hız ve konum olan $y(0) = y_0$, $\dot{y}(0) = \dot{y}_0$ alındığında bağlı olan A_1 ve A_2 değerleri hesaplamak maksadıyla kullanılan denklemler (2.52-2.58) arasında gösterilmektedir.

$$y(0) = y_0 = M \frac{g}{K} + (A_1 + A_2) \quad (2.52)$$

$$A_2 = y_0 - \left[M \frac{g}{K} + A_1 \right] \quad (2.53)$$

$$\dot{y} = \lambda_1 A_1 e^{\lambda_1 t} + \lambda_2 A_2 e^{\lambda_2 t} \quad (2.54)$$

$$\dot{y}(0) = \dot{y}_0 = \lambda_1 A_1 + \lambda_2 A_2 \quad (2.55)$$

$$A_1 = \frac{\dot{y}_0 - \lambda_2 A_2}{\lambda_1} \quad (2.56)$$

$$A_2 = \frac{\dot{y}_0 + \left[y_0 - M \frac{g}{K} \right] \lambda_1}{2w_d} \quad (2.57)$$

$$A_1 = \frac{\left[y_0 - M \frac{g}{K} \right] \lambda_2 - \dot{y}_0}{2w_d} \quad (2.58)$$

Sistemin genel çözümü denklem (2.59) ile gösterilmektedir.

$$y = M \frac{g}{K} + (A_1 e^{\lambda_1 t} + A_2 e^{\lambda_2 t}) \quad (2.59)$$

Denklemin sade bir hal alması için denklem takımları değişkenlere atanmaktadır. Bu değişkenler (2.60 - 2.64) arasında verilmektedir.

$$b_0 = M \frac{g}{K} \quad (2.60)$$

$$b_1 = A_1 \quad (2.61)$$

$$b_2 = -\lambda_1 \quad (2.62)$$

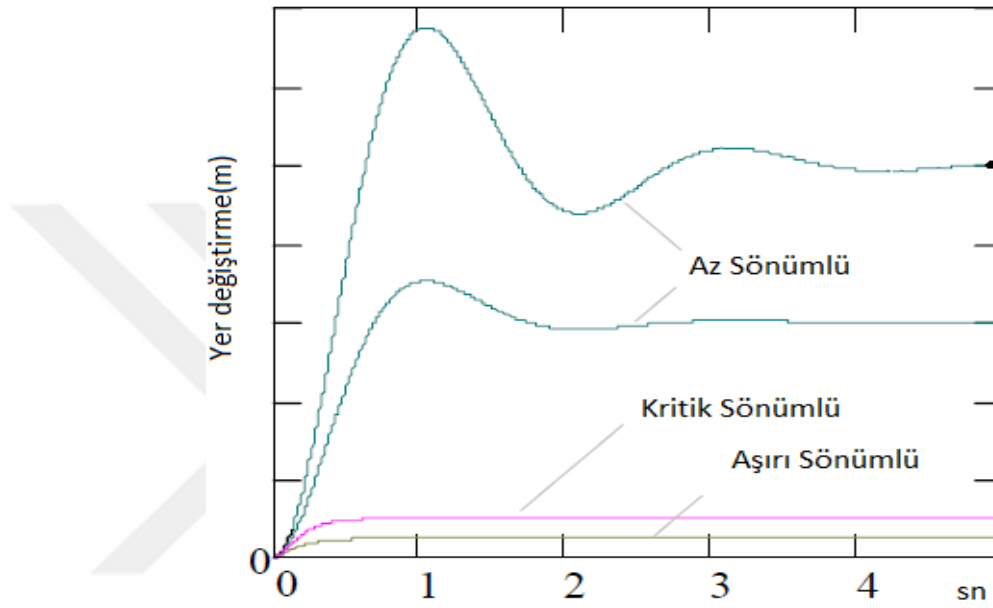
$$b_3 = A_2 \quad (2.63)$$

$$b_4 = -\lambda_2 \quad (2.64)$$

Sadeleştirme işlemleri sonucunda elde edilen genel çözüm denklem (2.65) ile gösterilmektedir.

$$y = b_0 + (b_1 e^{-b_2 t} + b_3 e^{-b_4 t}) \quad (2.65)$$

Sönümlenme sistem cevapları Şekil 2.4 ile gösterilmektedir.



Şekil 2.4 Ağırlık Ölçme Sistemi Birim Basamak Cevapları [16]

BÖLÜM 3

GEREÇ VE YÖNTEMLER

3.1. Optimizasyon

Matematiksel modelin çıkarılması mühendislik problemlerinin çözülmesinde en önemli hususlardan biridir. Çözüm için gerekli olan diğer bir husus ise eğitim bilgisidir. Mühendislik problemleri çoğunlukla lineer yapıda olmadığından matematiksel modelin çıkarılması her zaman mümkün olmadığı gibi model oluşturulsa bile eğitim bilgisinin hesaplanması maliyetli bir iş olabilmektedir. Sezgisel optimizasyon algoritmaları eğitim bilgisine gereksinim duymadan çözüm yapabilen yöntemlerdir. Bunun yanında hızlı cevap verme, hesaplama gücünün yüksek olması, probleme basit bir şekilde uygulanabilmesi ve farklı problemler için tekrar kullanılabilme yeteneklerinin bulunması sebebiyle bu problemlerde kullanılma oranı günden güne artmaktadır [61].

Çoğu varlık problemlerini çözebilmek için sosyal bir tabanı paylaşmaktadır. Birey olarak iş yapma becerisi düşük olan varlıkların sürü halinde zeki hareket edebilme özellikleri olduğu görülmektedir. Topluluğu oluşturan bireylerin iyi davranışları, topluluk içindeki en iyi bireyin davranışı ve kendi deneyimlerinden yararlanarak karşılaşacakları problemlere çözüm getirmektedir. Kısaca kaynak arama veya bir tehlike durumunda bireyler birbirleriyle haberleşmekte ve ortak bir tepki ile zeki çözümler üretebilmektedir [10,61].

Bu bölümde sezgisel optimizasyon algoritmalarından PSO, ABA ve EYA tanıtılacaktır.

3.2. Parçacık Sürü Optimizasyon Algoritması

3.2.1. Giriş

PSO bireyler arası işbirliğini temel alan popülasyon tabanlı sezgisel bir optimizasyon ve arama algoritmasıdır. Temel PSO algoritmasında pozisyon, hız(parçacığın bir sonrakini konumunu belirlemek için ona uygulanacak olan vektörel bir parametre), bilgiyi diğer bireylerle değiştirme, önceki pozisyonu hafızada tutma ve bir karar vermek için bilgiyi kullanma kabiliyetlerine sahiptir [62].

Uyarlamalı Kültür Modeli değerlendirme, karşılaştırma ve taklit etme olarak üç temel ilkeye dayanır ve parçacık sürülerinin temelini oluşturur [62].

Canlılık özelliğine sahip organizmaların en temel karakteristik davranışı uyarıyı hızlı bir şekilde değerlendirip uyarıya karşı tepki verme eğilimidir. Bu değerlendirme pozitif, negatif, itici veya çekici olarak yapılmaktadır. Öğrenme işlemi değerlendirme işlemi olmadan düşünülemez. Değerlendirme mekanizmasına sahip olan bir canlı mutlaka karşılaştırma kabiliyetine sahip olmalıdır. Uyarlamalı kültür modelindeki ve parçacık sürülerindeki bireyler, kendilerini sürü içerisindeki diğer bireylerle karşılaştırarak kendinden daha iyi sonuçlar elde etmiş komşularını taklit ederler. Taklit etme deneyim paylaşımı bakımından en basit yapıdır. Bu yapı gözlemle birlikte amaç gerçekleştirme ve zamanlama yeterliliğini de kapsar [62].

PSO algoritmasında parçacık olarak tanımlanan her bir birey, kendi en iyi pozisyonunu ve diğer bireylerden en iyi pozisyona sahip olan komşusunun deneyimini hesaba katarak bir sonraki pozisyonunu belirler [62].

3.2.2. Parçacık Sürü Optimizasyonu

Sezgisel yöntemlerden olan parçacık sürü optimizasyon algoritması sosyolog-psikolog James Kenedy ve elektrik-elektronik mühendisi olan Russel Eberhart tarafından 1995-1996 yıllarında kuş ve balık sürülerin hareketlerinden esinlenilerek numerik olmayan optimizasyon problemlerinin çözümü için ortaya atılmıştır. PSO popülasyon tabanlı stokastik bir optimizasyon algoritmasıdır [62-64].

PSO optimum sonuçları bulabilmek için öncelikli olarak rastgele düzende, önceden belirlenmiş veya her ikisi birlikte olarak aday çözümler sunan bireyler(parçacık) üretir. Bu bireylerin bir araya gelmesiyle sürü olarak adlandırılan popülasyon meydana gelir. PSO algoritmasında 20 birey(parçacık) genellikle iyi çözümü bulmak için yeterlidir, pratikte 2 ile 100 arasında parçacık kullanılabilir. Parçacık sayısı uyarlamalı olarak da ayarlanabilir. PSO bireyler arası bilgi paylaşımına dayanan bir yapıya sahiptir, her bir parçacık kendini sürüdeki en iyi pozisyona ayarlamaya çalışırken kendinin sahip olduğu en iyi pozisyon ve sürü içerisinde elde edilen en iyi pozisyon bilgisini kullanır.

3.2.2.1. PSO parametreleri

PSO algoritmasında parçacık adı verilen bireylerin sayısı, yani sürünün büyüklüğünü belirten parametre 's' harfiyle gösterilir. Sürü büyüklüğü algoritmanın adım sayısını doğrudan etkileyen parametrelerden biridir. Bu parametre 2 den büyük seçilir ve 20 parçacığa sahip popülasyon iyi bir sonuç elde etmek için yeterlidir.

Her parçacık indeks değeri ile gösterilir. İndeks değeri 'i' ile ifade edilir. Her parçacığa özel olarak üretilen bu değerlerin gösterimde, alt indis olarak parçacığa ait pozisyon, hız vb. parametreleri kullanılır. İndeks değeri olan 'i' 1 ile sürü büyüklüğü değeri arasındadır.

Her bir parçacığın pozisyon bilgisi ' x_i ' parametresiyle gösterilir. Her bir parçacığın hızını belirten parametre ' v_i ' ile gösterilir. PSO her bir parametrenin sahip olduğu en iyi pozisyonu tutar, bu en iyi pozisyon ' p_i ' parametresiyle gösterilir. En iyi pozisyon bilgilerinin kıyaslanmasıyla sürüde elde edilmiş en iyi pozisyon belirlenir, bu parametre ' p_{best} ' olarak gösterilir. Araştırmanın kaç boyutlu olarak yapılacağını belirleyen parametre ise 'd' ile gösterilir. Bu parametre optimize edilecek denklemin kaç bilinmeyen içerdiğini göstermektedir.

Parçacıkların sahip olduğu başlangıç değerleri belli bir aralıkta rastgele olarak önceden belirlenmiş değerler kullanılarak belirlenebilir.

Başlangıçta x_i ve v_i her parçacık için pozisyon ve hız değerleri rastgele veya tanımlanmış atama yapılarak belirlenir. Pozisyon değerleri her bir parçacık için en iyi

pozisyon olarak p_i değerlerine atanır, yani ilk olarak tanımlanan pozisyonlar parçacıklar için en iyi pozisyon olarak kabul edilir.

Algoritmanın ilerleyen adımlarında 'i' parçacığı bir sonraki pozisyonu belirlerken önceki adımlarda ziyaret etmiş olduğu en iyi pozisyondan yararlanır. Pozisyonun güncellenmesi için 'i' parçacığı tarafından ziyaret edilen güncel değerin 'f' ile maliyeti hesaplanır. Bu maliyet karşılaştırmasında yeni değerin daha iyi bir pozisyon olması durumunda pozisyon güncellenir, aksi durum gerçekleştiğinde ise pozisyon güncellenmez. PSO algoritmasında 'i' indeksli parçacığın pozisyon güncellemesi için kullanılan denklem (3.1) ile gösterilmektedir.

$$p_i(t+1) = \begin{cases} p_i(t), & f(x_i(t+1)) \geq f(x_i(t)) \\ x_i(t+1), & f(x_i(t+1)) < f(x_i(t)) \end{cases} \quad (3.1)$$

PSO algoritması her parçacığın sahip olduğu en iyi değeri, yani yerel en iyi pozisyonu tutmasının yanında bütün sürü içindeki en iyi pozisyon bilgisini de tutmaktadır. Global en iyinin tanımlandığı eşitlik denklem (3.2) ile gösterilmektedir.

$$p_{best}(t) \in \{p_0(t), p_1(t), \dots, p_s(t)\} | f(p_{best}(t)) = \min\{f(p_0(t)), f(p_1(t)), \dots, f(p_s(t))\} \quad (3.2)$$

Parçacığın her adımında yeni ve daha iyi bir pozisyona gitmek için hem sürünün en iyi tecrübesinden hem de kendi en iyi tecrübesinden yararlanır. Bunun yanında parçacığın bir sonraki pozisyonunu belirlemesi için hızlanma vektörü kullanılır. Yeni hızın ve yeni aday çözümün belirlendiği formüller denklem (3.3-3.4) ile gösterilmektedir

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 * r_{1,j}(t) * [p_{i,j}(t) - x_{i,j}(t)] + c_2 * r_{2,j}(t) * [p_{best}(t) - x_{i,j}(t)] \quad (3.3)$$

$$x_i = x_i(t) + v_i(t+1) \quad (3.4)$$

Yeni hız ve hız denklemine bağlı olarak yeni pozisyonun belirlenmesinden sonra aday çözüm daha iyi bir uygunluk değerine sahipse, aday çözüm yeni çözüm olacaktır.

$x_i(t)$: i parçacığının t anında sahip olduğu pozisyonu belirler.

$v_i(t)$: i parçacığının t anında sahip olduğu hızı belirler.

$p_i(t)$: parçacığın o ana kadar ki en iyi pozisyonunu saklar.

p_{best} : sürü içinde bulunan en iyi pozisyon bilgisini saklar.

PSO algoritmasında hız ve pozisyon bilgileri güncellenirken kullanılan algoritma üzerinde doğrudan etkili parametreler vardır. Bu parametreler sabit olarak belirlenebileceği gibi algoritma ilerledikçe iyileştirme yapılacak şekilde dinamik olarak da belirlenebilir.

PSO algoritmasında hızlanma adımlarının belirlenmesi için kullanılan iki parametre vardır. Bu parametreler ' c_1 ' ve ' c_2 ' karakterleri ile temsil edilir, hızlanma veya öğrenme katsayıları olarak da adlandırılır. Her bir parçacığın bir sonraki hızının belirlenmesinde aktif(dolaylı bir şekilde parçacığın bir sonraki pozisyonunun belirlenmesinde) rol oynarlar. Bu parametrelerin $c_1 + c_2 = 4$ ve $c_1 > c_2$ olarak seçildiği durumlarda daha iyi sonuçlar verdiği ortaya konulmuştur. Bu parametrelerin büyük seçilmesi parçacığın çok hızlanmasını ve en iyi pozisyon bilgisinde sapma olasılığının artmasına sebep olurken küçük olması algoritmanın çözüme ulaşması için gereken adım sayısını artırmaktadır.

Algoritmanın rastgele uzayda gezmesini sağlamak için ' r_1 ' ve ' r_2 ' parametreleri $[0,1]$ aralığında rastgele belirlenen sayılardır. Bir sonraki adımda kullanılacak olan hızın hesaplanmasında doğrudan kullanılır. Bu parametreler algoritmanın stokastik yapısını etkilemek için kullanılır.

PSO algoritmasında kullanılan diğer bir parametre parçacıkların arama aralığını belirleyen ' w ' değişkenidir. Bu değişkenin büyük seçilmesi algoritmanın global düzeyde arama yeteneğini artırırken yerel bölgede arama yeteneğini zayıflatır. Tam tersi olarak küçük seçilmesi algoritmanın yerel arama yeteneğini artırırken, global düzeyde arama yeteneğini zayıflatır. Bu durum göz önüne alınarak iki arama bölgesinde de iyi çalışabilecek değerlere başvurulmalıdır.

3.2.3. PSO Adımları ve Akış Diyagramı

PSO algoritmasının sahte kodu:

```

for: her i parçacığı için
    İlk atamasını rastgele veya önceden belirlenmiş olarak yap
end
do

```

for: her i parçacığı için
 Uygunluk değerini hesapla
 if i parçacığı için bulunan değer önceki değerinden daha iyi olduğu durumda
 yeni yerel iyi olarak belirle(p_i)
 else i parçacığı için bulunan değer öncekinden daha iyi değilse
 Yerel iyi değerinde değişiklik yapma

end

Belirlenen yerel en iyiler arasından en iyi pozisyon sahip olanı yeni global en iyi olarak atamasını yap (p_{best})

for: her i parçacığı için

Parçacığın sonraki adımdaki hız değerini denklem ve kısıtlara göre hesapla

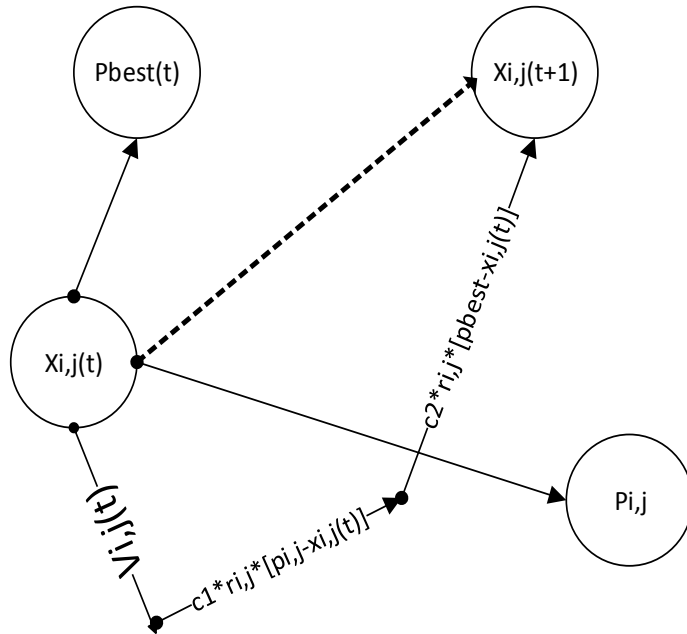
i parçacığının konumunu kısıtlara göre güncelle

end

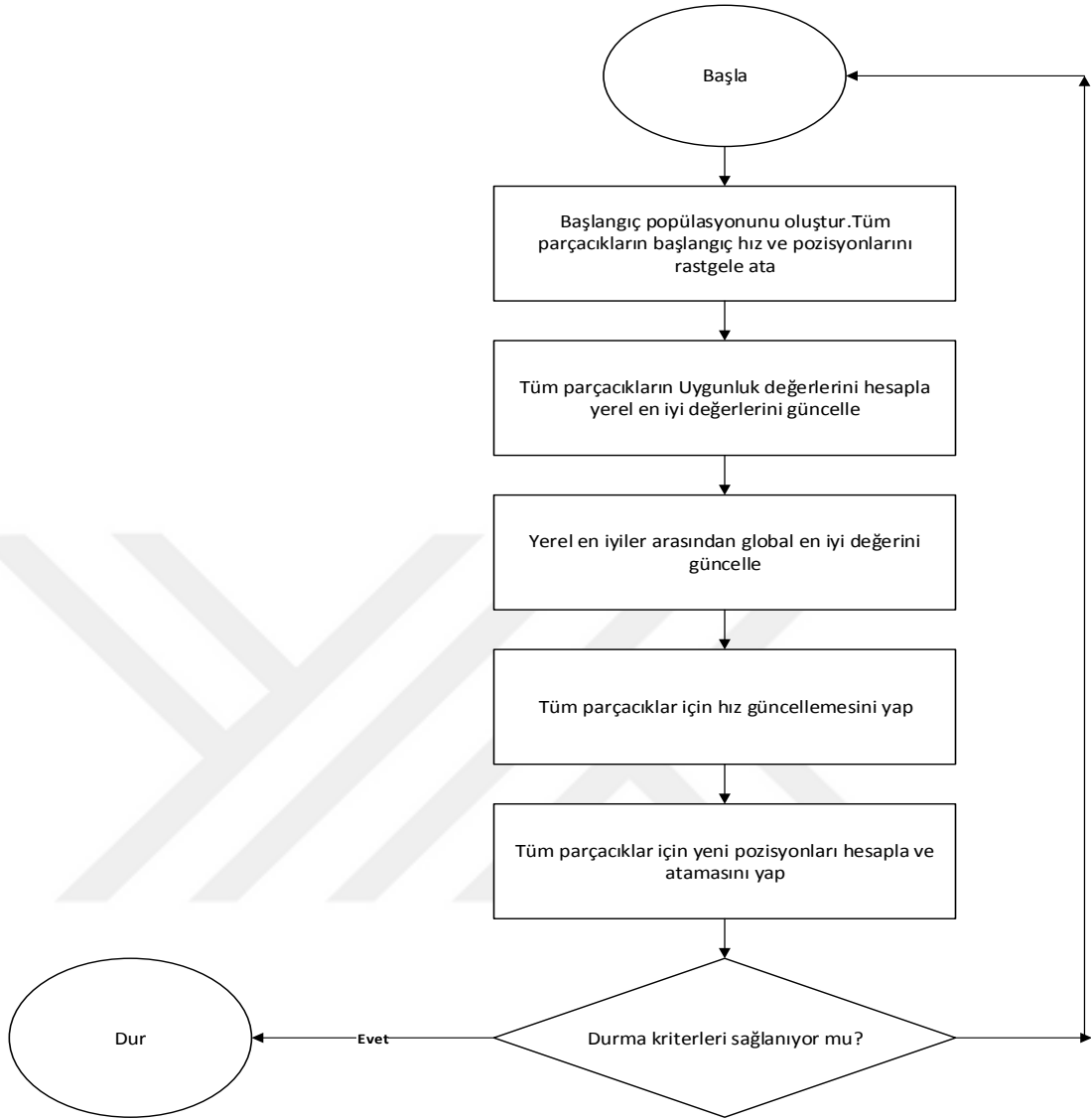
while durdurma kriterleri sağlanıyorsa dur

Bir parçacığın eski konumu ve en iyi konumu kullanarak pozisyonunun denklem (3.3) ve (3.4) ile güncellenme işlemi Şekil 3.1 ile verilen akış şematüğünde gösterilmektedir.

PSO algoritmasının akış diyagramı Şekil 3.1 ile gösterilmektedir.



Şekil 3.1 Parçacığın hareketi



Şekil 3.2 PSO akış diyagramı

3.2.4. PSO Örnek Çözüm

4 parçacığa sahip bir sürü için iki boyuta sahip bir uzayda $x_i(t)$ değişkenin tek iterasyon için çözümü incelenmektedir. Örnek çözüm için kullanılan birinci ($i = 1$) parçacığın ve diğer parçacıkların pozisyon değerleri $j=1, j=2$ boyutlarıyla aşağıda Şekil 3.3 ile gösterilmektedir. İlk hızları ve diğer başlangıç değerlerinin ataması da yapılmaktadır. Dördüncü parçacığın konumu (x_4) değeri $global(p_{best})$ en iyi olarak temsil edilmektedir. Birinci parçacığın lokal(p_1) en iyi değeri olarak verilmektedir. İki boyuta sahip ekseninde $j=1$ boyutu 'x' eksenini $j = 2$ boyutu 'y' eksenini temsil etmektedir.

Algoritmanın stokastik yapısına müdahale için kullanılan $[0,1]$ aralığında rastgele olarak belirlenen r_1 ve r_2 değerleri, öğrenme ve hızlanma parametreleri olarak belirlenen c_1 ve c_2 değerleri atanmıştır. Örnek çalışmada tek adımlık bir çözüm sunulacağından durdurma kriterleri verilmemiştir. Bu şartlar altında gerçekleştirilen bir adımlık çözüm aşağıda verilmektedir.

j_1 boyutu için:

$$v_{1,1}(t) = 2, r_{1,1} = 0.35, r_{2,1} = 0.25, c_1 = 2, c_2 = 2$$

$$v_{1,1}(t+1) = v_{1,1}(t) + c_1 * r_{1,1}(t) * [p_{1,1}(t) - x_{1,1}(t)] + c_1 * r_{2,1}(t) * [p_{best}(t) - x_{1,1}(t)]$$

$$v_{1,1}(t+1) = 2 + 2 * 0.35 * [1 - 1] + 2 * 0.25 * [4 - 1] = 1.5$$

$$x_{1,1}(t+1) = x_{1,1}(t) + v_{1,1}(t+1)$$

$$x_{1,1}(t+1) = 1.5 + 1 = 2.5$$

j_2 boyutu için:

$$v_{1,2}(t) = 2, r_{1,2} = 0.2, r_{2,2} = 0.3, c_1 = 2, c_2 = 2$$

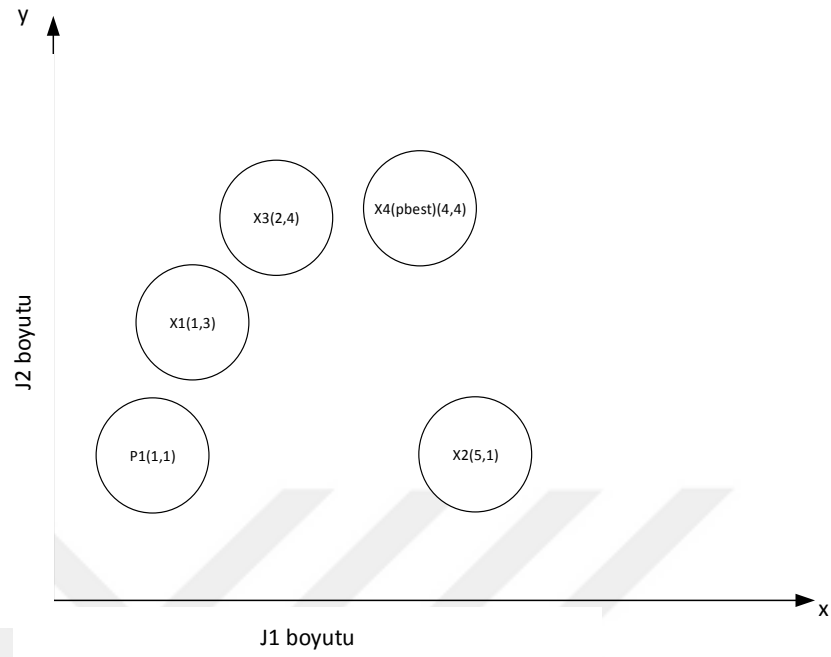
$$v_{1,2}(t+1) = v_{1,2}(t) + c_1 * r_{1,2}(t) * [p_{1,2}(t) - x_{1,2}(t)] + c_1 * r_{2,2}(t) * [p_{best}(t) - x_{1,2}(t)]$$

$$v_{1,2}(t+1) = 3 + 2 * 0.2 * [1 - 3] + 2 * 0.3 * [4 - 3] = 2.4$$

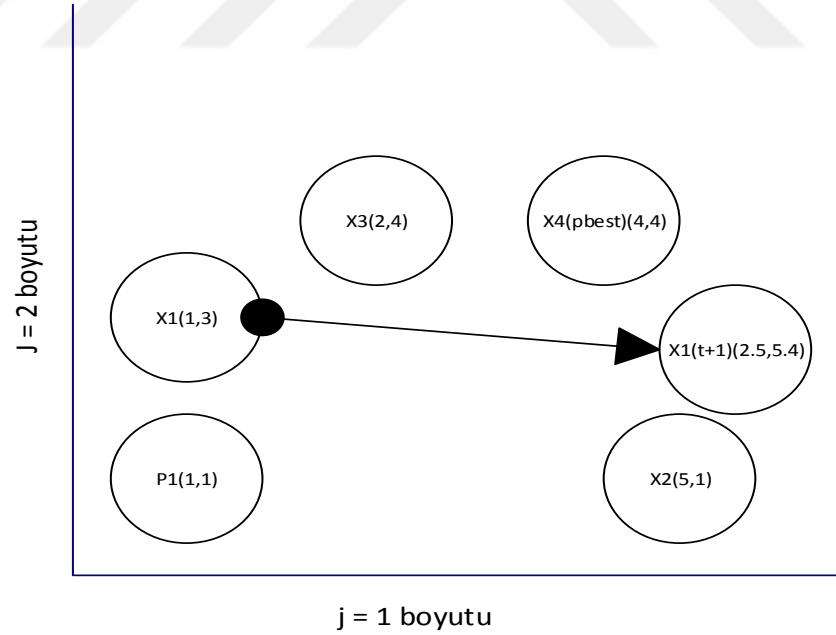
$$x_{1,2}(t+1) = x_{1,2}(t) + v_{1,2}(t+1)$$

$$x_{1,2}(t+1) = 3 + 2.4 = 5.4$$

Çözüm sonucu yeni yerel en iyi olarak kabul edilen değer için parçacığın hareketi Şekil 3.4 ile gösterilmektedir.



Şekil 3.3 Parçacıkların konumları



Şekil 3.4 Çözüm sonunda parçacığın güncellenen konumu

3.3. Ateş Böceği Algoritması

3.3.1. Giriş

Ateş böceği algoritması, Dr. Xin-She Yang tarafından 2007 yılında geliştirilen ve tropikal iklim bölgelerinde yaşayan ateş böceklerinin sosyal davranışlarını temel alan sürü zekası temelli bir optimizasyon algoritmasıdır [65]. Diğer sürü zekâsına dayalı algoritmalarla benzerlikler göstermesine rağmen kavranabilmesi ve uygulanabilirliği daha kolaydır. Ateş böceğinin ışık hareketi yapmasının ilk amacı diğer ateş böceklerini çekmektir. Bu hareket bir sinyal sistemi olarak düşünülebilir. Işık hareketlerinin altında yatan karmaşık biyokimyasal sürecin detayları ve gerçek amacı bilim dünyasında henüz kendisine tam olarak bir yer edinememiştir. Araştırmacılar göre, yanıp sönme şeklindeki bu ışık hareketi diğer bireyleri bulma, muhtemel avlarını çekme ve tehlikelerden korunmak için yapılmaktadır. Ateş Böceği algoritmasında, optimum çözümler elde etmek için verilen problemin uygunluk fonksiyonu, ateş böceklerinin daha çekici bireylere gitmelerine yardım eden ışık ya da ışık şiddeti ile ilişkilidir. Ateş böceklerinin tamamı aynı özelliklere sahip olan bireyler olarak kabul edilmektedir. Birbirilerini kendi bölgelerine çekmeleri bu algoritmanın temelini oluşturmaktadır. Bir ateş böceği parlaklık seviyesi diğer ateş böcekleri tarafından çekici bulunmalarıyla doğru orantılıdır. Bir birey kendisinden daha parlak bir ateşböceği olduğunu görürse ona doğru gider [65-67].

3.3.2. Ateş Böceği Algoritması

Ateşböceği Algoritmasının dayandığı üç temel prensip vardır:

1-Popülasyonun bütün bireyleri cinsiyetsizdir. Popülasyon içerisinde yer alan bütün bireyler birbirlerini etkileyebilir.

2-Bireyler arasında ki çekim, bireylerin parlaklığı ile doğru orantılıdır. Daha az parlaklığa sahip olan birey çok olana doğru hareket eder. Daha parlak bir bireyin bulunmaması durumunda ise hareket rastgele olarak yapılır. Çekicilik parlaklıkla doğru orantılı olmasının yanında aynı zamanda mesafe ile ters orantılıdır. Mesafe arttıkça çekicilik azalır.

3- Sürü içindeki bireylerin parlaklığı ve bu parlaklığının derecesi uygunluk fonksiyon değeri ile orantılıdır. Ateş böceği algoritmasının sahip olduğu iki önemli durum bulunmaktadır. Bunlar ışık yoğunluğu ve çekiciliğin formülasyonunun yapılmasıdır.

Işık yoğunluğunu 'I' ve iki ateş böceği arasında ki uzaklık 'r' ile ifade edilir. Işık kaynağına olan uzaklığın karesiyle ışığın yoğunluğu ters orantılı olarak değişmektedir ($I \propto 1/r^2$). Mesafe arttıkça ışığın yoğunluğu azalmaktadır. ' I_s ' ışık kaynağının yoğunluğunu gösteren değişkendir. Işığın yoğunluk formülü denklem (3.5) ile gösterilmektedir.

$$I = \frac{I_s}{r^2} \quad (3.5)$$

Işığın yoğunluğu ile birlikte havanın sahip olduğu ışık Emilimi söz konusudur. Mesafenin artmasıyla ışık kaynağı yoğunluğu azalmaktadır. Havanın ışık Emilimi katsayısı 'y' değişkeni ile ifade edilmektedir. Bunlara ek olarak $r = 0$ olduğu durumda $I = I_s/r^2$ ifadesinden kaçınmak gerekir. Işık yoğunluğunun alternatif formülü denklem (3.6) ile gösterilmektedir

$$I = I_0 * e^{-yr} \quad (3.6)$$

Teorik olarak 'y' katsayısı 0 ile $+\infty$ arasında değer alabilmektedir. Genel olarak katsayının değeri 1 olarak alınmaktadır.

Algoritmanın esas aldığı davranış olarak tanımlanabilecek ışık yoğunluğu ateş böceği ile nitelenen çözümün çekiciliği (β) ile doğru orantılıdır. Bu değer yüksek olması çözümün tercih oranını artırmaktadır. Çekicilik denklem (3.7) ile gösterilmektedir.

$$\beta = \beta_0 e^{-yr^2} \quad (3.7)$$

β_0 , $r = 0$ anında ateş böceğinin sahip olduğu çekiciliği temsil etmektedir. Üstel ifadelerin çözümü denkleme karmaşıklık getirebileceğinden, bu çözüme muadil olarak $1/(1+r^2)$ modeli ile denklem (3.8) çözümde kullanılabilir.

$$\beta = \frac{\beta_0}{1+y*r^2} \quad (3.8)$$

Bu varsayım uygulamalarda kolaylık sağlayabilmektedir. Mesafe 'r' ve 'y' değişkeni ile

‘ $r = 1/y$ ’ şeklinde ifade edilir, bu durum çekicilik denkleminde dikkate değer değişiklikler meydana getirir. Üstel olarak ifade edilen denklem (3.7) için $\beta = \beta_0$ doğrusal ifadesi düzenlendiğinde denklem $\beta = \frac{\beta_0}{2}$ şeklini alacaktır. Çekicilik fonksiyonunda yerine koyulduğunda, ‘ m ’ değişkeni 1’e eşit veya büyük olma şartını sağlamak kaydıyla denklem (3.9) ile ifade edilebilir.

$$\beta(r) = \beta_0 * e^{-ym}, m \geq 1 \quad (3.9)$$

‘ y ’ sabiti algoritma içinde karakteristik bir uzunluğu temsil etmekte olup denklem (3.10) ile gösterilmektedir.

$$\Gamma = y^{-1/m} \quad m \rightarrow \infty \quad (3.10)$$

‘ Γ ’ uzunluk birimi olarak optimizasyon problemine verilen değişken ‘ y ’ değişkeni formuna çevrilerek giriş değişkeni olarak kullanılabilmekte olup denklem (3.11) ile gösterilmektedir.

$$y = \frac{1}{\Gamma^m} \quad (3.11)$$

İki ateş böceği arasındaki mesafe kartezyen koordinatlarla ifade edilmek istenirse ‘ i ’ ve ‘ j ’ indisleri ateş böceklerini temsil eder. ‘ k ’ değişkeni i . veya j . ateş böceğinin içerisinde bulunduğu boyutu, ‘ d ’ değişkeni kaç boyuta sahip olduğunu göstermektedir. Burada boyuttan kasıt optimize edilecek değişken sayısını gösterirken, ‘ x ’ değişkeni i . veya j . ateş böceğinin konumunu göstermektedir. Mesafe denklemi (3.12) ile gösterilmektedir.

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (3.12)$$

‘ i ’ indisine sahip bir ateş böceğinin daha çekici olan ‘ j ’ indisine sahip bir ateşböceğine yaptığı hareket denklemi (3.13) ile gösterilmektedir.

$$x_i(t+1) = x_i(t) + \beta_0 * e^{-yr_{ij}^2} * (x_j(t) - x_i(t)) + \alpha * \varepsilon_i(t) \quad (3.13)$$

‘ α ’ değişkeni rastlantısallık parametresi olarak kullanılırken ‘ ε ’ değişkeni 0 ile 1 arasında rastgele bir vektör içermektedir.

3.3.3. Ateş Böceği Algoritması Adımları ve Şeması

Ateşböceği algoritmasının kodu ve adımları bu başlık altında gösterilmektedir.

Ateş Böceği Algoritmasının sahte kodu:

n_{pop} büyüklüğünde ateş böceğinin başlangıç popülasyonunun oluşturulması

Işık yoğunluğunu hesapla(I)

Işık emilim katsayılarını hesapla

while ($t < it_{max}$)

for $i=1:n$ (n_{pop})

for $j=1:n$ (n_{pop})

if ($I_i < I_j$)

(Ateş böcekleri i den j ye doğru hareket ederler.)

endif

Çekicilik, uzaklıkla(r) $\exp(-\gamma r^2)$ ile değişir

Yeni sonuçlar değerlendirilir ve ışık yoğunluğunu güncelle

endfor

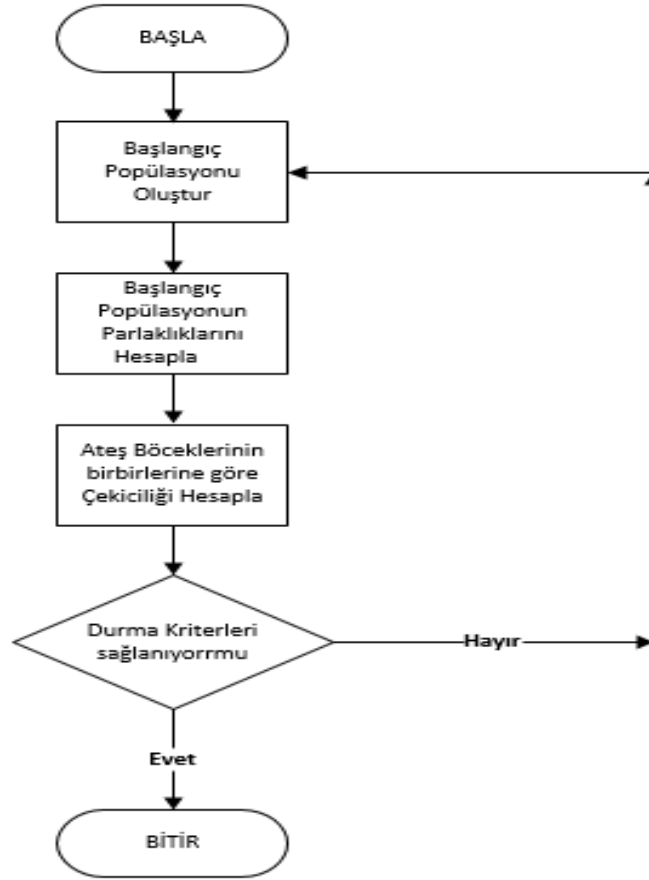
endfor

endwhile

Ateş böcekleri sıraya konulur ve güncel global en iyi bulunur

Son işlem sonrası sonuçlar ve simülasyon çıkışı

Ateşböceği algoritmasının akış diyagramı Şekil 3.5 ile gösterilmektedir.



Şekil 3.5 Ateş Böceği Algoritması akış diyagramı

3.3.4. Ateş Böceği Örnek Çözüm

1. Başlangıç parametrelerinin oluşturulması gerekmektedir.

Uygunluk fonksiyonu: Algoritmada bulunan çözümlerin iyi olup olmadığını deneyebilmek için kullanılan fonksiyondur. Örnek çözüm için kareler fonksiyonu belirlenmiştir. Kareler fonksiyonu denklem (3.14) ile gösterilmektedir.

$$\sum_{i=1}^n x_i^2 \quad (3.14)$$

Popülasyon boyutu(p): Popülasyondaki ateş böceğinin toplam sayısıdır. Bir adımda bulunan çözüm sayısı olarak da ifade edilebilir. ($n_{pop}= 5$)

Parametre boyutu(d) : Uygunluk fonksiyonu üzerinde bilinmeyen parametre sayısıdır.

(d = 2)

Parametre aralığı(ub, lb): Bilinmeyen parametrelerin sayısını ifade etmektedir. ‘ub’ değişkeni üst sınırı ‘lb’ alt sınırı belirlemektedir. $ub = 10$, $lb = -10$.

Maksimum adım sayısı (it_{max}): Algoritmanın maksimum olarak kaç adım koşaacağını göstermektedir. Bir adım için örnek çözüm yapılacaktır.

Alpha(α): Rasgelelik değişkeni 0 ile 1 arasındadır. Çözülen örnek için $\alpha = 0.2$ olarak alınacaktır.

Gama(γ): Sabit emilim katsayısı 0.01 ile 100 arasında alınabilir. Işığın hava ortamındaki emilim miktarı olarak değerlendirilebilir. Örnekte $\gamma = 1$ olarak kullanılacaktır.

Başlangıç parametrelerinin belirlenmesi işlemini bitirdikten sonra algoritmanın temel adımlarına geçmeden önce başlangıç konumlarının rastgele veya önceden tanımlı olarak verilmesi gerekmektedir. Çözömlenen örnekte ilk konumları rastgele olarak belirlenmektedir.

2. Ateş böceklerine rastgele konumlarının belirlenmesi

En büyük ve en küçük parametre boyutuna göre rastgele bir çözüm kümesi oluşturulur.

‘i’ çözüm kümesinin indeksi, ‘K’ seçilen çözüm kümesinde seçilen parametrenin indeks numarasıdır.

$$x_{ik} = lb_j + rand(0,1) * (ub_j - lb_j)$$

‘ x_{ik} ’ seçilen çözümün geçerli parametresi olarak alınmaktadır.

Çözüm içinde kullanılan rastgele sayıları ‘rand’ vektörü ile gösterilmektedir.

$$rand = [0.8147 \quad 0.9058 \quad 0.1270 \quad 0.9134 \quad 0.6324 \quad 0.0975 \quad 0.2785 \quad \dots$$

$$\dots 0.5469 \quad 0.9575 \quad 0.9649]$$

$$rand_1 = 0.8147$$

$$rand_2 = 0.9058$$

Rastgele sayılar oluşturabilmek için gereken iki noktayı, başlangıç koşullarını belirleyen formül kullanılarak bulunabilmektedir.

$$x_{1,1} = -10 + rand_1 * (10 + 10) = 6.2945$$

$$x_{1,2} = -10 + rand_2 * (10 + 10) = 8.1158$$

Diğer çözümler için gereken noktalar bulunarak yerlerine koyulmaktadır.

$$x_1 = 6.2945 \quad 8.1158$$

$$x_2 = -7.4603 \quad 8.2675$$

$$x_3 = 2.6472 \quad -8.0492$$

$$x_4 = -4.4300 \quad 0.9376$$

$$x_5 = 9.1501 \quad 9.2978$$

Konumların belirlenmesinden sonra uygunluk fonksiyonuna göre noktaların maliyetlerini 'sphere' fonksiyonu kullanılarak hesaplanmaktadır.

$$u_1 = x_{1,1}^2 + x_{1,2}^2 = 105.4872$$

$$u_2 = x_{2,1}^2 + x_{2,2}^2 = 124.0074$$

$$u_3 = x_{3,1}^2 + x_{3,2}^2 = 71.7971$$

$$u_4 = x_{4,1}^2 + x_{4,2}^2 = 20.5044$$

$$u_5 = x_{5,1}^2 + x_{5,2}^2 = 170.1735$$

Uygunluk değerleri 'u' değişkeniyle gösterilmektedir. Hesaplanan uygunluk değerleri karşılaştırmalarda kullanılmak üzere saklanır.

Örnek çözümde bütün uygunluk değerleri birbirleriyle karşılaştırılmayacaktır. Ateş böceğinin hareketinde karşılaşılabilecek iki durum incelenmektedir. Sırasıyla; daha parlak bir ateş böceğinin bulunmaması ve daha parlak bir ateş böceğine hareket durumları gösterilmektedir.

- Durum 1: daha yüksek çözüm kalitesine sahip ateş böceğinin bulunmaması,

Uygunluk değerlerine bakılarak 4 ve 5. çözüm karşılaştırıldığında $u_4 < u_5$ olarak görünür ve kendinden daha parlak bir ateş böceğinin olmadığını göstermektedir. Bu durumda ateş böceğinin hareketi rastgele olacaktır.

$$x_4 = [-4.4300 \quad 0.9376]$$

$$x_{i,k} = x_{i,k} + a * \varepsilon_{i,k}$$

Örnek çözüm iki parametre için yapıldığından k değişkeninin alacağı iki farklı değer vardır.

$$k = 1,2$$

$$\varepsilon_{i,k} = rand(0,1) - 0.5$$

‘ ε ’ değişkenin belirlenmesi için rastgele iki sayıya ihtiyaç duyulmaktadır.

$$r = [0.8 \quad 0.5]$$

$$\varepsilon_{4,1} = 0.8 - 0.5 = 0.3$$

$$\varepsilon_{4,2} = 0.5 - 0.5 = 0$$

Rastgele sayıların üretilmesinden sonra yeni konumun hesaplanması işlemi yapılmaktadır.

$$x_{4,1} = -4.43 + 0.2 * 0.3 = -4.37$$

$$x_{4,2} = 0.9376 + 0.2 * 0 = 0.9376$$

$$x_4 = [-4.37 \quad 0.9376]$$

Yeni konumun hesaplanmasından sonra artık uygunluk değeri hesaplanarak yeni çözüm elde edilebilir.

$$u_4 = x_{4,1}^2 + x_{4,2}^2 = 19.976$$

- Durum 2: karşılaştırmada daha iyi bir parlaklığın bulunması durumu

Bu durumda 2. Çözüm ile 3. Çözüm vektörleri karşılaştırıldığında $u_2 > u_3$ durumu göz önünde bulundurularak, u_3 çözümünün daha iyi olduğu görülmektedir. Bu durumda,

$$x_2 = [6.2945 \ 8.1158]$$

$$x_3 = [2.6472 \ -8.0492]$$

İkinci ateş böceğinin üçüncü ateş böceğine hareketini gerçekleştirmek için ateş böceğinin çekiciliğinin hesaplanması gerekir.

$$x_i(t+1) = x_i(t) + \beta_0 * e^{-yr_{ij}^2} * (x_j(t) - x_i(t)) + \alpha * \varepsilon_i(t)$$

$$\beta = \frac{\beta_0}{1 + y * r^2}$$

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} = r_{12} = \sqrt{\sum_{k=1}^d (x_{1k} - x_{2k})^2}$$

$$= \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2} = \sqrt{(6.2945 - 8.1158)^2 + (2.6472 - (-8.0492))^2}$$

$$= 8.2119$$

Ateş böceklerinin arasındaki mesafe ve uygunluk değeri ile ateş böceğinin cazibesi hesaplanmaktadır.

r(iki ateş böceği arasındaki mesafe) = 8.2119

$$\beta_0 = 124.0074$$

$$\beta = \frac{124.0074}{1 + 1 * 8.2119^2} = 1.81203$$

$$k = 1,2$$

$$rand = [0.2 \ 0.6]$$

$$\varepsilon_{1,1} = 0.2 - 0.5 = -0.3$$

$$\varepsilon_{1,2} = 0.6 - 0.5 = 0.1$$

Rastgelelik değişkenlerini hesapladıktan sonra hareket işlemi tamamlanmaktadır.

$$6.2945 \ 8.1158$$

$$x_{1,1} = 6.2945 + 1.81203 * (6.2945 - 2.6472) + (0.2 * -0.3) = 12.84317$$

$$x_{1,2} = 8.1158 + 1.81203 * (8.1158 - (-8.0492)) + (0.2 * 0.1) = 37.427265$$

$$x_2 = [12.84317 \ 37.427265]$$

Görüldüğü üzere parametreler üst sınırı aşmaktadır. Sınırların kontrolü yapılarak tekrardan belirlenen üst sınırlar parametrelere atanacaktır.

3. Sınırların Kontrolü

Bu işlem sonrasında çözümlerin belirlenen sınırlar arasında kalması için kontrol etme işlemi yapılmaktadır. Bu adım isteğe bağlı olarak yapılır. Taşma problemi olarak adlandırılan bu problem için birçok çözüm yöntemi geliştirilmiştir. Bunlardan en basiti sınır aşıldığında sınırın aşılacak değişkene tekrar atanmasıdır.

3.4. Emperyalist Yarışmacı Algoritma

3.4.1. Giriş

Emperyalist Yarışmacı Algoritma (EYA) sosyal tabanlı sezgisel bir yöntemdir. EYA ilk olarak 2007 yılında Atashpaz Gargari ve Lucas tarafından önerilmiştir [68]. Bu algoritmanın avantajı yakınsamanın hızlı olması ve global optimizasyonda yüksek arama yeteneğine sahip olmasıdır [69,70]. Bu yöntem emperyalist rekabetin sosyo-politik sürecine dayanır.

3.4.2. Emperyalist Yarışmacı Algoritma

Algoritma rastgele bir popülasyonla başlatılır ve popülasyonun her üyesi bir ülkeyi temsil eder. Popülasyon içinden en iyi uygunluk değerine sahip bazı ülkeler,

emperyalist ülkeler olarak atanır ve diğer ülkelerde bu emperyalist devletlerin kolonileridir. Başlangıç nüfusunda koloni olarak kalan ülkeler, maliyet fonksiyonundan gelen uygunluk değerlerine bakılarak emperyalistler arasında sırayla bölüşülür. Örneğin emperyalist devletlerin sayısı beştir. Bu durumda, altıncı ile onuncu koloni ülkeleri emperyalist ülkelerin ilk kolonileri olarak belirlenecektir. Bundan sonraki on birinci ülkeden on beşinci ülkeye kadar sırasıyla emperyalistlerin ikinci kolonileri atanır. Bu eylem koloni haline getirilecek devlet kalmayana kadar devam eder. Bu örnek Şekil 3.6 ile gösterilmektedir.

İmparatorlukların ayrı ayrı normalleştirilmiş maliyetleri hesaplanır, ' M_n ' n. imparatorluğun normalleştirilmiş maliyetini temsil eden değişkendir. ' M_n ' nin hesaplanması denklem (3.15) ile gösterilmektedir, $\max_i\{c_i\}$ değişkeni emperyalistler arasındaki en yüksek maliyete sahip olan ülkeyi temsil etmektedir. m_n değişkeni ise n. emperyalistin maliyetini gösterir.

$$M_n = m_n - \max_i\{c_i\} \quad (3.15)$$

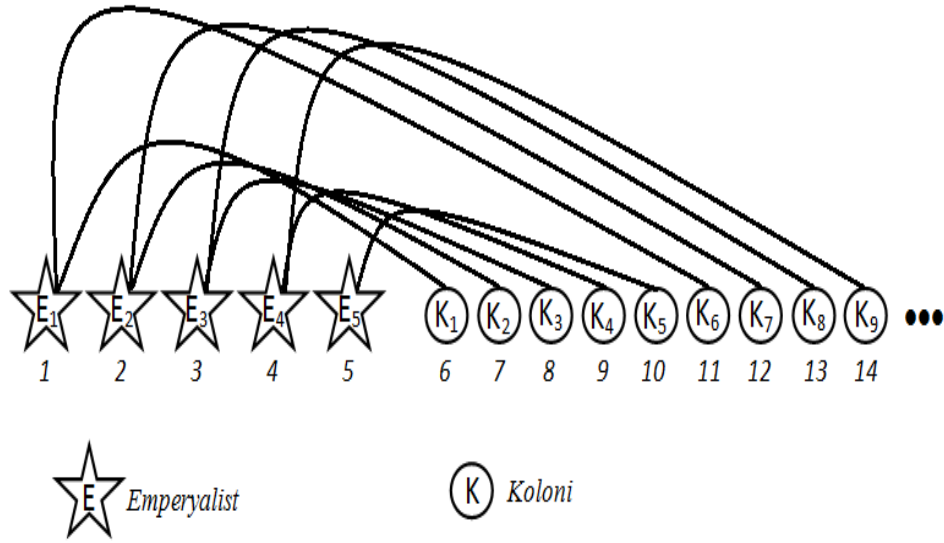
Her bir imparatorluk için normalleştirilmiş maliyetlerin hesaplanmasından sonra her bir emperyalistin (P_n) normalleştirilmiş gücü denklem (3.16) ile hesaplanır. ' N_{emp} ' değişkeni emperyalistlerin toplam sayısını göstermektedir.

$$P_n = \left| \frac{M_n}{\sum_{i=1}^{N_{emp}} M_i} \right| \quad (3.16)$$

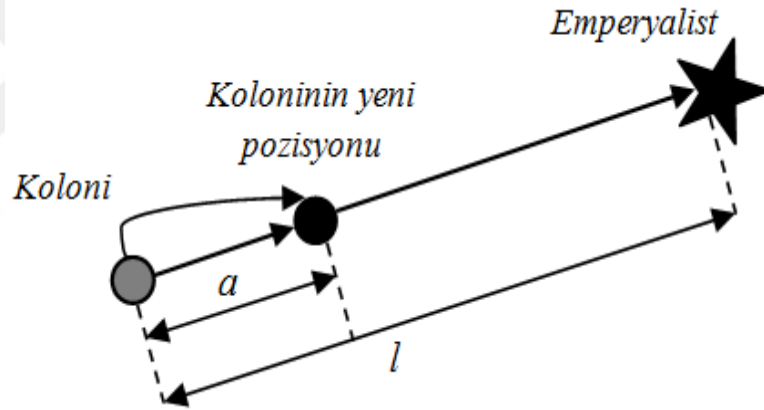
Emperyalist ülkelerin ilgili adım için sahip olduğu koloni sayısının belirlenmesi işlemi denklem (3.17) ile gösterilmektedir. NM_n değişkeni n. emperyalistin ilgili adımda sahip olduğu koloni sayısını gösterir. N_{col} değişkeni ise nüfus içerisindeki toplam koloni sayısını gösterir.

$$NM_n = \text{round} \{P_n \cdot (N_{col})\} \quad (3.17)$$

İmparatorluklara bağlı olan koloniler kendi emperyalist devletlerine doğru hareket etmeye başlar ve bu işlem sonucunda yeni bir pozisyona sahip olurlar. Bu hareket Şekil 3.7 ile gösterilmektedir.



Şekil 3.6 Emperyalistlerin kolonilerle ilişkisi



Şekil 3.7 Kolonilerin Emperyalistlere doğru hareketi

$$a \sim U(0, \beta \times l) \quad (3.18)$$

‘ a ’ değişkeni koloninin yaptığı hareketin mesafesini gösteren değişkendir. ‘ l ’ değişkeni koloni ile emperyalist arasındaki mesafedir. β değişkeni ‘ a ’ değerine uygulanan bir katsayıdır ve genellikle 1 den büyük seçilir. Yönelim mesafesi için kullanıcıya veya probleme bağlı olarak rastgele sapma miktarları eklenebilir. ‘ a ’, ‘ l ’ ve ‘ β ’ arasındaki ilişkiyi gösteren denklem (3.18) ile verilmektedir.

Koloni hareketlerinin tamamlanmasının ardından bir koloni üyesi olduğu emperyalistten daha iyi uygunluk değerine sahip olursa imparatorluk ve koloni yer değiştirmektedir.

Bu adımdan sonra ise imparatorluklar arası rekabet başlar. Bu aşamada en zayıf imparatorluğun kolonisini imparatorluktan ayrılır ve daha güçlü bir imparatorluğa katılır. Hangi imparatorluk daha güçlüyse koloniyi o alacaktır. Bütün kolonilerini kaybeden imparatorluklar koloniye dönüşerek bir imparatorluğun üyesi haline gelecektir. Algoritmada durma koşulu sağlanana kadar bu işlemler devam etmektedir [71].

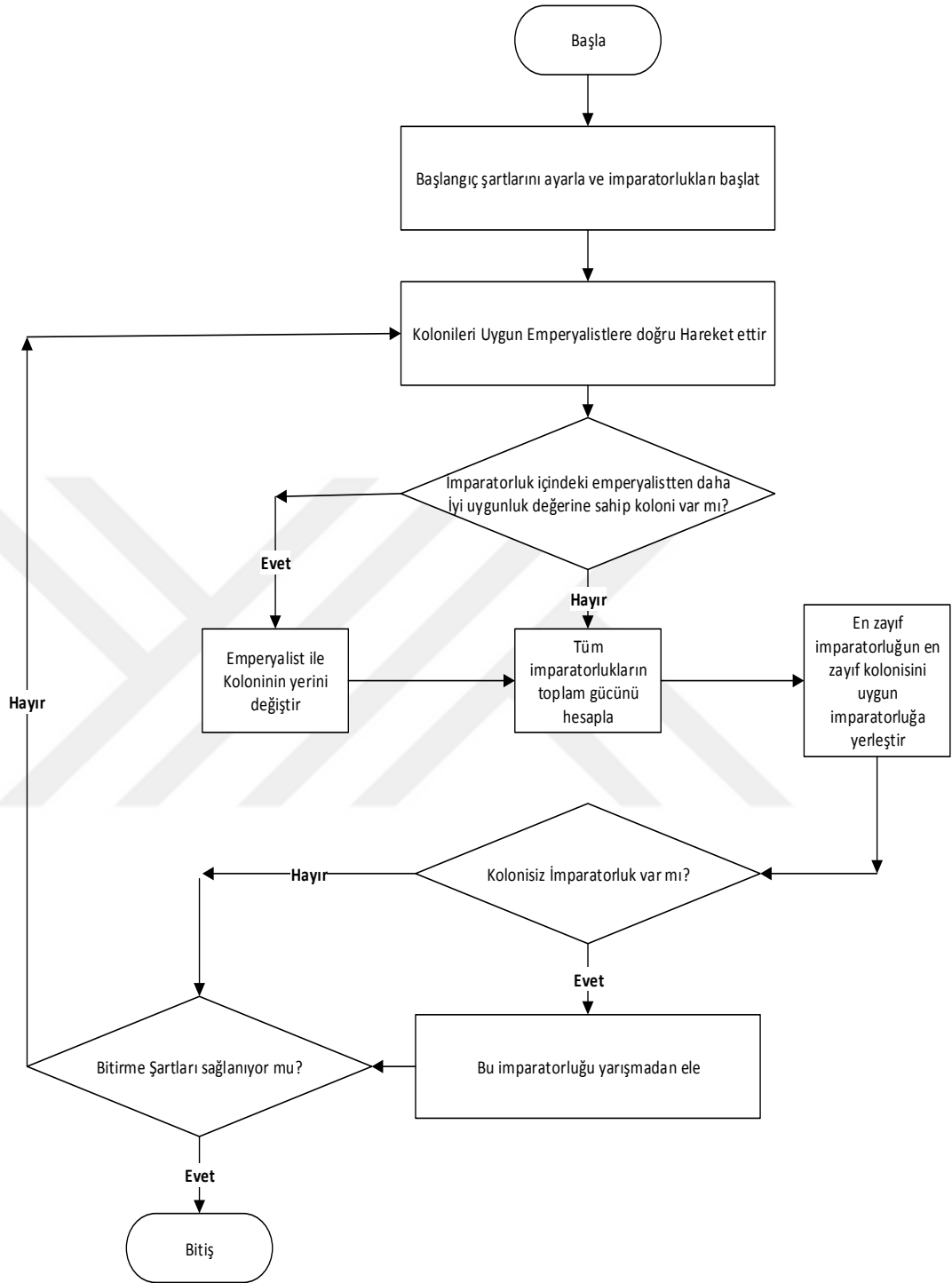
Emperyalist rekabetin sonucu, güçlü imparatorlukların toplam gücünde artış yaşanırken rekabet sonucu güçsüz kalan imparatorluklar yarıştan elenecektir. EYA' nın akış diyagramı Şekil 3.8 ile gösterilmektedir [72].

Algoritmanın formülleri örnek çözüm ile verilmektedir.

3.4.3. Emperyalist Yarışmacı Algoritma Adımları

Emperyalist Yarışmacı Algoritmanın adımları:

- 1) Başlangıç şartlarını ayarla ve imparatorlukları başlat,
- 2) Kolonileri emperyalistlere doğru hareket ettir (Assimilation),
- 3) Bazı kolonilerin yerlerini rastgele olarak değiştir (Revolution),
- 4) İmparatorluk içindeki kolonilerden birisi emperyalistten daha iyi bir sonuca sahip ise emperyalist ile koloniyi yer değiştir,
- 5) Benzer imparatorlukları birleştir,
- 6) İmparatorlukların toplam gücünü hesapla,
- 7) En zayıf imparatorluk içerisindeki en zayıf koloniyi imparatorluktan ayırarak uygun koloniye ata(Imperialistic competition),
- 8) Güçsüz olan imparatorluğu ele,
- 9) Durma şartları sağlanıyorsa dur yoksa 2. adıma dön.



Şekil 3.8 Emperyalist Yarışmacı Algoritma akış diyagramı

3.4.4. Emperyalist Yarışmacı Algoritma Örnek Çözüm

İlk önce emperyalistlerin oluşumu işlemi gerçekleştirilmektedir. EYA içerisinde kullanılan popülasyonun bireyleri ülke(country) olarak adlandırılır. Popülasyonun

bireylerini oluşturmak için tamamen rastgele olarak üst ve alt sınırlar içerisinde oluşturulur. Çözümler maddeler halinde gösterilmektedir.

1- Başlangıç popülasyonunun oluşturulması

Başlangıç koşullarını oluşturmak için başlangıçta kullanılacak parametrelerin değerlerinin verilmesi gerekmektedir.

- Bilinmeyen sayısı, $d = 1$,
- Parametrenin alt sınırı, $lb = 0$,
- Parametrenin üst sınırı, $ub = 10$,
- Emperyalist devlet sayısı, $n_{emp} = 3$,
- Koloni devlet sayısı, $n_{col} = 7$,
- Toplam popülasyon sayısı, $n_{pop} = 10$ olarak belirlenmektedir.

Başlangıç parametrelerine değerler verildikten sonra başlangıç popülasyonu oluşturulur. Ülkelerin ilk konumları rastgele olarak belirlenir. Başlangıç konumları denklem (3.19) ile hesaplanmaktadır.

$$x_i = \text{rand}(lb, ub, d) \quad (3.19)$$

Popülasyon sayısı 10 olarak belirlenmesinden sonra 10 adet çözüm denklem (3.19) ile gerçekleştirilmektedir.

$$x = [2.45 \ 9.37 \ 6.87 \ 9.30 \ 4.18 \ 8.14 \ 4.40 \ 7.62 \ 0.09 \ 9.95]$$

Konumların belirlenmesinden sonra uygunluk fonksiyonu yardımıyla uygunluk değerleri hesaplanır. 'sphere' uygunluk fonksiyonu olarak kullanılmaktadır.

$$p_i = [6.02 \ 87.89 \ 47.21 \ 86.59 \ 17.47 \ 66.25 \ 19.36 \ 58.14 \ 0.01 \ 99.01]$$

Uygunluk değerlerinin belirlenmesinden sonra en küçük uygunluk değerine sahip üç ülke emperyalist ülkeler olarak seçilir. Belirlenen emperyalist devletlerin konumları denklem (3.20) ile belirlenmektedir:

$$emp_i = \min_j(x_j) \quad (3.20)$$

$$emp_i = [0.09 \ 2.45 \ 4.18]$$

Emperyalistlerin uygunluk değerleri:

$$p_{emp,i} = [0.01 \ 6.02 \ 17.47]$$

Emperyalist devletlere koloniler güçlerine göre atanmaktadır. Atanan koloniler sırasıyla 5 koloni, 2 koloni ve 0 koloni olarak belirlenmiştir.

emp_1 'e ait koloniler:

$$col_1 = [4.40 \ 6.87 \ 9.30 \ 9.37 \ 9.95]$$

emp_1 'e ait kolonilerinin uygunluk değerleri:

$$p_{col,1} = [19.36 \ 47.21 \ 86.59 \ 87.89 \ 99.09]$$

emp_2 'e ait koloniler:

$$col_2 = [7.62 \ 8.14]$$

emp_2 'e ait kolonilerinin uygunluk değerleri:

$$p_{col,2} = [58.14 \ 66.25]$$

emp_3 başlangıç kolonilerinin atanmasında koloni alamamıştır.

Bu adımlardan sonra toplam uygunluk hesaplanmaktadır. Toplam uygunluk değeri denklem (3.21) ile hesaplanmaktadır.

$$\begin{cases} p_{total,i} = p_{emp,i} + \varepsilon * \frac{(\sum_j^{col} p_{col,j})}{col} & col > 0 \\ p_{total,i} = p_{emp,i} & col = 0 \end{cases} \quad (3.21)$$

Toplam uygunluk değeri sadece emperyalist devletler için hesaplanır.

$$p_{total} = [13.61 \ 18.46 \ 17.47]$$

2- Kolonilerin emperyalistlere doğru hareketi

Emperyalistlerin kolonileri atandıktan sonra iyileştirme işlemleri yapılır. İyileştirme işlemi denklem (3.22) ile gösterilmektedir.

$$col_i = col_i + b * r * (emp_j - col_i) \quad (3.22)$$

emp_1 'e ait kolonilerin emperyaliste doğru hareketi sonucu elde edilen konumlar:

$$col_1 = [0 \ 4.09 \ 5.58 \ 8.68 \ 6.93]$$

Konumların güncellenmesiyle birlikte uygunluk değerleri de uygunluk fonksiyonu aracılığı ile güncellenmektedir. Güncellenen uygunluk değerleri:

$$p_{col,1} = [16.77 \ 20.96 \ 75.33 \ 47.93]$$

emp_2 'e ait kolonilerin emperyaliste doğru hareketi sonucu elde edilen konumlar:

$$col_2 = [3.56 \ 6.11]$$

Konumların güncellenmesiyle birlikte uygunluk değerleri de uygunluk fonksiyonu aracılığı ile güncellenmektedir. Güncellenen uygunluk değerleri:

$$p_{col,2} = [12.69 \ 37.40]$$

emp_3 'e ait koloni bulunmamaktadır.

3- Devrim Yapılması(Revulotion)

Bu adımda emperyalist devletler kendilerinden daha iyi bir rastgele çözümle değiştirilir. Değiştirme işleminde uygunluk değeri daha iyi olan rastgele bir çözüm üretildiyse emperyalist devlet ile o çözüm yer değiştirmektedir. Örnekte üçüncü emperyalist devlet daha iyi olan rastgele bir çözümle değiştirilmektedir.

$$emp_i = [0.09 \ 2.45 \ 3.96]$$

$$p_{emp,i} = [0.01 \ 6.02 \ 15.69]$$

Kolonilerde herhangi bir değişiklik olmamıştır.

4- İmparatorlukla koloninin yer deęiřtirmesi

Emperyalist devletlerin kolonileri arasındaki herhangi bir koloni, emperyalist devletten daha iyi uygunluk deęerine sahipse koloni devlet emperyalist, emperyalist devlet ise koloni devleti olur. Örnekte birinci emperyalist devletin birinci kolonisi daha iyi uygunluk deęerine sahip olduęu için yeni emperyalist olarak atanmıřtır. Deęiřimden sonra emperyalist ve koloniler:

$$emp_1 = 0$$

$$p_{emp,1} = 0$$

Emperyalistin kolonileri:

$$col_1 = [0.9 \ 4.09 \ 4.58 \ 8.68 \ 7.53]$$

olarak güncellenmektedir. Emperyalist devlet artık kolonini yerini almıřtır. Bu iřlem sonucunda kolonilerin uygunluk deęerleri:

$$p_{col,1} = [0.01 \ 16.77 \ 20.96 \ 75.33 \ 9.76]$$

olarak güncellenmektedir.

emp_2 ve emp_3 olarak ifade edilen deęerlerde herhangi bir deęiřim olmamıřtır.

5- Toplam uygunluk deęerlerinin güncellenmesi

Emperyalist devletlerin toplam uygunluk deęerleri güncellenmektedir. Güncel toplam uygunluk deęerleri:

$$p_{total} = [6.79 \ 11.03 \ 15.69]$$

olarak güncellenmektedir.

6- Emperyalistler Arası Yarıř

Burada iki durum söz konusudur. 1. durumda en güçsüz emperyalistin kolonisi varsa bu koloni ile dięer emperyalistler bir yarıřa sokulur. Kazanan emperyalist bu zayıf koloniyi bünyesine katar.

Emperyalist devletin kolonisi yoksa bu emperyalist kazanan yarışı kazanan emperyalistin kolonisi olur.

Yapılan örnekleme üçüncü emperyalistin kolonisi olmadığı için yarışmada kazanan emperyalistin kolonisi olmuştur. Birinci emperyalist yarışmayı kazanmıştır. Üçüncü emperyalist devlet artık birinci emperyalist devletin kolonisi haline gelmiştir.

$$col_1 = [0.09 \ 4.09 \ 4.58 \ 8.68 \ 7.53 \ 3.96]$$

Bu durumda birinci imparatorluğa ait kolonilerin uygunluk değerleri güncellenmiştir.

İkinci emperyalist devletinde herhangi bir değişiklik olmamıştır. Artık üçüncü emperyalist devleti de bir koloni haline gelmiştir.

BÖLÜM 4

SEZGİSEL OPTİMİZASYON ALGORİTMALARININ AĞIRLIK ÖLÇME SİSTEMİNDE KULLANILMASI

Sistem kimliklendirme süreci ilk sinyalin alınmasından itibaren geçen süre boyunca giriş ve çıkış tepkilerinin gözlemlenmesiyle yapılmaktadır. Kullanılan sinyaller; dürtü, adım, rampa, sinüzoidal veya rastgele olarak seçilebilmektedir [45,47,55]. Sinyalin işlenebilmesi için analog-dijital dönüştürücü vasıtasıyla toplanan veriler veya sistem modelinin bilinmesi durumunda simülasyon ortamında üretilen veriler sistemin kimliklendirilmesinde kullanılmaktadır. Parametre değerlerinin belirlenmesi işlemi sistem parametrelerine sayısal değerlerin atanmasıyla yapılmaktadır. Sistemden toplanan referans verilerine göre eğri uydurma teknikleri kullanılmaktadır. Alınan sonuçlar tatmin edici bir seviyeye ulaşana kadar veri seti üzerinde farklı yöntem ve modeller denenmektedir [4-9,73].

Sistem kimliklendirme işleminde TA(Transient Analizi) gibi analitik yöntemler, NLR ve ANN gibi parametrik yöntemler kullanılmaktadır. Bunların yanı sıra tercih edilen diğer bir sistem kimliklendirme yöntemi ise sezgisel optimizasyon algoritmalarıdır. Literatürde gemi modeli, biyoproses modeli, helikopter motoru, su tankı gibi sistemlerin kimliklendirilmesi çalışmaları yer almaktadır [32-40].

Mühendislik uygulamalarını içeren birçok alan, doğrusal olmayan problem yapılarını barındırmaktadır. Doğrusal olmayan problemlerin çözülmesinde kullanılmak için geliştirilmiş birçok yöntem bulunmaktadır. Özellikle değişken sayısına ve veri tipine bağlı olarak problemlerin zorluk dereceleri artmaktadır. Bu problemlerin deterministik yöntemlerle çözülmesi, hem problemlerin yapısını hem de çözüm süreleri açısından dezavantaj içermektedir. İstenilen sonuca ulaşamaması veya kabul edilebilir sınırların dışındaki sürelerde istenilen çözüme ulaşılabilmesi bu dezavantajlardan bazılarıdır.

Bunların üstesinden gelebilmek için sezgisel optimizasyon yöntemlerine dayalı algoritmalar kullanılmaktadır. Özellikle popülasyon temelli sezgisel yöntemler ve çok noktalı arama yöntemleri sayesinde hızlı bir şekilde sonuç alınabilmektedir. Bu yöntemlere örnek olarak EYA, PSO ve ABA algoritmaları verilebilir.

Parametrelerin belirlenmesi işleminde, problemi minimize etmek için simülasyon ortamında optimizasyon algoritmaları kullanılmıştır. Bunlar;

- Parçacık Sürü Optimizasyon Algoritması
- Ateş Böceği Algoritması
- Emperyalist Yarışmacı Algoritma

Dördüncü bölüm iki kısımdan oluşmaktadır. İlk kısımda algoritmaların kimliklendirme performansları zaman değiştirilmeden veri seti büyüklüğünün kimliklendirme üzerindeki etkileri incelenmektedir. Bu kısımda yapılan çalışmalar test amaçlı olup algoritmaların kimliklendirme göstermektedir. Ardından, belirlenmiş sınırlar içerisindeki kütle(M) değerlerine karşılık ‘C’ ve ‘K’ parametreleri kimliklendirilmektedir

İkinci kısımda daha az veri ile ön tahmin işlemlerinin yapılacağı sistem cevabı parçasının bölge karşılaştırması yapılmaktadır. Sonrasında kimliklendirme işlemi tamamlanan sistem farklı kütle değerleriyle test edilmektedir. Kullanılan bu algoritmaların başarılarının değerlendirilmesi bu bölümde anlatılmaktadır. Bu çalışmanın her bölümünde yer çekim ivmesi olarak ‘ $g = 10 \text{ m/s}^2$ ’ alınmaktadır.

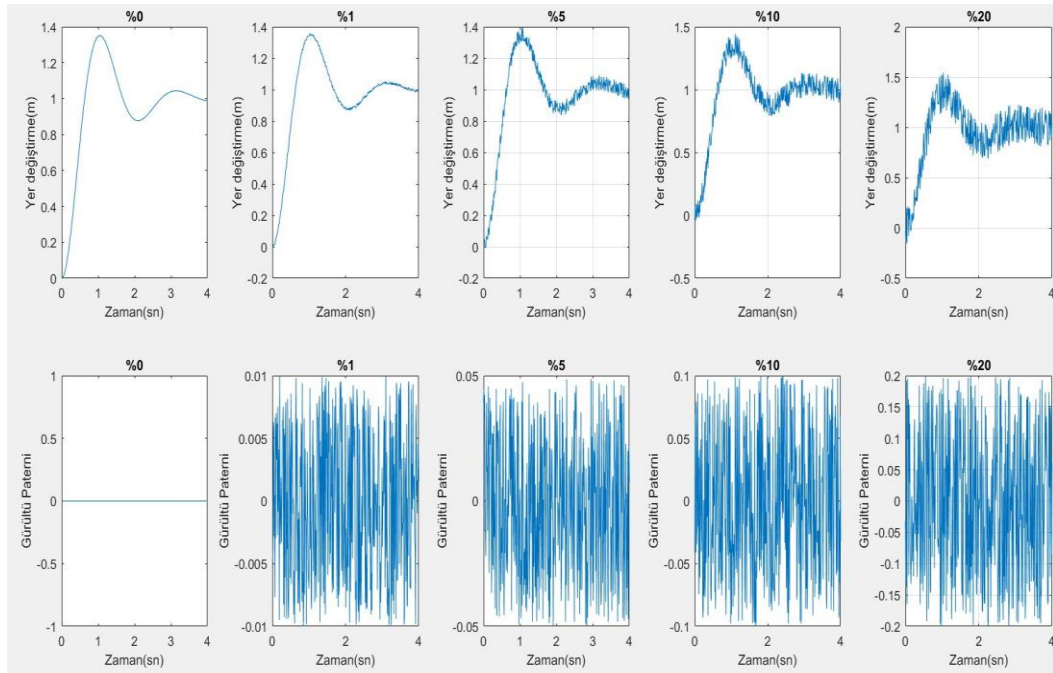
4.1. Verinin Hazırlanması

Parametrelerin bulunabilmesi için öncelikli olarak modele bağlı referans verilerine ihtiyaç duyulmaktadır. Bu veriler ölçüm yapılarak toplanabileceği gibi modelin bilinmesi durumunda modele bağlı olarak üretilebilir. Veri üretiminde Bölüm 2 de anlatılan az sönümlü model kullanılmaktadır. Üretim için kullanılan sistem parametreleri kütle(M) =1000 Kg, yay sabiti(K) = 10000 N/m sönümleme sabiti(C)= 2000 N/(m/s) olarak alınmaktadır [14].

Optimizasyon algoritmalarında parametre sınırları ortak olarak belirlenmektedir. Parametre sınırları kütle(M) değeri için 0 ile 10000, yay sabiti(K) değeri 8000 ile 12000 ve sönümlleme sabiti(C) 1600 ile 2400 arasında belirlenmektedir.

Uygulamalar yaşam koşullarında gerçekleştirildiğinde sistem cevaplarında genellikle içsel ve dışsal etkilere dayalı olarak gürültü oluşmaktadır. Ağırlık ölçme sistemleri çevresel gürültülere karşı oldukça hassas mekanizmalardır. Sistem cevapları bu etkiler altında incelenmektedir [11-13,16,44,51,52,57]. Yapılan çalışma kapsamında simülasyon ortamında üretilen veriler kullanılmaktadır. Üretilen sistem cevabına çeşitli oranlarda rastgele beyaz gürültü eklenerek optimizasyon algoritmalarının performansları incelenmektedir.

Önceden bilinen parametre değerleri modele verilerek simülasyon ortamında sistem cevapları elde edilmiş ve referans sistem cevabı olarak adlandırılmıştır. Sezgisel optimizasyon algoritmaları kullanılarak bulunan parametre değerlerinin modele verilmesiyle alınan veriler yeni sistem cevabı olarak adlandırılmıştır. Bu iki çıktı, her adımda kareler farkı toplamı denklemi ile karşılaştırılarak, maksimum adım sayısına kadar en iyi parametre değerleri bulunmaktadır. Denklem (4.1) ile gösterilmektedir.



Şekil 4.1 Hazırlanan veriler

$$\Delta y = \sum (y_{orijinal} - y_{tahmin})^2 \quad (4.1)$$

Az sönümlü modelden elde edilen referans sistem cevapları gürültü oranları ile birlikte Şekil 4.1 ile gösterilmektedir.

4.2. Algoritmaların Sistem Kimliklendirme Performanslarının Test Edilmesi

Bu başlık altında yapılan çalışmada birer adet ‘M’, ‘C’ ve ‘K’ değerleriyle algoritmaların kimliklendirme performansları ve algoritmalar üzerinde zaman aralığı değişmeden veri seti boyutunun değişmesinin kimliklendirme performansına etkisi test edilmektedir.

4.2.1. Parçacık Sürü Optimizasyon Algoritması ile Sistem Kimliklendirme

Parçacık Sürü Optimizasyonu ile %0, %1, %5, %10 ve %20 gürültülü sistem cevapları kullanılarak parametrelerin bulunması işlemi yapılmaktadır. Parçacık Sürü Optimizasyonu Algoritması başlangıç şartları olarak sürü büyüklüğü = 20, adım sayısı = 100 ayarlanmıştır. Sistem cevapları 0 ile 4 saniye arasında 0.01 adım büyüklüğü kullanılarak gösterilmektedir.

4.2.1.1. Parçacık Sürü Optimizasyon Algoritma Değişkenleri

Parçacık Sürü Optimizasyonu içerisinde kullanılan algoritma değişkenleri Bölüm 3 ile verilmektedir. Algoritmada kullanılan başlangıç şartları:

- Popülasyon büyüklüğü, $n_{pop} = 15$,
- Maksimum adım sayısı, $it_{max} = 100$,
- Üç bilinmeyen için parametre boyutu, $d = 3$,
- Parçacıkların arama ağırlığını belirleyen değişken, $w = 0.6$,
- ‘w’ parametresinin her adımda ayarlanmasını sağlayan katsayı, $w_{damp} = 0.7$,
- Hızlanma veya öğrenme katsayıları olarak ifade edilen, $c_1 = 3$, $c_2 = 1$ olarak belirlenmiştir.

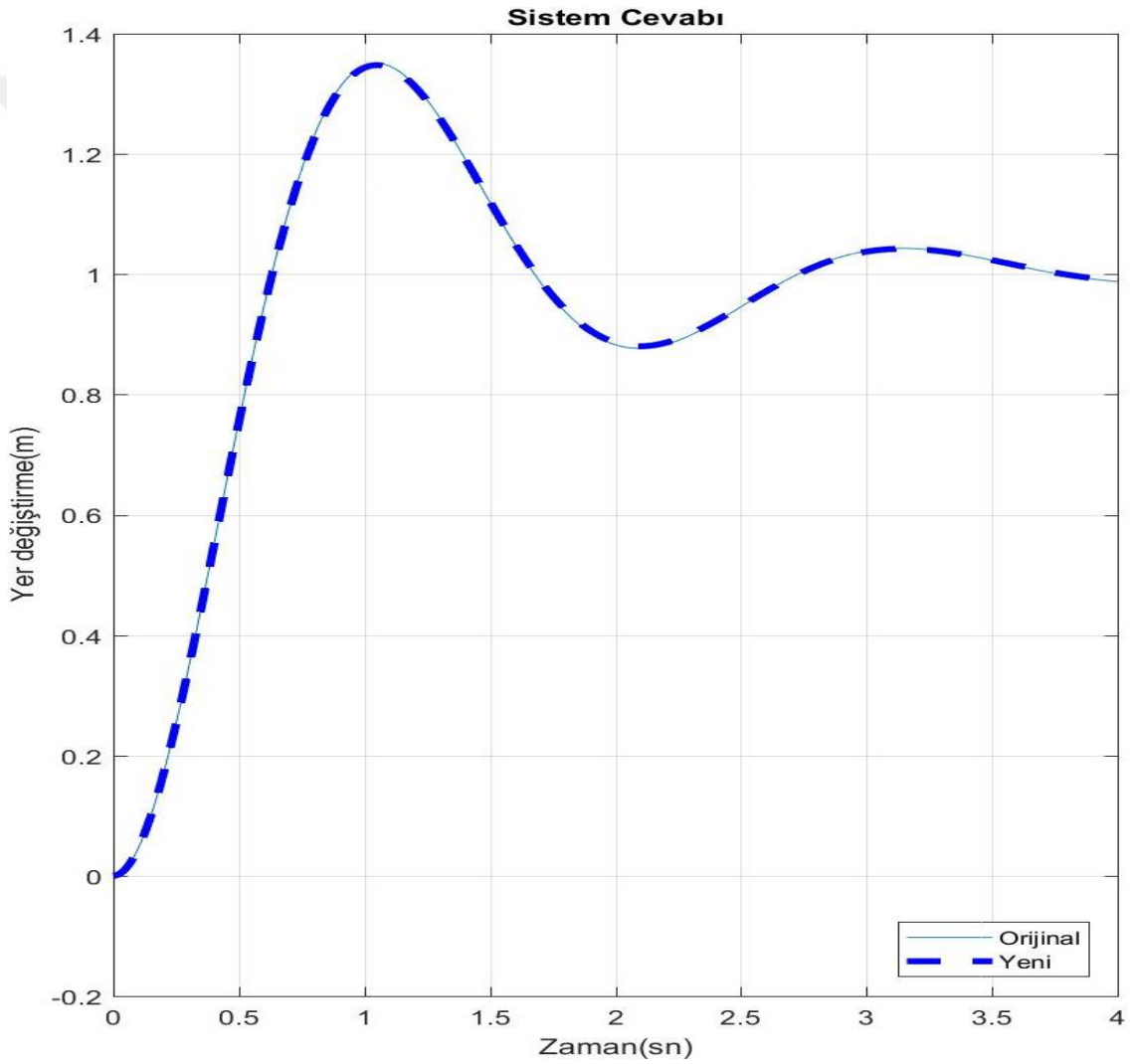
4.2.1.2. Gürültüsüz Sistem Cevabı Sonuçları

Şekil 4.1 ile gösterilen görsel üzerinde soldan birinci sırada yer alan gürültü eklenmemiş sistem cevabı referans alınarak ve Bölüm 4.2.1.1 de verilen Parçacık sürü Optimizasyonu algoritma başlangıç koşulları kullanarak parametrelerin bulunması

işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.1 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.2 ile gösterilmektedir.

Tablo 4.1 PSO ile gürültüsüz referans kullanılarak bulunan parametreler.

	Orijinal	Bulunan
Kütle(M)	1000	1005,725
Yay Sabiti(K)	10000	10050,728
Sönüm Sabiti(C)	2000	2031,107
SSE	0.0016548	
Süre(sn)	1.4451	



Şekil 4.2 Gürültü eklenmiş referans, bulunan parametrelerle alınan sistem cevabı

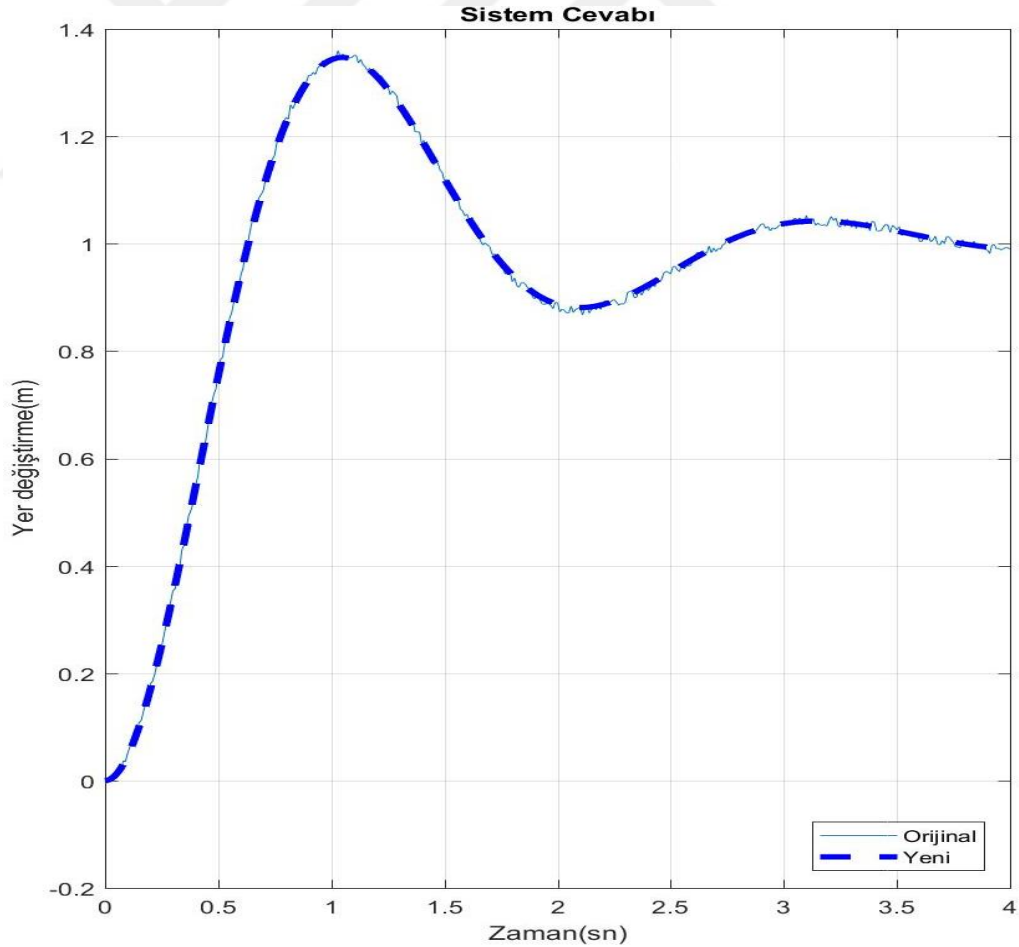
4.2.1.3. Gürültü Eklenmiş Sistem Cevabı Üzerinden Sonuçlar

- %1 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan ikinci sırada yer alan %1 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.1.1 verilen Parçacık sürü Optimizasyonu algoritmasının başlangıç koşulları kullanarak parametre değerlerinin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.2 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.3 ile gösterilmektedir.

Tablo 4.2 %1 gürültü referans cevabına göre PSO ile bulunan parametreler

	Orijinal	Gürültüsüz Bulunan
Kütle(M)	1000	1006,193
Yay Sabiti(K)	10000	10053,808
Sönüm Sabiti(C)	2000	2036,230
SSE	0.0024352	
Süre(Sn)	1.4302	



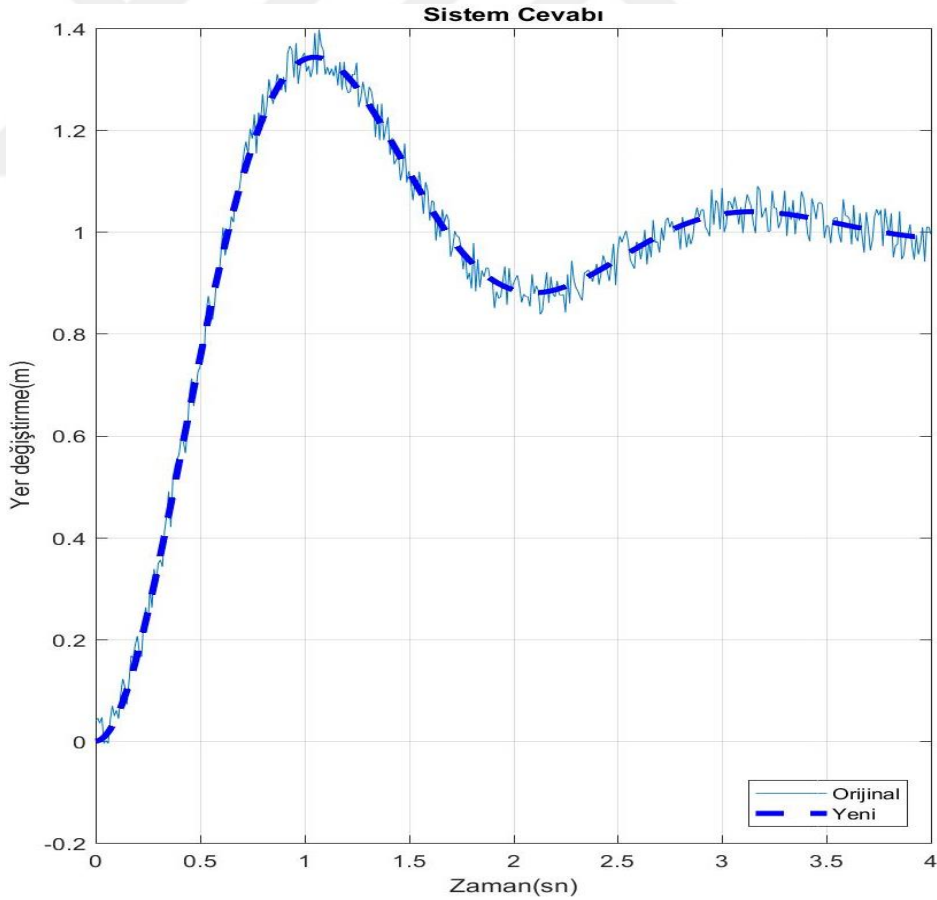
Şekil 4.3 %1 Gürültü eklenmiş referans, bulunan parametrelerle alınan sistem cevabı

- %5 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan üçüncü sırada yer alan %5 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.1.1 verilen Parçacık Sürü Optimizasyonu algoritmasının başlangıç koşulları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.3 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.4 ile gösterilmektedir.

Tablo 4.3 %5 gürültü referansa göre PSO ile belirlenen parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	998,601
Yay Sabiti(K)	10000	9997,455
Sönüm Sabiti(C)	2000	2029,893
SSE	0.005513	
Süre(sn)	1.4614	



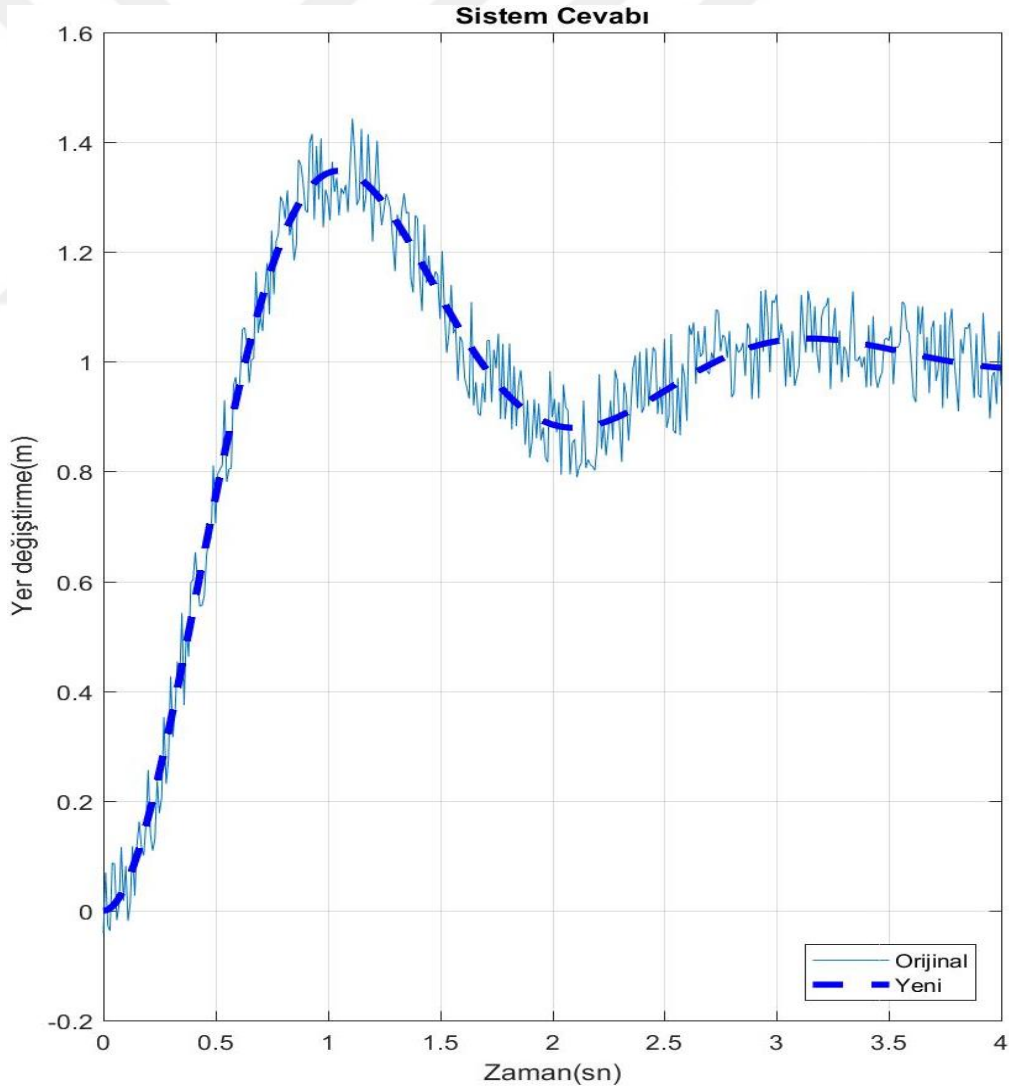
Şekil 4.4 %5 gürültü eklenmiş referans, bulunan parametrelerle alınan sistem cevabı

- %10 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan dördüncü sırada yer alan %10 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.1.1 verilen Parçacık sürü Optimizasyonu algoritma başlangıç koşulları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.4 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.5 ile gösterilmektedir.

Tablo 4.4 %10 gürültü referansa göre PSO ile bulunan parametreler

	Orijinal	Belirlenen Parametreler
Kütle(M)	1000	1000,108
Yay Sabiti(K)	10000	9997,359
Sönüm Sabiti(C)	2000	2017,544
SSE		0.001263
Süre(sn)		1.4726



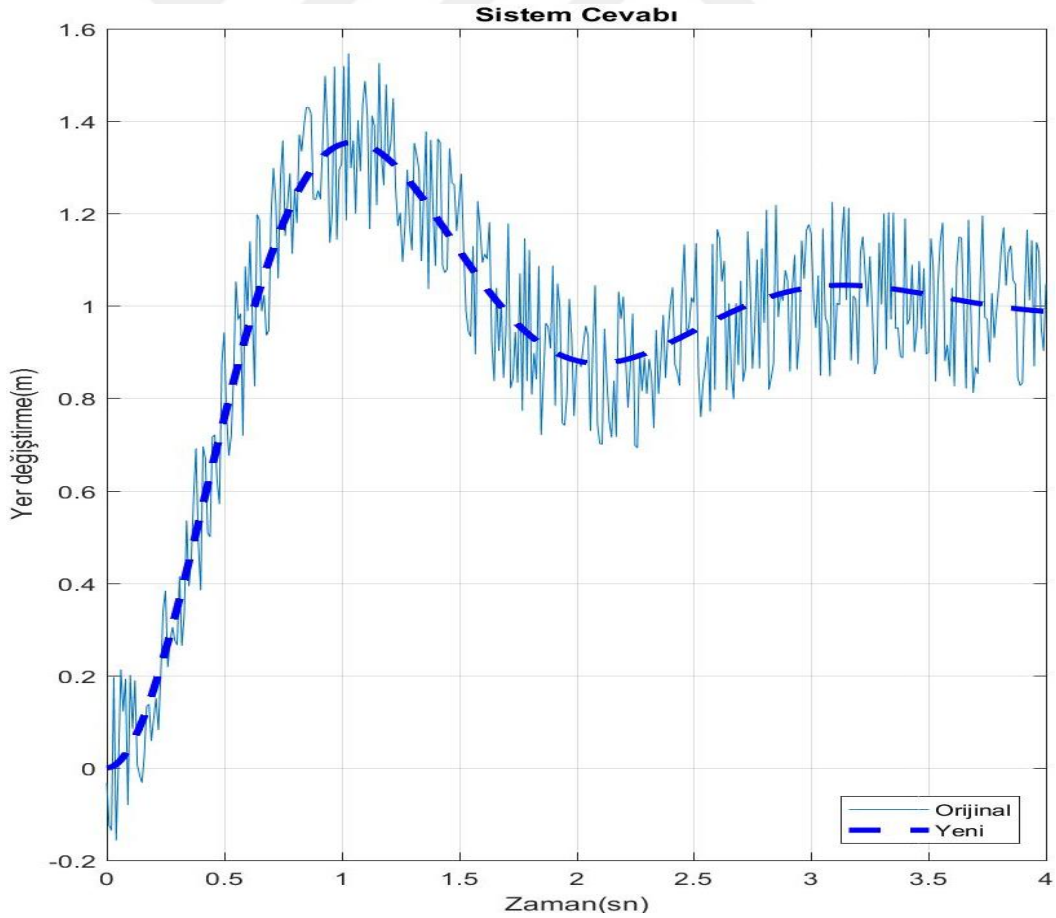
Şekil 4.5 %10 gürültü eklenmiş referans, bulunan parametrelerle alınan sistem cevabı

- %20 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan beşinci sırada yer alan %20 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.1.1 verilen Parçacık sürü Optimizasyonu algoritma başlangıç koşulları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.5 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.6 ile gösterilmektedir.

Tablo 4.5 %20 gürültü referansa göre belirlenen parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	999,906
Yay Sabiti(K)	10000	9992,158
Sönüm Sabiti(C)	2000	1988,170
SSE	0.0008843	
Süre(sn)	1.4854	



Şekil 4.6 %20 gürültülü referans, bulunan parametrelerle alınan sistem cevabı

4.2.2. Ateş Böceği Algoritması ile Sistem Kimliklendirme

Ateş Böceği Algoritması için gürültüsüz, %1, %5, %10 ve %20 gürültü eklenmiş sistem cevapları kullanılarak tahmin yapılmaktadır. Sistem cevabı, 0 ile 4 saniye arasında 0.01 adım büyüklüğü kullanılarak gösterilmektedir.

4.2.2.1. Ateş Böceği Algoritmasının Parametreleri

Parçacık sürü optimizasyonu içerisinde kullanılan algoritmanın başlangıç şartlarını oluşturan değişkenleri Bölüm 3 ile verilmektedir. Belirlenen başlangıç şartları:

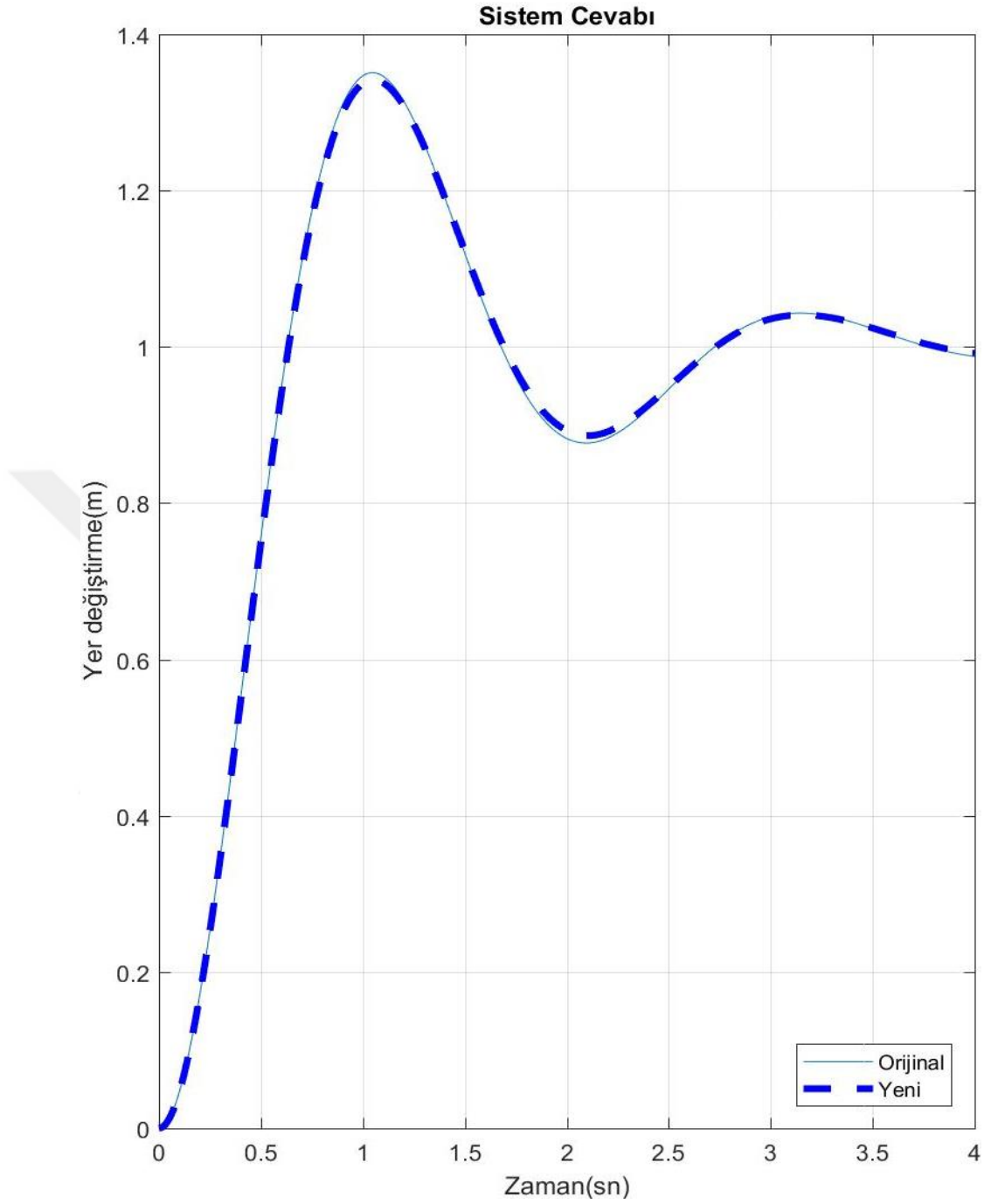
- Popülasyon sayısı, $n_{pop} = 15$,
- Maksimum adım sayısı $it_{max} = 100$,
- Parametre boyutu, $d = 3$,
- Işık emilim katsayısı $\gamma = 0.98$,
- Çekicilik katsayısı temel değişkeni, $\beta_0 = 0.6$,
- Mutasyon katsayısı, $\alpha = 0.8$,
- Mutasyon katsayısının her adımdaki değişme oranı, $\alpha_{damp} = 0.7$ olarak belirlenmiştir.

4.2.2.2. Gürültüsüz Çıktı Sonuçları

Şekil 4.1 ile gösterilen görsel üzerinde, soldan birinci sırada yer alan gürültü eklenmemiş sistem cevabı referans alınarak ve Bölüm 4.2.2.1 ile verilen Ateş Böceği algoritmasının başlangıç şartları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.6 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.7 ile gösterilmektedir.

Tablo 4.6 Gürültü eklenmemiş referansa göre ABA ile belirlenen parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	1006,239
Yay Sabiti(K)	10000	10045,222
Sönüm Sabiti(C)	2000	2068,560
SSE	0.013077	
Süre(sn)	24.5625	



Şekil 4.7 Gürültü eklenmemiş referans, bulunan parametreler ile alınan sistem cevabı

4.2.2.3. Gürültü Eklenmiş Sistem Cevabı Üzerinden Sonuçlar

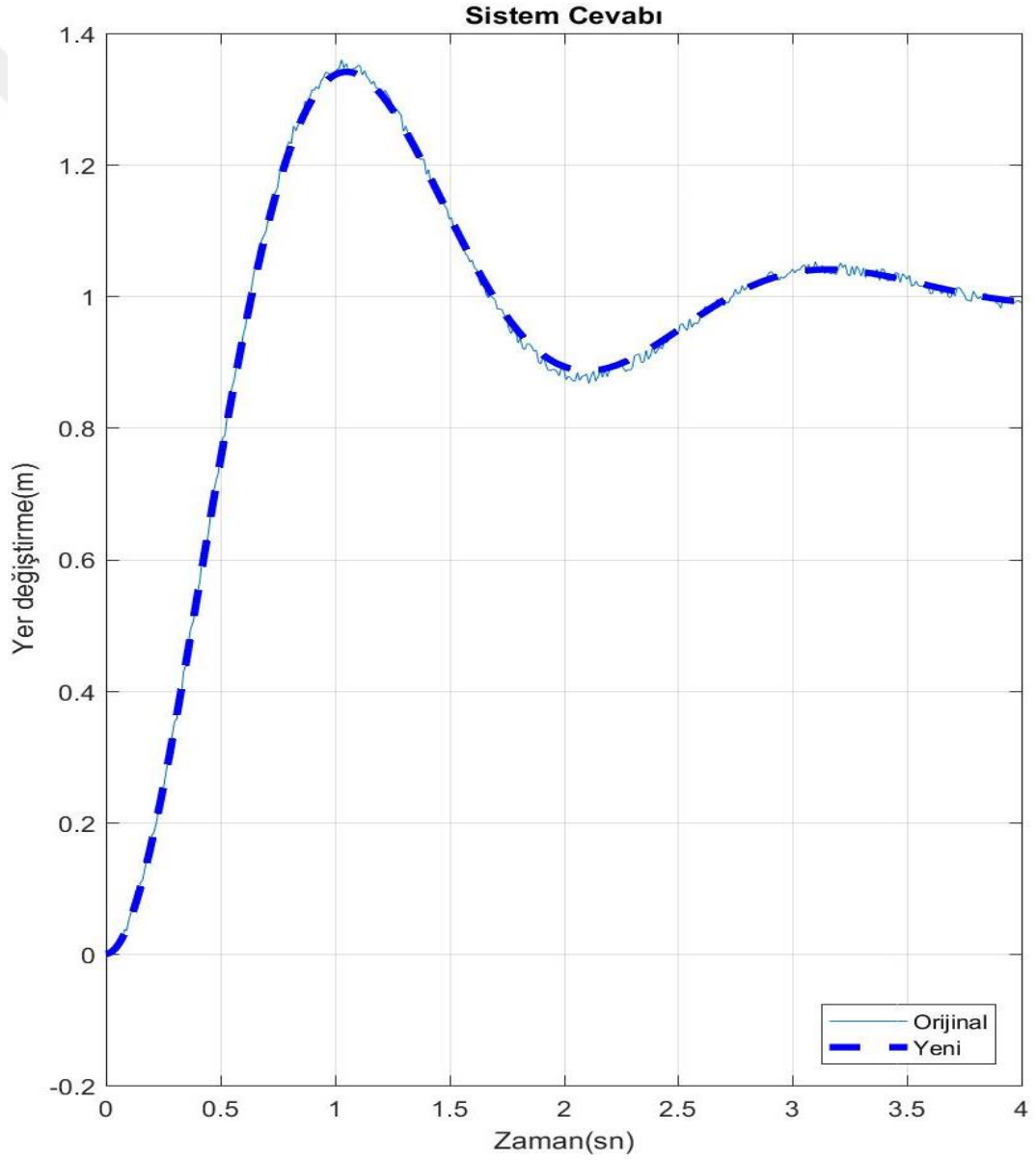
- %1 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan ikinci sırada yer alan %1 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.2.1 ile verilen Ateş Böceği algoritmasının başlangıç şartları kullanılarak parametrelerin bulunması işlemi

gerçekleştirilmektedir. Bulunan parametreler Tablo 4.7 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.8 ile gösterilmektedir.

Tablo 4.7 %1 gürültü eklenmiş referansa göre ABA ile bulunan parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	1006,668
Yay Sabiti(K)	10000	10049,444
Sönüm Sabiti(C)	2000	2071,391
SSE	0.013945	
Süre(sn)	24.6875	



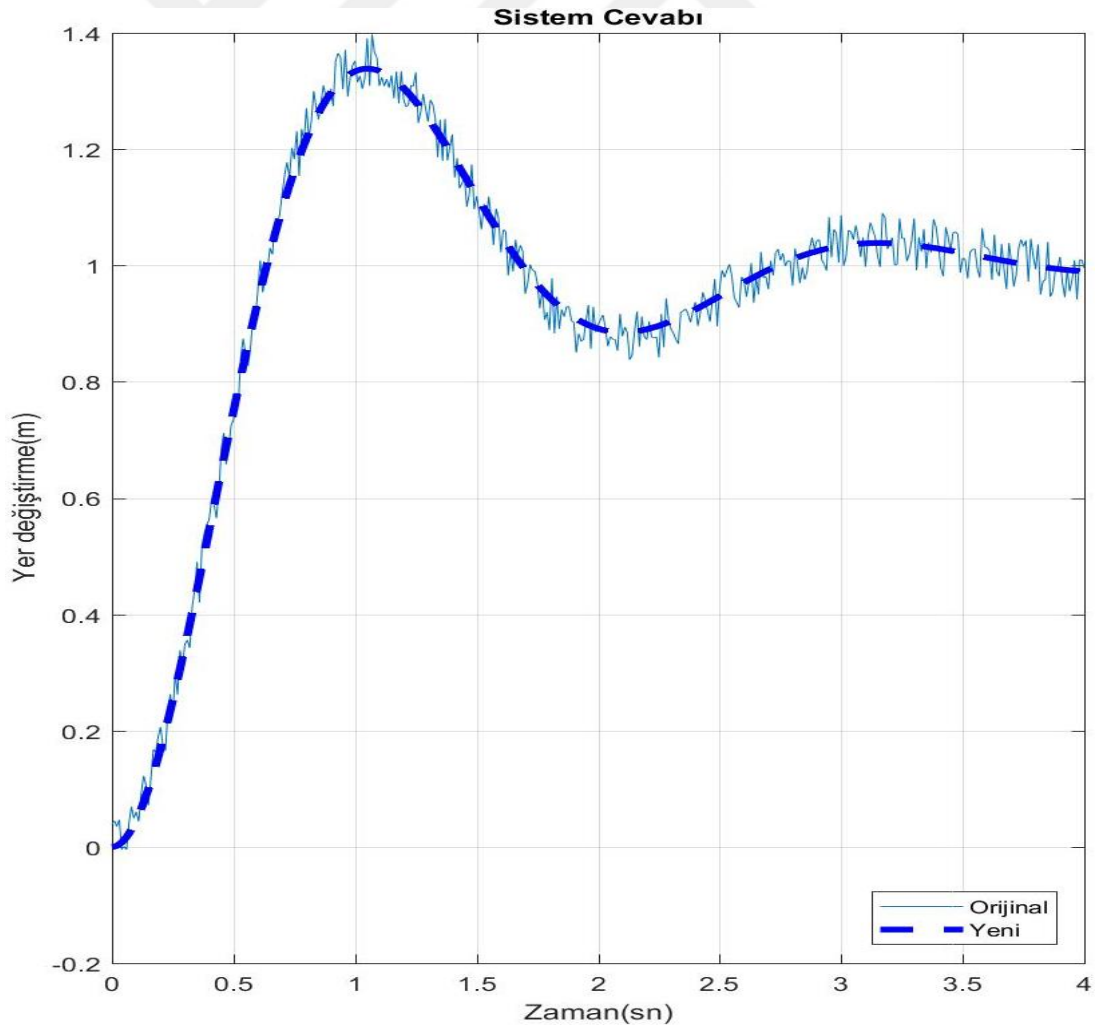
Şekil 4.8 %1 gürültü eklenmiş referans, bulunan parametreler ile alınan sistem cevabı

- %5 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan üçüncü sırada yer alan %5 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.2.1 ile verilen Ateş Böceği algoritmasının başlangıç şartları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.8 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.9 ile gösterilmektedir.

Tablo 4.8 %5 gürültü eklenmiş referansa göre ABA ile bulunan parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	998,404
Yay Sabiti(K)	10000	9986,325
Sönüm Sabiti(C)	2000	2057,583
SSE	0.015599	
Süre(sn)	24.5469	



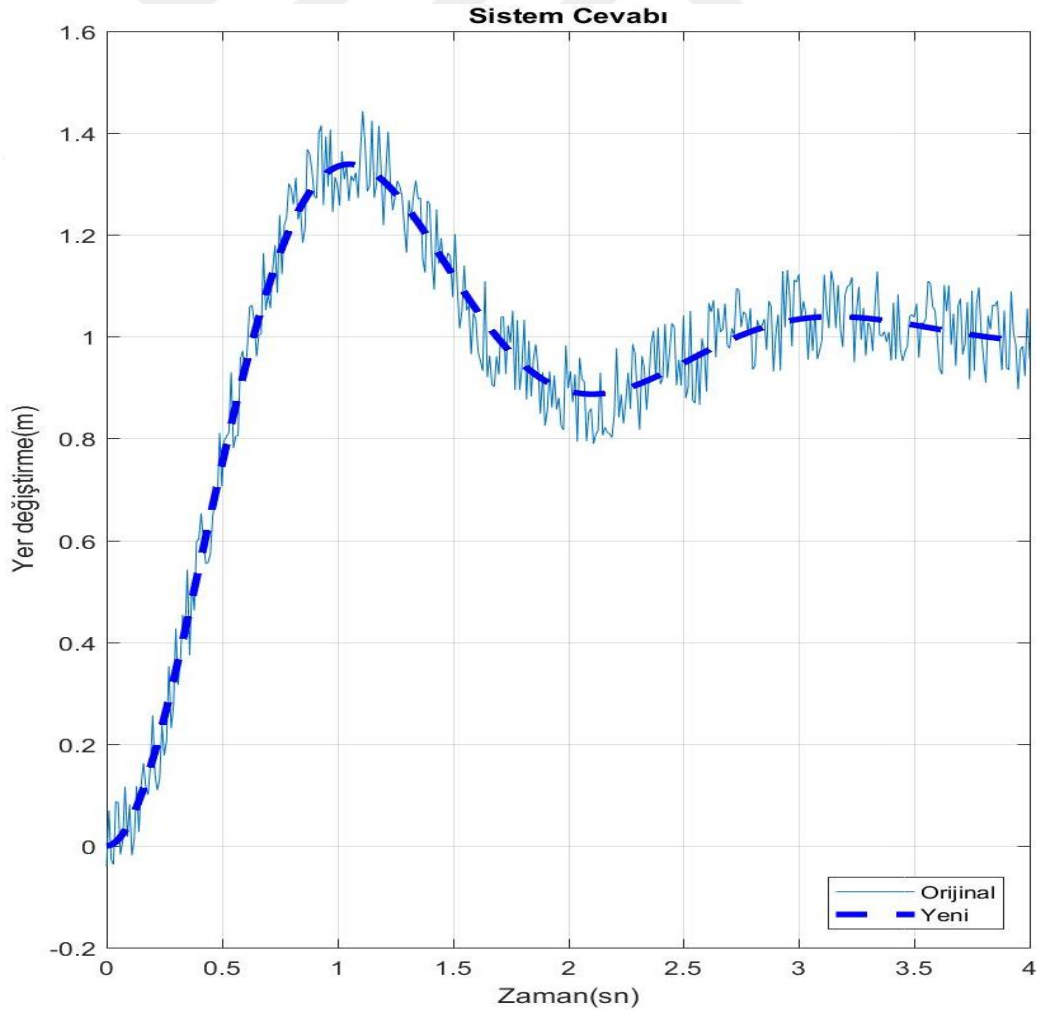
Şekil 4.9 %5 gürültülü referans, bulunan parametreler ile alınan sistem cevabı

- %10 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan dördüncü sırada yer alan %10 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.2.1 ile verilen Ateş Böceği algoritmasının başlangıç şartları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.9 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.10 ile gösterilmektedir.

Tablo 4.9 %10 gürültü eklenmiş referansa göre ABA ile bulunan parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	1003,770
Yay Sabiti(K)	10000	10027,260
Sönüm Sabiti(C)	2000	2073,999
SSE		0.017842
Süre(sn)		24.2344



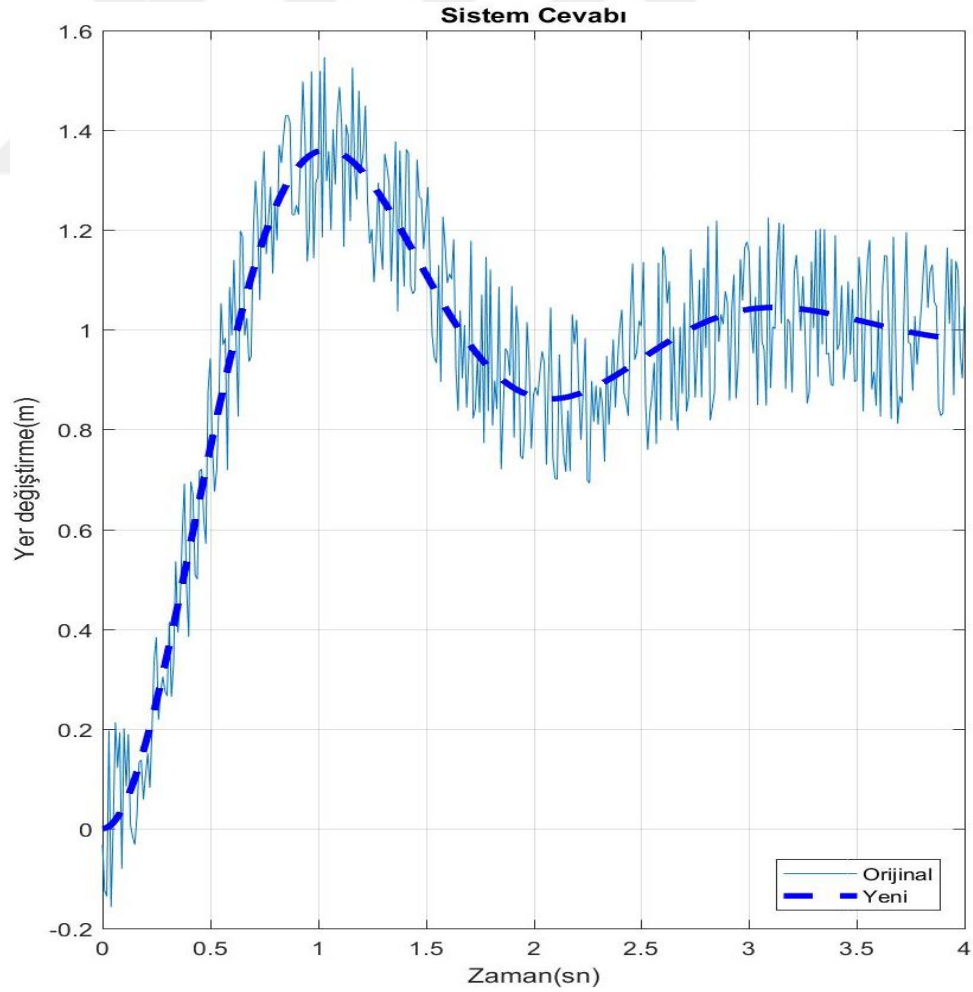
Şekil 4.10 %10 gürültülü referans, bulunan parametreler ile alınan sistem cevabı

- %20 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan beşinci sırada yer alan %20 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.2.1 ile verilen Ateş Böceği algoritmasının başlangıç şartları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.10 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.11 ile gösterilmektedir.

Tablo 4.10 %20 gürültü eklenmiş referansa göre ABA ile bulunan parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	989,925
Yay Sabiti(K)	10000	9942,216
Sönüm Sabiti(C)	2000	1907,048
SSE	0.028376	
Süre(sn)	21.6719	



Şekil 4.11 %20 gürültülü referans, bulunan parametreler ile alınan sistem cevabı

4.2.3. Emperyalist Yarışmacı Algoritması ile Sistem Kimliklendirme

Emperyalist Yarışmacı Algoritma için gürültüsüz, %1, %5, %10 ve %20 gürültü eklenmiş sistem cevapları kullanılarak, sistem parametrelerinin bulunması işlemi yapılmaktadır. EYA'nın başlangıç şartları olarak popülasyon büyüklüğü=40, emperyalist devlet sayısı= 8, maksimum adım sayısı=100 belirlenmiştir. Sistem cevabı, 0 ile 4 saniye arasında 0.01 adım büyüklüğü kullanılarak gösterilmektedir.

4.2.3.1. Emperyalist Yarışmacı Algoritmasının Değişkenleri

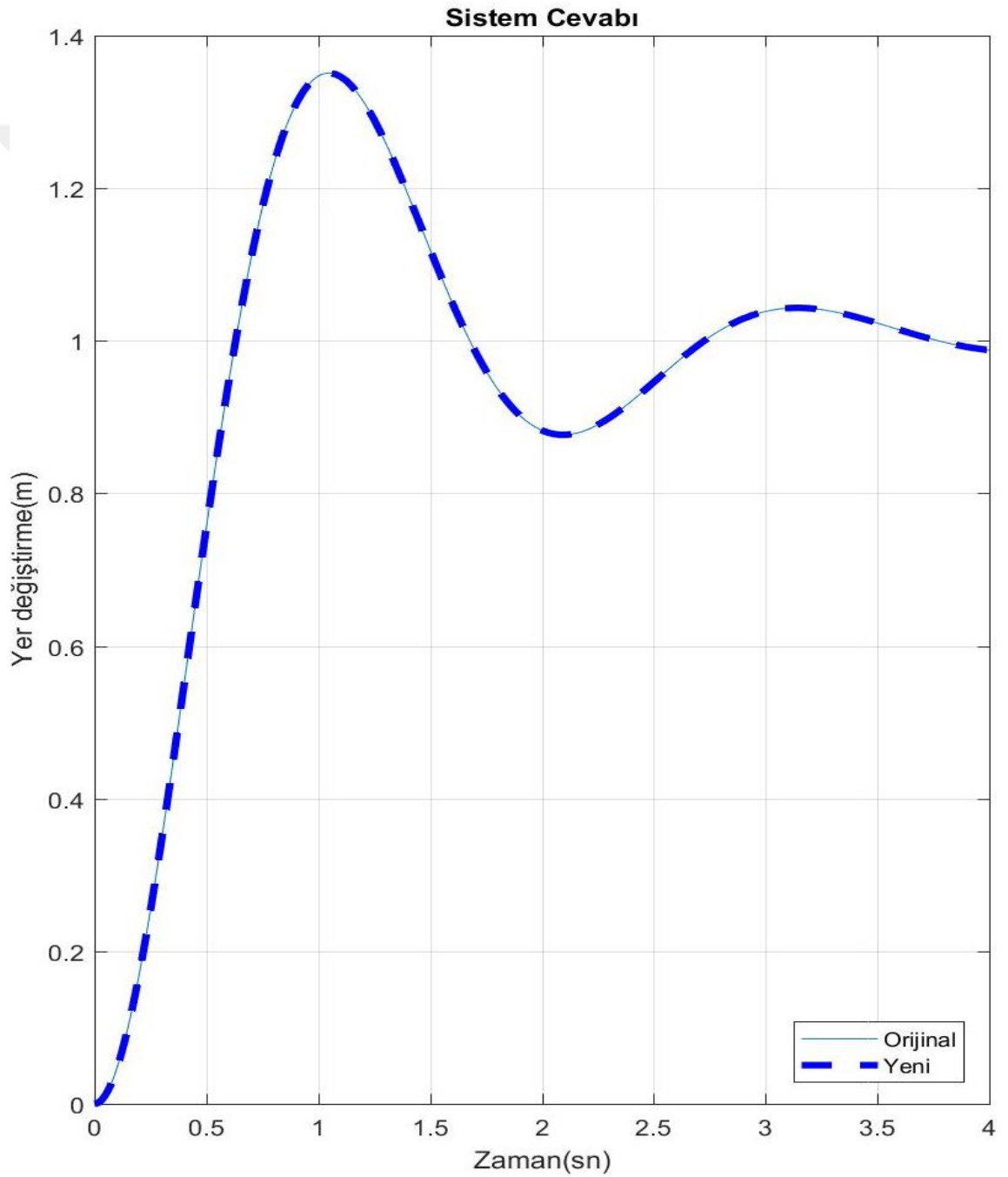
- Popülasyon büyüklüğü, $n_{pop} = 20$,
- Emperyalist sayısı, $n_{emp} = 8$,
- Maksimum adım sayısı, $it_{max} = 100$,
- Parametre boyutu, $d = 3$,
- Seçim baskısı, $\alpha = 0.5$,
- Asimilasyon katsayısı, $\beta = 1.25$,
- Devrim olma olasılığı, $p_{rev} = 0.4$,
- Devrim oranı, $\mu = 0.15$,
- Ortalama koloni uygunluk değeri hesaplama katsayısı, $\varepsilon = 0.5$ olarak belirlenmiştir.

4.2.3.2. Gürültüsüz Sistem Cevabı Sonuçları

Şekil 4.1 ile gösterilen görsel üzerinde, soldan birinci sırada yer alan gürültü eklenmemiş sistem cevabı referans alınarak ve Bölüm 4.2.3.1 verilen Emperyalist Yarışmacı Algoritmanın başlangıç koşulları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.11. ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.12 ile gösterilmektedir.

Tablo 4.11 Gürültü eklenmemiş referansa göre EYA ile bulunan parametreler

	Orijinal	Bulunan
Kütle(M)	1000	1001,191
Yay Sabiti(K)	10000	10011,913
Sönüm Sabiti(C)	2000	2002,391
SSE	3.5387e-10	
Süre(sn)	2.1604	



Şekil 4.12 Gürültüsüz referansa, bulunan parametreler ile alınan sistem cevabı

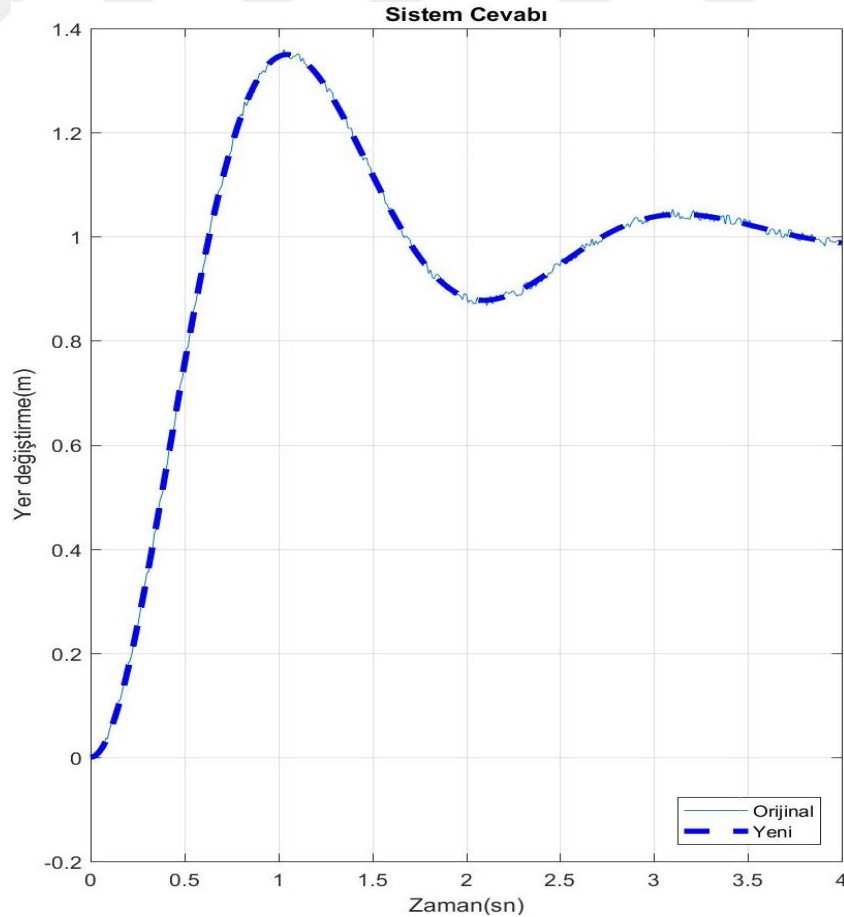
4.2.3.3. Gürültü Eklenmiş Sistem Cevapları Üzerinden Sonuçlar

- %1 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan ikinci sırada yer alan %1 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.3.1 ile verilen Emperyalist Yarışmacı Algoritmanın başlangıç şartları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.12 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.13 ile gösterilmektedir.

Tablo 4.12 %1 gürültü eklenmiş referansa göre EYA ile bulunan parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	1001,047
Yay Sabiti(K)	10000	10009,969
Sönüm Sabiti(C)	2000	2007,739
SSE	0.00013424	
Süre(sn)	1.9689	



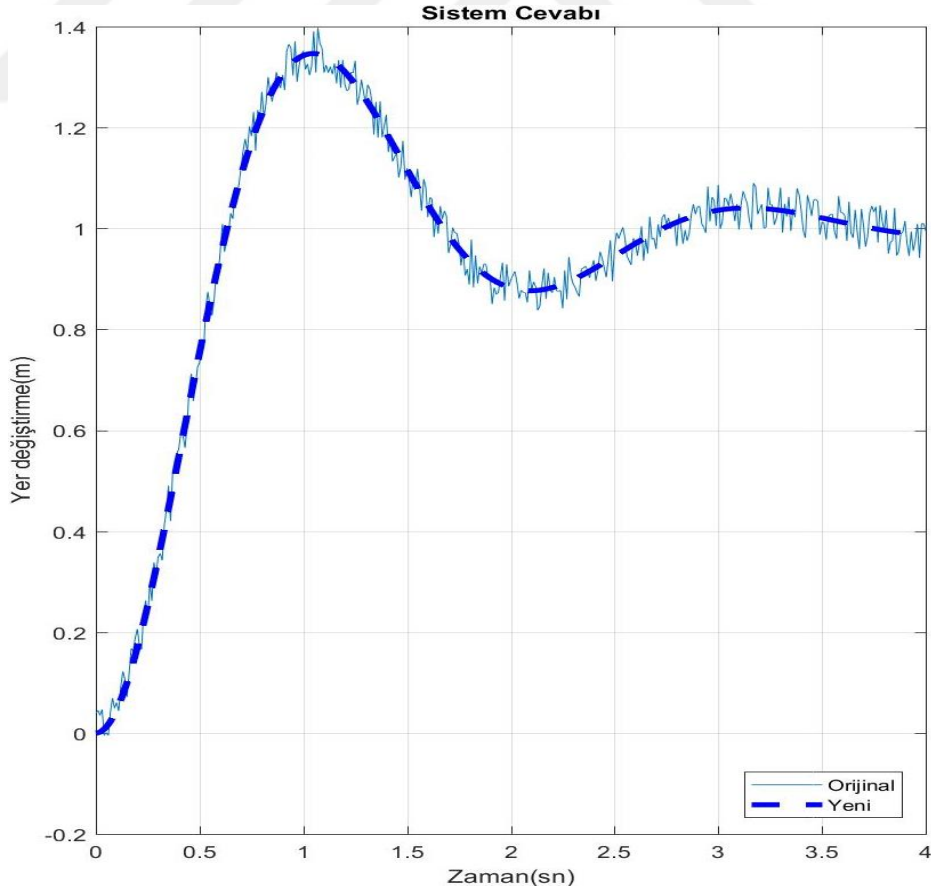
Şekil 4.13 %1 gürültülü referans, bulunan parametreler ile alınan sistem cevabı

- %5 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan üçüncü sırada yer alan %5 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.3.1 ile verilen Emperyalist Yarışmacı Algoritmanın başlangıç şartları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.13 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.14 ile gösterilmektedir.

Tablo 4.13 %5 gürültü eklenmiş referans çıktısına göre EYA ile bulunan parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	1002,416
Yay Sabiti(K)	10000	10042,649
Sönüm Sabiti(C)	2000	2015,651
SSE	0.0021482	
Süre(sn)	1.9737	



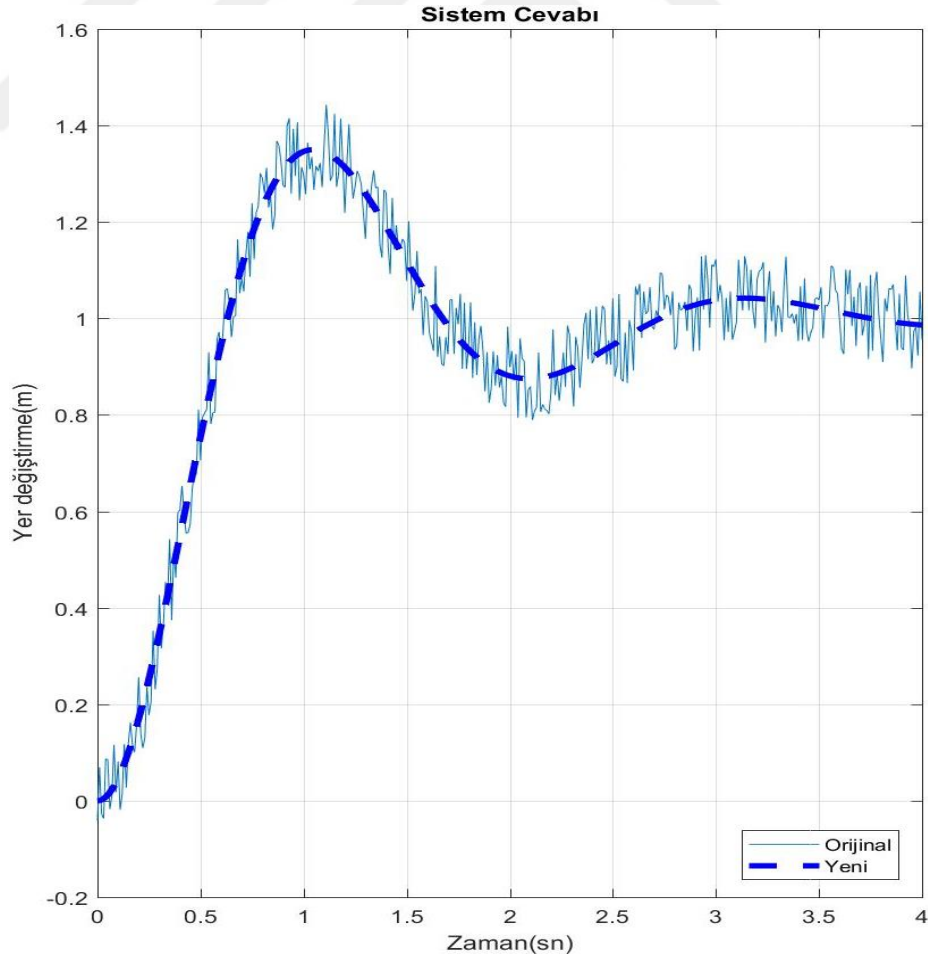
Şekil 4.14 %5 gürültü eklenmiş referans, bulunan parametreler ile alınan sistem cevabı

- %10 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan dördüncü sırada yer alan %10 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.3.1 ile verilen Emperyalist Yarışmacı Algoritmanın başlangıç şartları kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.14 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.15 ile gösterilmektedir.

Tablo 4.14 %10 gürültü eklenmiş referansa göre EYA ile bulunan parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	1002,459
Yay Sabiti(K)	10000	10037,028
Sönüm Sabiti(C)	2000	2002,552
SSE	0.00063252	
Süre(sn)	2.0331	



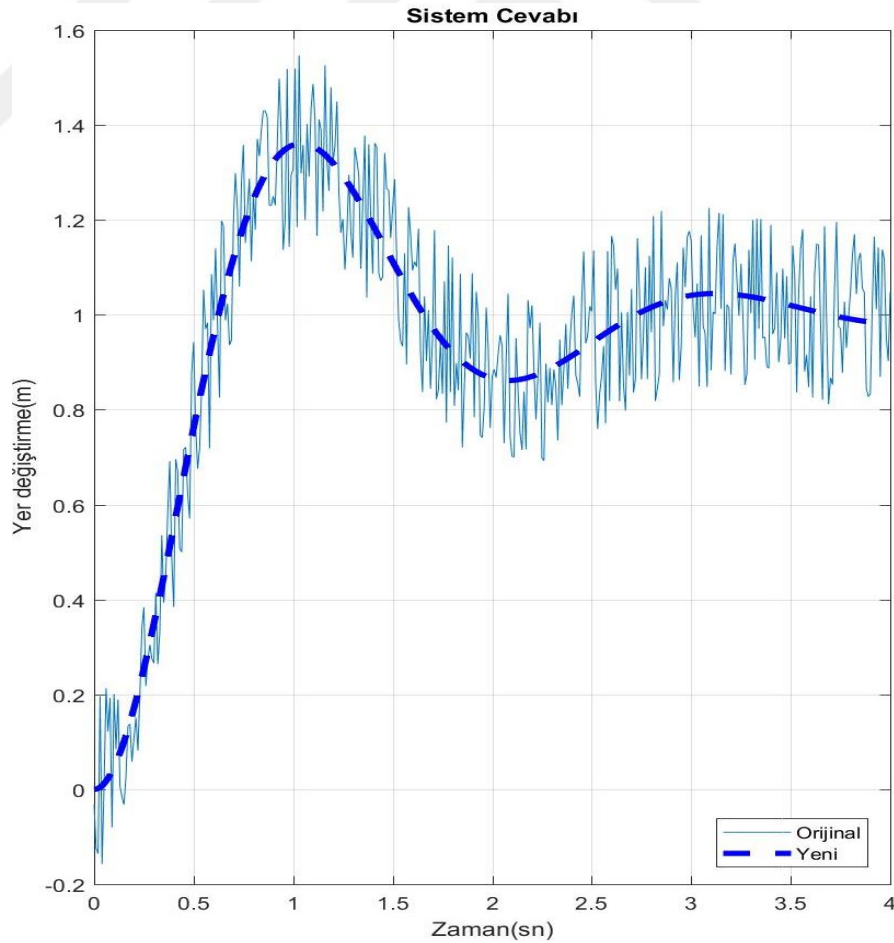
Şekil 4.15 %10 gürültülü referans, bulunan parametreler ile alınan sistem cevabı

- %20 Gürültü Eklenen Sistem Cevabı

Şekil 4.1 ile gösterilen görsel üzerinde, soldan beşinci sırada yer alan %20 oranında gürültü eklenmiş sistem cevabı referans alınarak ve Bölüm 4.2.3.1 ile verilen Emperyalist Yarışmacı Algoritmanın başlangıç şartları ve orijinal sistem cevabı kullanılarak parametrelerin bulunması işlemi gerçekleştirilmektedir. Bulunan parametreler Tablo 4.15 ile verilmektedir. Bu parametrelere ait az sönümlü sistem cevabı Şekil 4.16 ile gösterilmektedir.

Tablo 4.15 %20 gürültü eklenmiş referansa göre EYA ile bulunan parametreler

	Orijinal	Bulunan Parametreler
Kütle(M)	1000	1000,239
Yay Sabiti(K)	10000	10045,718
Sönüm Sabiti(C)	2000	1927,0863
SSE	0.028248	
Süre(sn)	2.0825	



Şekil 4.16 %20 gürültülü referans, bulunan parametreler ile alınan sistem cevabı

4.2.4. Farklı Veri Büyüklüklerinde Kimliklendirme

Yapılan çalışmada 400 veri içeren bir vektör kullanılmaktadır. Farklı veri büyüklüklerine sahip referans sistem cevabının çalışma üzerinde ki etkileri incelenecektir. Referans cevabın veri büyüklükleri 100, 500, 1000 ve 2000 olarak alınmıştır.

Veri büyüklüğü 100 olarak alındığında gürültüsüz,%1, %5, %10, %20 gürültülü referans değerlerinde %0, %1, %5 ve %10 gürültü oranlarında EYA algoritmasının diğer algoritmalarla kıyasla SSE değerlerinin daha iyi çıktığı Tablo 4.16-4.18 arasına gösterilmektedir. Zaman performansı açısından kıyaslandığında en iyi performanstan kötüye doğru PSO, EYA ve ABA olarak sıralanmaktadır

Tablo 4.16 N=100 adet veri için PSO sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1005,725	998,065	998,877	1002,170	1019,852
Yay Sbt(K)	10000	10050,746	9980,095	9985,944	10051,425	10186,038
Sönüm Sbt(C)	2000	2031,242	2015,246	2033,056	2066,830	1989,666
SSE		0.00041961	0.00038823	0.0012984	0.0054762	0.0030107
Süre(sn)		1.0732	0.96904	0.96034	0.95951	0.97197

Tablo 4.17 N=100 adet veri için ABA sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1006,609	1003,624	998,660	988,769	995,010
Yay Sbt(K)	10000	10048,555	10022,841	9987,554	9918,328	9938,842
Sönüm Sbt(C)	2000	2069,133	2068,756	2048,070	2034,500	1937,464
SSE		0.0032612	0.0038604	0.0027178	0.004961	0.0034187
Süre(sn)		6.9288	6.9272	6.8859	5.804	4.9372

Tablo 4.18 N=100 adet veri için EYA sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	999,893	1002,099	1001,658	998,925	1011,441
Yay Sbt(K)	10000	9998,874	10027,683	10025,182	10020,228	10102,958
Sönüm Sbt(C)	2000	1999,864	2001,589	2024,935	2055,322	1969,454
SSE		8.6328e-9	4.8994e-5	0.00062657	0.0049528	0.0034196
Süre(sn)		1.7181	1.6336	1.6765	1.7182	1.785

Referans sistem cevabının veri büyüklüğü 500 olarak kullanıldığında gürültüsüz, %1, %5, %10 ve %20 oranında gürültü eklenmiş referans değerleri ile çalışma yapılmıştır.

%0, %1, %5 ve %10 gürültü oranlarında EYA algoritmasının diğer algoritmalarla kıyasla SSE değerlerinin daha iyi çıktığı Tablo 4.19-4.21 arasına gösterilmektedir. Zaman performansı açısından kıyaslandığında en iyi performanstan kötüye doğru PSO, EYA ve ABA olarak sıralanmaktadır.

Tablo 4.19 N=500 adet veri için PSO sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1005,786	1006,1842	998,254	1005,586	996,588
Yay Sbt(K)	10000	10052,040	10052,513	9971,875	10045,522	9915,944
Sönüm Sbt(C)	2000	2029,966	2029,095	2011,162	2036,438	1951,708
SSE		0.0018022	0.0016971	0.0015175	0.0035252	0.026084
Süre(sn)		0.99326	0.96388	0.97559	0.97204	0.96192

Tablo 4.20 N=500 adet veri için ABA sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1006,243	1007,093	1007,638	1002,766	1012,825
Yay Sbt(K)	10000	10045,174	10049,151	10051,203	10012,367	10049,202
Sönüm Sbt(C)	2000	2068,556	2078,951	2079,179	2064,822	2071,305
SSE		0.016346	0.021897	0.021854	0.018109	0.037621
Süre(sn)		6.9388	7.2153	7.3514	5.9286	5.004

Tablo 4.21 N=500 adet veri için EYA sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1001,191	1001,286	1001,020	1002,313	1010,885
Yay Sbt(K)	10000	10011,913	10011,309	10005,248	10026,129	10058,529
Sönüm Sbt(C)	2000	2002,391	2003,150	2002,732	2002,002	1979,257
SSE		4.4233e-10	1.304e-05	0.00012608	7.2213e-05	0.026131
Süre(sn)		1.5296	1.4842	1.5442	1.5963	1.4419

Referans sistem cevabının veri büyüklüğü 1000 olarak kullanıldığında gürültüsüz, %1, %5, %10 ve %20 gürültüye sahip referans değerleri ile çalışma yapılmıştır.

EYA algoritmasının diğer algoritmalarla kıyasla SSE değerlerinin daha iyi çıktığı Tablo 4.22-4.24 arasına gösterilmektedir. Zaman performansı açısından kıyaslandığında en iyi performanstan kötüye doğru PSO, EYA ve ABA olarak sıralanmaktadır.

Tablo 4.22 N=1000 adet veri için PSO sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1005,795	1007,536	998,311	1004,957	1005,417
Yay Sbt(K)	10000	10052,045	10063,653	9976,546	10013,781	10044,744
Sönüm Sbt(C)	2000	2029,967	2037,990	2011,026	2016,591	2035,990
SSE		0.003603	0.0061355	0.0023967	0.013113	0.0068764
Süre(sn)		1.9923	1.9038	1.9486	1.8905	1.9371

Tablo 4.23 N=1000 adet veri için ABA sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1006,233	1007,169	1006,165	1008,911	1003,566
Yay Sbt(K)	10000	10045,220	10049,368	10036,044	10039,170	10022,356
Sönüm Sbt(C)	2000	2068,541	2076,698	2078,219	2068,250	2063,248
SSE		0.032662	0.040998	0.046419	0.0446	0.032297
Süre(sn)		13.6031	13.3409	13.4809	13.4162	13.4489

Tablo 4.24 N=1000 adet veri için EYA sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1001,191	1000,293	1000,319	1009,473	1003,406
Yay Sbt(K)	10000	10011,913	10001,434	9998,860	10065,718	10040,403
Sönüm Sbt(C)	2000	2002,391	1996,171	1999,262	2005,189	2000,876
SSE		8.8463e-10	0.00025355	0.00023427	0.011897	0.0006847
Süre(sn)		2.9389	2.7365	2.6328	2.7206	2.6017

Tablo 4.25 N=2000 adet veri için PSO sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1005,795	1006,195	998,150	1007,495	997,645
Yay Sbt(K)	10000	10052,045	10051,578	9979,649	10062,671	9996,350
Sönüm Sbt(C)	2000	2029,967	2032,788	2016,149	2038,396	2018,134
SSE		0.0072051	0.0097415	0.0082764	0.012911	0.022079
Süre(sn)		2.7859	2.7912	2.6116	2.6219	2.5943

Tablo 4.26 N=2000 adet veri için ABA sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1006,233	1007,066	1002,720	1006,967	995,595
Yay Sbt(K)	10000	10045,217	10048,929	10013,907	10046,249	9981,467
Sönüm Sbt(C)	2000	2068,539	2078,145	2064,880	2076,788	1998,756
SSE		0.065314	0.085682	0.072273	0.083841	0.016039
Süre(sn)		18.8688	19.3437	18.8761	19.0445	17.7184

Tablo 4.27 0.30 N=2000 adet veri için EYA sonuçları

Gürültü Parametre	Orijinal	%0	%1	%5	%10	%20
Kütle(M)	1000	1001,191	1001,346	1002,016	1000,2072	1001,351
Yay Sbt(K)	10000	10011,913	10011,266	10024,229	9999,580	10039,132
Sönüm Sbt(C)	2000	2002,391	2003,127	2006,105	1997,164	2010,281
SSE		1.7692e-9	9.9594e-5	0.00048539	0.00040581	0.015976
Süre(sn)		3.8601	3.7875	3.654	3.5675	3.7103

Referans sistem cevabının veri büyüklüğü 2000 olarak kullanıldığında gürültüsüz, %1, %5, %10 ve %20 gürültü eklenen referans sistem cevaplarında EYA algoritmasının diğer algoritmalarla kıyasla SSE değerlerinin daha iyi çıktığı Tablo 4.25-4.27 arasına gösterilmektedir. Zaman performansı açısından kıyaslandığında en iyi performanstan kötüye doğru PSO, EYA ve ABA olarak sıralanmaktadır

4.2.5. Yay Sabiti ve Sönümleme Sabitinin Kimliklendirilmesi

Sisteme 50 kg ile 240 kg arasında 10 kilo aralıklarla kütle uygulanmaktadır. EYA, ABA, PSO ile uygulanan her bir kütleye karşılık birer adet 'K' ve 'C' değeri belirlenmiştir. Sistemin 'K' ve 'C' değerleri, belirlenen değerlerin ortalaması alınarak sabitleştirilmiştir. 50 kg ile 240 kg arasında, 'K' ve 'C' değerlerinin belirlenmesinde kullanılmayan ağırlıklar ve bu aralık içinde olmayan kütleler sisteme uygulanmıştır. Sistem cevabının %15lik kısmı referans olarak kullanıldığında 'M = 55, 135,215 ve 300'kg ağırlıklarla test edildiğinde algoritmaların başarılı olduğu gözlemlenmiştir. Çalışmalar gürültüsüz, %1, %5, %10, %20 oranında gürültü paternleri kullanılarak test edilmiştir. Bulgular grafik ve tablolar halinde paylaşılmıştır.

4.2.5.1. Veri Setinin Hazırlanması

Kimliklendirme işleminde kullanılmak üzere veri seti oluşturulmuştur. Çalışmanın ilk kısmı için, 'K' ve 'C' parametrelerinin belirlenmesi yapılmaktadır. Bu adımda kullanılan referans sistem cevabının üretilmesinde 'M = 50-240 Kg, 'K = 1000 N/m' ve 'C = 50 N/(m/s)' olarak alınmıştır[13]. 50 kg ile 240 kg arasında 10 kg aralıklarla arasında kütle uygulaması yapılmıştır. ABA, PSO, EYA kullanılarak, her 'M' değerine karşılık birer 'K' ve 'C' değeri bulunmuştur. Bulunan değerlerin ortalaması 'K' ve 'C' için ayrı ayrı alınmıştır. Bu ortalamalar, sabit 'K' ve 'C' değerleri olarak belirlenmiştir.

Bu çalışmanın ikinci kısmında 50 kg ile 240 kg aralığında olan ve bu aralık dışında kütleler uygulanmıştır. Kütleler kimliklendirme işlemi içerisinde kullanılmayan değerlerden seçilmiştir. ‘M = 55, 135, 215, 300’. Bu kütleler ile üretilen sistem cevabının ilk %15lik kısmının referans olarak kullanılmasında, alınan sonuçlara bakılarak, algoritmaların başarılı olduğu gözlemlenmiştir.

Dinamik ağırlık ölçme sistemi, diğer dinamik sistemler gibi iç ve dış etkilere fazlasıyla etkilenebilmektedir. Bu durum göz önüne alınarak %1, %5, %10 ve %20 gürültü referans sistem cevabına eklenmiştir. Çalışmanın adımları gürültülü ve gürültüsüz referans sistem cevapları kullanılarak yapılmıştır. Sonuçlar bu veriler üzerinde incelenmiştir. Algoritmaların parametrelerin belirlenmesinde başarılı olduğu gözlemlenmiştir

4.2.5.2. Yay ve Sönümleme Sabitlerinin Kimliklendirilmesi

Simülasyon üzerinde referans sistem cevabının üretilmesi için kullanılan platform parametrelerinin değerleri ‘M = [50:10:240] Kg’, ‘ $m_s = 0$ ’, ‘K= 1000 N/m’, ‘C = 50 N/(m/s)’, t ile ifade edilen zaman aralığı 0 ile 4 saniye arasında 0.01 aralıkla gösterilmektedir. Bulunan sonuçlar yay ve sönüm sabitlerinin alt ve üst arama limitleri 0 ile 10000 arasında belirlenmiştir[13].

• PSO Kimliklendirme Sonuçları

Algoritmanın maksimum adım sayısı 100, popülasyon büyüklüğü 15 olarak belirlenmiştir PSO ile bulunan yay ve sönüm sabitleri Tablo 4.28 ile gösterilmektedir.

Problemin çözümünde kullanılan algoritmanın parametre değerleri:

- Popülasyon büyüklüğü, $n_{pop} = 15$,
- Maksimum adım sayısı, $it_{max} = 100$,
- Üç bilinmeyen için parametre boyutu, $d = 2$,
- Parçacıkların arama ağırlığını belirleyen değişken, $w = 0.6$,
- ‘w’ parametresinin her adımda ayarlanmasını sağlayan katsayı, $w_{damp} = 0.7$,

• ABA Kimliklendirme Sonuçları

Algoritmanın maksimum adım sayısı 100, popülasyon büyüklüğü 15 olarak belirlenmiştir ABA ile bulunan yay ve sönüm sabitleri Tablo 4.29 ile gösterilmektedir.

ABA parametre değerleri:

- Popülasyon sayısı, $n_{pop}=15$,
- Maksimum adım sayısı $it_{max}=100$,
- Parametre boyutu, $d=2$,
- Işık emilim katsayısı $\gamma=0.9$,
- Çekicilik katsayısı temel değişkeni, $\beta_0=2$,
- Mutasyon katsayısı, $\alpha=0.9$,
- Mutasyon katsayısının her adımdaki değişme oranı, $\alpha_{damp}=0.7$ olarak belirlenmiştir.

Tablo 4.28 PSO ile bulunan yay ve sönümleme sabitlerinin ortalamaları

Parametre Gürültü	Yay Sabiti(K)	Sönüm Sabiti(C)	SSE
%0	1002,447	49,906	0.0197
%1 Gürültülü	1001,945	49,920	0.0339
%5 Gürültülü	1002,1092	49,834	0.0569
%10 Gürültülü	1003,459	49,819	0.0972
%20 Gürültülü	1002,550	50,691	0.1822
Süre(sn.)	148.5885		

Tablo 4.29 ABA ile bulunan yay ve sönümleme sabitlerinin ortalamaları

Parametre Gürültü	Yay Sabiti(K)	Sönüm Sabiti(C)	SSE
%0	1000	50	1.8335e-30
%1 Gürültülü	999,983	50,049	2.0474e-04
%5 Gürültülü	999,936	49,989	0.0041
%10 Gürültülü	1000,022	50,070	0.0158
%20 Gürültülü	1000,220	50,242	0.0723
Süre(sn.)	935.2068		

Tablo 4.30 EYA ile bulunan yay ve sönümleme sabitlerinin ortalamaları

Parametre Gürültü	Yay Sabiti(K)	Sönüm Sabiti(C)	SSE
%0	999,977	50,018	7.6936e-05
%1 Gürültülü	999,985	50,050	3.3301e-04
%5 Gürültülü	999,906	49,981	0.0042
%10 Gürültülü	999,958	50,063	0.0158
%20 Gürültülü	1000,245	50,220	0.0720
Süre(sn.)	220.6556		

• **EYA Kimliklendirme Sonuçları**

Algoritmanın maksimum adım sayısı 100, popülasyon büyüklüğü 20 olarak belirlenmiştir EYA ile bulunan yay ve sönüm sabitleri

Tablo 4.30 ile gösterilmektedir. EYA parametre değerleri:

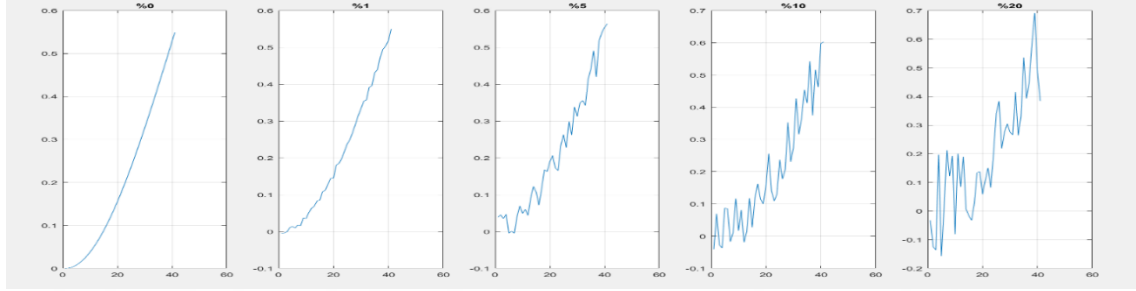
- Popülasyon büyüklüğü, $n_{pop}=20$, emperyalist sayısı, $n_{emp}=8$,
- Maksimum adım sayısı, $it_{max}=100$, parametre boyutu, $d=2$,
- Seçim baskısı, $\alpha=0.5$,
- Asimilasyon katsayısı, $\beta=1.25$,
- Devrim olma olasılığı, $p_{rev}=0.4$,
- Devrim oranı, $\mu=0.15$,
- Ortalama koloni uygunluk değeri hesaplama katsayısı, $\varepsilon=0.5$ olarak belirlenmiştir.

4.3. Ön Tahmin(Prediction) ile Kütlenin Belirlenmesi

4.3.1. Ön Tahmin ile Kütlenin Belirlenmesinde Bölge Seçimi

Bu bölüm altında sezgisel yöntemlerle erken safhada kütle(m) belirlenmiştir. ‘K’ ve ‘C’ değerleri bilindiği varsayılarak, ‘M’ değeri belirlenmiştir. 400 veri içeren sistem cevabının %10’luk kısmı referans alınarak belirleme işlemi tamamlanmıştır. Yay sabiti ‘K’ ve sönüm sabiti ‘C’, sırasıyla 10000 ve 2000 olarak belirlenmiştir. Burada yapılan çalışma iki kısımdan oluşmaktadır. İlk olarak sistem cevabının $t = 0$ ile 0.4 arasındaki yer değiştirme miktarı referans olarak kullanılmıştır. Kullanılan verilerin sırasıyla gürültüsüz, %1, %5, %10 ve %20 gürültü olarak Şekil 4.17 ile gösterilmektedir.

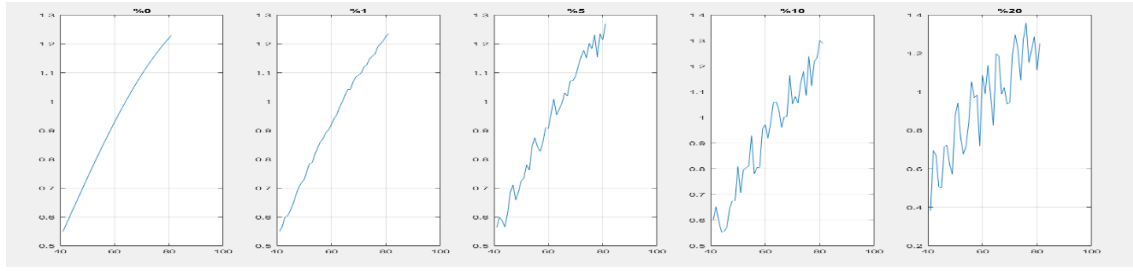
İkinci kısmında sistem cevabının rastgele bir kısmından alınan $t \neq 0$ ile sistem cevabı büyüklüğünün %10’luk kısmı arasındaki yer değiştirme miktarı referans olarak kullanmıştır.



Şekil 4.17 $t = 0$ anından $N = 40$ veri için sırasıyla %0, %1, %5, %10 ve %20 referanslar

Tablo 4.31 $t = 0$ anından $N = 40$ veri için kütlelin bulunması

Gürültü		%0	%1	%5	%10	%20
Algoritma						
PSO	Bulunan	1000	1001,428716	1076,510323	1018,205161	920,3020165
	SSE	0	0.00093837	2.7163	0.1527	2.8852
	Süre(sn.)	1.5385	1.3985	1.3978	1.3991	1.4014
ABA	Bulunan	1000	1001,428717	1076,510317	1018,205171	920,302014
	SSE	0	0.00093837	2.7163	0.1527	2.8852
	Süre(sn.)	24.8454	15.8066	10.067	7.8889	7.0274
EYA	Bulunan	1000,000427	1001,420004	1076,49001	1018,205175	920,3019069
	SSE	8.3727e-11	0.00092696	2.7148	0.1527	2.8852
	Süre(sn.)	1.7141	1.5939	1.3439	1.7605	1.6538



Şekil 4.18 $t \neq 0$ anından $N=40$ veri için sırasıyla %0, %1, %5, %10 ve %20 referanslar

Bulunan parametre değerlerine dayanarak gürültü oranını %10 ve altında kalan kısmında erken safha da yapılan parametre bulma çalışmasının başarılı olmuştur. %20'lik gürültü oranı sahip referans sistem cevabının $t = 0$ ile 0.4 arasındaki yer değiştirme miktarı kullanıldığında belirlenen parametrenin sapma oranının arttığı gözlemlenmektedir. Alınan sonuçlar Tablo 4.31 ile gösterilmektedir.

Çalışmanın bu bölümünün ikinci kısmında sistem cevabının orta bölümlerinden bir parça alınarak parametrelerin bulunması işlemi yapılmaktadır. Alınan sistem cevabı parçası Şekil 4.18 ile gösterilmektedir.

Tablo 4.32 $t \neq 0$ anından $N=40$ veri için kütlenin bulunması

Gürültü Algoritma		%0	%1	%5	%10	%20
PSO	Bulunan	1000	999,445291	1005,58614	1001,33737	1032,716395
	SSE	0	0.00014141	0.014353	0.00082221	0.49408
	Süre(sn.)	1.4838	1.4255	1.3871	1.3875	1.4283
ABA	Bulunan	1000	999,445291	1005,58614	1001,33737	1032,716395
	SSE	0	0.00014141	0.014353	0.00082221	0.49408
	Süre(sn.)	9.5473	9.8126	9.2341	8.244	6.9961
EYA	Bulunan	999,999392	999,456288	1005,59208	1001,33586	1032,716346
	SSE	1.6963e-10	0.00013586	0.014384	0.00082035	0.49408
	Süre(sn.)	2.2233	2.0401	1.71	1.6863	1.9142

Bulunan parametre değerlerine dayanarak gürültüsüz ve gürültülü, sistem cevabının $t=0.4$ ile 0.8 arasında kalan yer değiştirme miktarının referans olarak kullanılması, $t = 0$ ile

0.4 kısmından alınmasına nazaran daha iyi sonuçlar verdiği gözlemlenmiştir. Alınan sonuçlar Bulunan parametre değerlerine dayanarak gürültü oranını %10 ve altında kalan kısmında erken safha da yapılan parametre bulma çalışmasının başarılı olmuştur. %20'lik gürültü oranı sahip referans sistem cevabının $t = 0$ ile 0.4 arasındaki yer değiştirme miktarı kullanıldığında belirlenen parametrenin sapma oranının arttığı gözlemlenmektedir. Alınan sonuçlar Tablo 4.31 ile gösterilmektedir.

Çalışmanın bu bölümünün ikinci kısmında sistem cevabının orta bölümlerinden bir parça alınarak parametrelerin bulunması işlemi yapılmaktadır. Alınan sistem cevabı parçası Şekil 4.18 ile gösterilmektedir.

Tablo 4.32 ile gösterilmektedir.

4.3.2. Ön Tahmin ile Kütlenin Belirlenmesi

Yay ve sönüm sabitlerinin Bölüm 4.2.5' de belirlenmesinden sonra yay ve sönüm sabitlerinin belirlenmesinde kullanılmamış kütle değerlerinin uygulanmasıyla algoritmaların ön tahmin performansları test edilmektedir.

4.2.5 bölümü ile verilen algoritma parametreleriyle yapılan çalışmada kütlenin, gürültüsüz, %1, %5, %10 ve %20 oranında gürültülü referans sistem cevapları kullanılarak bulunan değerler Tablo 4.33 ile gösterilmektedir. Algoritmaların ön tahmin performansı 'C' ve 'K' parametrelerinin belirlenmesinde kullanılmamış 55, 135, 215, 300 ile test edildiğinde, Algoritmaların kütle belirlemesini başarılı bir şekilde yapmaktadır.

- **PSO Sonuçları**

Algoritmanın maksimum adım sayısı 100, popülasyon büyüklüğü 15 ve parametre boyutu 1 olarak belirlenmiştir. PSO ile bulunan yay ve sönüm sabitleri Tablo 4.33 ile gösterilmektedir.

- **ABA Sonuçları**

Algoritmanın maksimum adım sayısı 100, popülasyon büyüklüğü 15 ve parametre boyutu 1 olarak belirlenmiştir. ABA ile bulunan yay ve sönüm sabitleri Tablo 4.34 ile gösterilmektedir.

- **EYA Sonuçları**

Algoritmanın maksimum adım sayısı 100, popülasyon büyüklüğü 20 ve parametre

boyutu 1 olarak belirlenmiştir. EYA ile bulunan yay ve sönüm sabitleri

Tablo 4.35 ile gösterilmektedir.

Parametre boyutunun ‘1’ olmasının sebebi ‘K’ ve ‘C’ değerlerinin kimliklendirilerek sabitlenmesidir.

Tablo 4.33 PSO ile bulunan kütle değerleri

Gürültü Orijinal Kütle		%0	%1	%5	%10	%20
55	Bulunan	55	55,00657	54,91577	55,58501	54,729
	SSE	0,001535081	0,001689	0,000262	0,045904	0,002041
	Süre(sn.)	2.0998	2.0115	1.9943	2.1332	2.0012
135	Bulunan	135	134,981	135,008	135,6727	135,2399
	SSE	0,012544	0,011095	0,013179	0,12304	0,038769
	Süre(sn.)	2.117	2.1156	2.0536	2.0142	2.1686
215	Bulunan	215	214,9446	215,0831	216,3944	215,7554
	SSE	0,028281	0,022672	0,037889	0,379124	0,168239
	Süre(sn.)	2.1464	2.0015	2.081	2.0585	2.1095
300	Bulunan	300	299,6518	300,0878	301,3386	301,2412
	SSE	0,059269	0,016402	0,074407	0,479258	0,435043
	Süre(sn.)	2.0877	2.1238	2.067	2.1595	2.1903

Tablo 4.34 ABA ile bulunan kütle değerleri

Gürültü Orijinal Kütle		%0	%1	%5	%10	%20
55	Bulunan	55	55,00218	55,10033	55,53987	55,48592
	SSE	0	4,29E-07	0,000913	0,026397	0,021389
	Süre(sn.)	13.9981	14.1253	14.1253	13.5601	12.9334
135	Bulunan	135	135,0193	135,2193	135,533	135,2909
	SSE	0	4,76E-05	0,006126	0,036144	0,010777
	Süre(sn.)	15.3142	15.208	14.4485	13.3329	12.7502
215	Bulunan	215	215,0493	215,5001	216,0417	215,4459
	SSE	0	0,000251	0,025851	0,112269	0,020556
	Süre(sn.)	14.4524	14.4995	14.0311	13.7661	12.7612
300	Bulunan	300	300,1826	298,963	299,5979	296,3593
	SSE	0	0,003764	0,121412	0,018253	1,495756
	Süre(sn.)	14.4946	14.3635	13.8058	13.3686	9.1408

Tablo 4.35 EYA ile bulunan kütle değerleri

Gürültü		%0	%1	%5	%10	%20
Orijinal Kütle						
55	Bulunan	55,00015	54,98683	54,7785	54,85292	55,33765
	SSE	8,81E-07	1,77E-05	0,00447	0,001975	0,010304
	Süre(sn.)	2.6866	2.6613	2.6749	2.8754	2.593
135	Bulunan	134,9996	134,9662	134,7329	134,8341	135,1075
	SSE	3,45E-06	0,000159	0,009192	0,003566	0,001445
	Süre(sn.)	2.7861	2.6621	2.7173	2.7194	2.6912
215	Bulunan	214,9882	214,9176	214,4509	214,6968	214,9536
	SSE	3,13E-05	0,00078	0,031608	0,009766	0,000269
	Süre(sn.)	2.2616	2.0356	2.0812	2.1063	2.0012
300	Bulunan	300,0194	299,6527	299,845	299,3939	297,7934
	SSE	2,86E-05	0,014069	0,00292	0,042261	0,552419
	Süre(sn.)	2.5379	2.4478	2.409	2.4727	2.4509

Daha az veri kullanılarak Kütlenin ön tahmin işleminin yapılması işleminde %0, %1, %5, %10 ve %20 oranlarında gürültüye sahip sistem cevaplarıyla yapılan çalışmada %0 ve %1 gürültüde ABA algoritması başarılı olmuştur. %5 gürültü oranında PSO başarılı olmuştur. %10 gürültü oranında EYA ve %20 gürültü oranında 135 ve 215 kg kütlelerin tahmininde EYA, 55 ve 300 kg kütlelerin tahmininde PSO başarılı olmuştur.

4.4. Sonuçların Değerlendirilmesi

Tez çalışmasının 4.2 ile gösterilen bölüm üç kısımdan oluşmaktadır. İlk kısımda birer adet ‘M’, ‘C’ ve ‘K’ değerlerine karşılık kimliklendirme performansları incelenmiştir.

Çalışma sonucunda %0, %1, %5 ve %10 gürültülü oranlarına sahip sistem cevapları kullanılarak elde edilen en iyi SSE değerleri sırasıyla 3.5787.e-10, 0.00013424, 0.0021482, 0.00063252 olarak EYA ile bulunmuştur. %20 gürültü oranına sahip sistem cevabı ile yapılan çalışmada sonucunda elde edilen en iyi SSE değeri 0.0008843 olarak PSO ile elde edilmiştir.

EYA ile yapılan çalışmada ‘M’ değerleri sırasıyla, 1001.191, 1001.047, 1002.416 ve 1002.459 olarak, ‘C’ değerleri, 2002.391, 2007.739, 2015.651, 2002.552 olarak ve ‘K’ değerleri 10011.913, 10009.969, 10042.649 ve 10037.028 olarak elde edilmiştir. PSO

ile yapılan çalışmada 'M' değeri, 999.06 olarak, 'C' değeri, 1988.77 olarak ve 'K' değeri, 9992,158 olarak elde edilmiştir.

Algoritmaların saniye cinsinden çalışma sürelerine bakıldığında, PSO algoritmasının 1-2 saniye arasında, EYA 'nın 1-3 saniye arasında ve ABA 'nın 21-25 saniye arasında koşma sürelerine sahip olduğu görülmektedir. Süre performansı açısından en iyi performansın PSO algoritmasına ait olduğu, ardından sırasıyla EYA ve ABA' nın geldiği görülmektedir.

Çalışmanın geneline bakıldığında SSE değerlerine göre EYA, süre değerlerine göre PSO iyi performans değerlerine sahip olduğu görülmektedir.

İkinci kısımda zaman aralığı değiştirilmeden veri seti büyüklüğünün değiştirilmesinin, algoritmaların kimliklendirme performanslarına etkisi incelenmiştir. Referans sistem cevabının içerdiği nokta sayısının 100, 500, 1000 ve 2000 olması durumlarını Bölüm 3 ile gösterilmektedir. Optimizasyon algoritmaları kullanılarak elde edilen parametre değerleri ile orijinal sistem parametre değerleri karşılaştırıldığında çok az veri sayısı kullanıldığında orijinal değerden ciddi sapmalar olmakla birlikte çok miktarda veri sayısı kullanımında ise orijinal değere göre çok küçük miktarlarda sapmalara yol açtığı gözlemlenmiştir.

Üçüncü kısmında ise 'K' ve 'C' parametrelerinin kimliklendirme işlemi yapılmıştır. Kütlelerin belirli sınırlar içinde dinamik ağırlık ölçme sistem modeline ayrı ayrı uygulanmasıyla bulunan 'K' ve 'C' değerlerinin ayrı ayrı ortalamaları alınarak parametre değerleri atanmıştır.

'K' ve 'C' parametrelerinin kimliklendirme işleminin sonuçları Tablo 4.28-4.30 arasında gösterilmektedir. ABA' nın tablolarda gösterilen SSE değerine bakılarak, %0 , %1, %5 ve %10 gürültü oranına sahip sistem cevapları ile yapılan kimliklendirme çalışmasının diğer algoritmalara nazaran performansının daha iyi olduğu görülmektedir. %20 gürültü oranında EYA 'nın en iyi SSE değerlerine sahip olduğu görülmektedir.

Bölüm 4.3 ile yapılan çalışma iki kısımdan oluşmaktadır. İlk kısmında kimliklendirmenin tamamlanmış olduğu bir sistem için geçici rejim cevabının farklı bölgelerinde algoritmaların ön tahmin ile kütlenin belirlenmesinde ki performansları

incelenmiştir. Ön tahmin için geçici rejim cevabının iki farklı bölgesinden alınan parçalar ile test yapılmıştır. Kütlenin ön tahmin ile belirlenmesinde algoritmaların belirlediği kütle değerleri birbirlerine çok yakın çıkmaktadır. Bunun sebebi belli bir kütle aralığına bağlı olmadan 'K' ve 'C' değerlerinin sabit olarak verilmesinden kaynaklanmaktadır. Bu değerlere bağlı olarak bulunmak istenen 'M' değerleri birbirine yakın çıkmaktadır. Bu çalışma iki farklı sistem cevabı bölgesinden alınan değerlere göre karşılaştırıldığında $t=0$ anından itibaren alınan rejim cevabının küçük bir kısmını içeren vektör ile yapılan çalışmanın sonucunda %20 gürültüde tahminlerin kötüleştiği görülmektedir. Rejim cevabının küçük bir kısmını içeren vektörün $t \neq 0$ anından itibaren alınmasıyla çalışma içerisinde kullanılan bütün gürültü değerlerinde sonuçların başarılı olduğu görülmektedir.

İkinci kısımda ise ön tahmin ile kütlenin belirlenmesi çalışmalarında, gürültüsüz ve gürültülü sistem cevapları ile yapılan çalışmanın incelenmesi sonucunda; Ateş Böceği algoritmasının %0 ve %1 gürültü oranına sahip sistem cevapları ile yapılan çalışmalar arasında en iyi parametre değerlerine sahip olduğu görülmektedir. %5 gürültü oranına sahip sistem cevabı ile yapılan çalışmalar arasında PSO algoritmasının en iyi parametre değerlerine sahip olduğu görülmektedir. %10 gürültü oranına sahip sistem cevapları ile yapılan kütle tahmini çalışmalarında EYA' nın test verilerinin %10 gürültü oranına sahip sistem cevabı ile yapılan çalışmasında iyi parametre değerlerine sahip olduğu gözlemlenmiştir. Hem PSO hem de EYA %20 gürültü oranına sahip sistem cevapları ile yapılan çalışmalarda farklı kütle değerlerinde başarılı sonuçlar verdikleri görülmektedir.

BÖLÜM 5

TARTIŞMA SONUÇ VE ÖNERİLER

Dinamik ağırlık ölçme sistemi, bir kütlenin doğru ve hızlı bir şekilde belirlenmesinde yaygın olarak kullanılan doğrusal olmayan bir sistemdir.

Endüstriyel alanlarda yaygın olarak kullanılan dinamik ağırlık ölçme sistemin az sönümlü modeli üzerinden geçici rejim cevabı elde edilmektedir. Tez çalışmasında, sezgisel optimizasyon algoritmaları kullanılarak az sönümlü modele ait kütle(M), sönümleme sabiti(C) ve yay sabiti(K) değerlerinin belirlenmesi amaçlanmıştır. Sezgisel optimizasyon algoritmalarından EYA, ABA ve PSO yöntemleri modelin giriş/çıkış verilerine göre dinamik ağırlık ölçme sisteminin kimliklendirilmesi problemine başarılı bir şekilde uygulanmıştır. Az sönümlü modelin orijinal parametre değerleri kullanılarak sistemin geçici rejim cevapları simülasyon ortamında elde edilmiştir. Simülasyon ortamında üretilen bu geçici rejim cevapları referans veri seti olarak kullanılmıştır.

Kimliklendirme çalışmasında ‘ M ’, ‘ C ’ ve ‘ K ’ değerlerinin iyileştirilebilmesi için algoritmalar tarafından bulunan parametre değerleri modele uygulanarak yeni sistem cevapları elde edilmiştir. Bulunan yeni sistem cevapları, referans sistem cevaplarına benzetilerek orijinal parametre değerlerinin bulunması işlemi yapılmıştır.

Gürültü eklenmemiş sistem cevapları ile yapılan çalışma sonucunda algoritmalar kendi aralarında kıyaslandığında EYA yönteminin en iyi SSE değerlerine sahip algoritma olduğu görülmüştür.

Optimizasyon algoritmaları tarafından referans olarak alınan sistem cevaplarının birim zamanda sahip oldukları veri sayısındaki değişiminin algoritmaların kimliklendirme çalışması üzerindeki etkisi test edilmiştir. Optimizasyon algoritmaları kullanılarak elde edilen parametre değerleri ile orijinal sistem parametre değerleri karşılaştırıldığı zaman çok az veri sayısı kullanıldığında orijinal değerden ciddi sapmalar olmakla birlikte çok

miktarda veri sayısı kullanımında ise orijinal değere göre çok küçük miktarlarda sapmalara yol açtığı gözlemlenmiştir.

Kimliklendirme için kütle değeri belirli aralıklarla arttırılarak her bir uygulanan kütle(M) değerlerine karşılık az sönümlü sisteme ait geçici rejim cevapları için veri seti üretilmiştir. Algoritmalar tarafından geçici rejim cevaplarını referans alınarak sönümleme(C) ve yay(K) sabiti değerleri belirlenmiştir. Bu işlemde her algoritma ayrı bir kimliklendirme işlemi yapmakta ve farklı ‘C’ ve ‘K’ değerleri bulmaktadır. Bulunan ‘C’ ve ‘K’ değerlerinin ortalamaları alınarak yay ve sönüm sabitleri belirlenmiştir. Ayrıca bu çalışmada %1, %5, %10 ve %20 gürültü oranlarına sahip beyaz gürültü(white noise) eklenmiş geçici rejim cevapları için işlemler tekrarlanmıştır.

Gürültü eklenmemiş sistem cevapları ile yapılan çalışma sonucunda algoritmalar kendi aralarında kıyaslandığında ABA yönteminin en iyi SSE değerlerine sahip olduğu görülmüştür.

Kütlenin ön tahmin ile belirlenmesi işleminde ise, $t=0$ ve $t \neq 0$ durumları için gürültü ve gürültüsüz sistem cevaplarına ait daha az veri seti kullanılarak kütle değerinin daha erken bulunması amaçlanmış ve uygulanan optimizasyon algoritmalarının tamamında oldukça başarılı sonuçlar alınmıştır.

Gürültülü ve gürültüsüz sistem cevaplarından $t = 0$ anından itibaren alınan $N=40$ adet veri kullanılarak algoritmaların kütleyi ön tahmin ile bulma işlemindeki performanslarının birbirlerine yakın oldukları gözlemlenmiştir. Gürültülü sistem cevapları kullanılarak tahmin edilen kütle değeri ile orijinal kütle değerleri arasındaki sapmanın arttığı gözlemlenmiştir. $t \neq 0$ anından itibaren alınan gürültülü ve gürültüsüz sistem cevaplarının $N = 40$ adet verisini içeren bölgede kütle değerlerinin orijinallerine yakın bulunduğu gözlemlenmiştir. Bunun sebebi $t = 0$ anından itibaren alınan kısa sistem cevabı doğrusala yakın bir grafik oluştururken, $t \neq 0$ anından itibaren alınan sistem cevabında sinyalin sürekli zaman seviyesine daha yakın oluşu sonucu iyi yönde etkilemektedir. Çalışma sonucunda PSO ve EYA tahmin işlemlerinde ABA’ ya göre daha hızlı sonuçlar verdiği gözlemlenmiştir.

Kimliklendirme işlemindeki simülasyon veri setinde kullanılan kütle değerlerinden farklı rastgele seçilen kütle değerleri(55, 135, 215 ve 300) ile oluşturulan test veri seti,

sisteme uygulanarak algoritmaların kütle tahmini konusundaki performansları test edilmiştir. Üretilen sistem cevabının $t \neq 0$ anından itibaren ilk %15'lik kısmı kullanılarak kütle değerinin ön tahmini için uygulanan algoritmalarla dikkate değer başarılı sonuçlar elde edilmiştir.

Simülasyon çalışmaları sonucunda önerilen algoritmaların probleme başarıyla uygulandığı gözlemlenmiştir. Bulunan parametre değerleri, orijinal parametre değerleri ile kıyaslandığında oldukça başarılı sonuçlar alındığı gözlemlenmiştir.

Elde edilen sonuçlar göz önüne alındığında, sezgisel optimizasyon algoritmaları doğrusal olmayan sistem kimliklendirme işlemleri başta olmak üzere parametre tahmin yöntemlerinde alternatif bir çözüm olarak kullanılabilir.

Endüstriyel alanlarda dijital işaret işleme, regresyon ve adaptif filtreleme tekniklerine dayalı mikrobilgisayarlar gerçek zamanlı uygulamalar için yaygın olarak kullanılmaktadır. Çalışmada kullanılan algoritmaların cevap süreleri bakımından iyileştirilmeleri yapıldığı takdirde online uygulamalar için kullanılabileceği ön görülmektedir. Offline parametre belirleme işleminde sezgisel algoritmaların kullanılmasıyla parametre değerlerinin orijinale yakın değerler bulunduğu gözlemlenmiştir.

İleride yapılacak çalışmalar için optimizasyon algoritmalarının hibrit kullanımları, paralel işaret işleme teknikleri kullanılarak algoritmaların cevap sürelerinin iyileştirilmesi çalışmaları yapılabilir.

KAYNAKÇA

1. Ljung, L., 1987. System Identification, Prentice-Hall, Inc, Englewood Cliffs, New Jersey, 519 pp.
2. Candy, J.V., 1989 Signal Processing: The Model-Based Approach, Mcgraw Hill.
3. Giordano, F.R. & Weir, M.D., 1991. Differential Equations, A Modelling Approach, Addison-Wesley Publishing Co., NY
4. Marquardt, D.W., 1963. An algorithm for least squares estimation of non-linear parameters, **Journal of Society for Industrial & Applied Mathematics**, **11**:431-441.
5. Box, M.J., Davies, D., Swann, W.H., 1969. Nonlinear Optimization Techniques, Imperial Chemical Industries Ltd., UK.
6. Gallant, A.R., 1975. Nonlinear regression, **The American Statistician**, **29**:73-81
7. Burden, R.L. & Faires, J.D., 1983. Numerical Analysis, Prindle, Weber & Schmidt, 3rd Ed., Boston.
8. Nash, J.C., Smith, M.W., 1987. Non-Linear Parameter Estimation: An Integrated System İn Basic, Marcel Dekker Inc., NY
9. Sudarno, F., Rotella, I.Z., 1993. improvements in least squares methods for identification, pp. 227 - 229, *IEEE International Conference on Systems, Man & Cybernetics*, 17-20 Oct., 1993, Le Touquet, France
10. Karaboğa, D., 2011. Yapay Zekâ Optimizasyon Algoritmaları, Nobel Yayın Dağıtım, Kayseri, 246 s.
11. Shi, W.J., White, N.M., Brignell, J.E., 1993. Adaptive filters in load cell response correction, **Sensors & Actuators A**, **37**:280-285
12. Shu, W.Q., 1993. Dynamic weighing under non-zero initial conditions, **IEEE Transactions on Inst. & Meas.**, **42**(4): 806-811
13. Danaci, M. and Horrocks, D.H., 1995. A non-linear regression technique for improved dynamic weighing, pp. 507-510, *Proc. Euro. Conf. on Circuit Theory and Design*, Aug., 1995, Istanbul, Turkey.
14. Horrocks D.H. and Danaci M., 1995, High speed modelling by non-linear auto regression, *ISEM'95 Symposium*, D19, Sep.1995, Cardiff, UK,

15. Danaci, M. 1996, Non-Linear Regression Techniques For Dynamic Weighing Systems Ph.D. Thesis. Cardiff University of Wales, UK.
16. Horrocks, D.H., Danaci, M., 1996. High speed modeling for dynamic weighing by non-linear auto regression, **Nonlinear Electromagnetic Systems**, (Edited by Moses, A.J. and Basak, A.), IOS Press, 548-551, UK.
17. Danaci M. Kalınlı A., 1999. Detection of the starting of the newly applied mass for successive weighing, **Mathematical & Computational Applications**, 4(1): 53-59.
18. Danaci M., Yaz M., 1999. New approaches to successive continuous weighing, pp. 584-591, *Proceedings of the IASTED International Conference Control And Applications Banff*, July 25-30, 1999, Canada.
19. Danaci, M., 2000. Platform parameters identification by non-linear regression method, pp. 811-818, *14th European Simulation Multiconference, ESM'2000*, 23 – 26 May, 2000, Ghent, Belgium.
20. Danaci M., 2001. New approaches for high speed and accurate weight measurements, pp. 277-286, *Proceedings of the 17th International conference on Force, Mass, Torque and Pressure Measurements*, 17-21 Sept., 2001, Istanbul, Turkey.
21. Danacı M. Sağiroğlu Ş., 2001. A neural system for an automatic weighing, pp. 293-298, *Proceedings of the 17th International Conference on Force, Mass, Torque and Pressure Measurements, IMEKO TC3*, 17-21 Sept. 2001, Istanbul, Turkey.
22. Liao G. and Xi J., 2010. Design and implement of vehicle dynamic weighing system, pp. 1131-1134 *International Conference on Intelligent Computation Technology and Automation*, 11-12 May, 2010, Changsha, China.
23. Jeridi W., Benabdallah S., Hamdi M., Boudriga N., 2010. Dynamic expert weighing for security risk analysis team synergy, pp. 1 - 8, *Engineering Systems Management and Its Applications (ICESMA), 2010 Second International Conference on*, 30 March-1 April, 2010, Sharjah, United Arab Emirates.
24. Hu P., Liang L., Hu J., 2010. The control strategy and application for dynamic quantitative weighing system based on embedded linux pan, pp. 4917-4921, *Proceedings of the 8th World Congress on Intelligent Control and Automation*, , Jinan, China.
25. Sun J., Liu F., Si J., Mei S., 2010. Direct heuristic dynamic programming based on

- an improved pid neural network and initial weighs choosing method, 7-9 July, 2010 *Critical Infrastructure (CRIS), 2010 5th International Conference on*, 1-6, China.
26. Ying Z., Yunbin H., Min L., 2011. The research on data processing for dynamic batching-weighing system, pp. 888-890, *The 6th International Conference on Computer Science & Education (ICCSE 2011)*, 3-5 Aug., 2011, Singapore, Singapore.
 27. Martin A. D., Molteno T. C. A., 2015. Automated weighing by sequential inference in dynamic environments, pp. 274-278, *Proceedings of the 6th International Conference on Automation, Robotics and Applications*, 17-19 Feb. 2015, Queenstown, New Zealand.
 28. Niedźwiecki M., Pietrzak P., 2016. High-precision fir-model-based dynamic weighing system, **IEEE Transactions on Instrumentation and Measurement**, 65(10): 2349 – 2359.
 29. Hamilic M., Balacharhan W., 1995. Kalman filter for dynamic weighing system, industrial electronics, pp. 786-791, *ISIE '95, Proceedings of the IEEE International Symposium on*, 10-14 July, 1995, Dubrovnik, Croatia.
 30. Hamilic M. Balachandran W., Enab Y., 1996. Fuzzy logic estimator for dynamic weighing system, Fuzzy Systems, pp. 2123-2129, *Proceedings of the Fifth IEEE International Conference on*, 11-11 Sept., 1996, New Orleans, LA, USA.
 31. Li J. Zhou L. Wang Q., 2007. Study on dynamic weighing system base on intelligent algorithm, 765-768, *IEEE International Conference on Control and Automation WeDP-17 Guangzhou*, 30 May-1 June, 2007, Guangzhou, China, .
 32. Xiaoyan C., Zhenliang, L., 2008. An intelligent dynamic weighing controller, pp. 1609-1612, *Proceedings of the IEEE International Conference on Automation and Logistics Qingdao*, 1-3 Sept., 2008, Qingdao, China.
 33. Gao H., Pang W., 2009, A high-accuracy dynamic weighing system based on single-idler conveyor belt, pp. 2483-2487, *Proceeding of the Eighth International Conference on Machine Learning and Cybernetics*, 2-15 July, 2009, Hebei, China.
 34. Wu Q., Pan R., Luo X., Li Y., 2009. A signal processing method for dynamic weighing system by ssa-lvq network, pp. 4-403 - 4-407, *The 9th International Conference on Electronic Measurement & Instruments*, 6-19 Aug., 2009,

Beijing, China.

35. Gray J.G., Murray-Smith, D.J., Li Y., Sharman K.C., Weinbrenner T., 1998, Nonlinear model structure identification using genetic programming, **Control Engineering Practice**, 6(11): 1341-1352.
36. Sendrescu D., Roman M., 2013, Parameter Identification of Bacterial Growth Bioprocesses using Particle Swarm Optimization, *2013 9th Asian Control Conference (ASCC)*, 23-26 June, 2013, Istanbul, Turkey.
37. Sendrescu D., Bobasu E., 2013, Parameter Identification of Bacterial Growth Bioprocesses using Heuristics for Global Optimization, pp. 485-490, *2013 17th International Conference on System Theory, Control and Computing (ICSTCC)*, 11-13 Oct. 2013, Sinaia, Romania
38. Ulinowicz M., Andrzejczak M., 2014 An Heuristic Approach to a Ship Nonlinear Model Identification, pp. 239-244, *2014 European Modelling Symposium*, 21-23 Oct., 2014, Pisa, Italy.
39. Kumar M., Aggarwal A. Rawat T.K., 2016, Bat Algorithm: Application to Adaptive Infinite Impulse Response System Identification, **Arabian Journal for Science and Engineering**, 41(9):3587-3604.
40. Ryzhikov S. and Semenkin E.S., 2016, Restart Operator Meta-heuristics for a Problem-Oriented Evolutionary Strategies Algorithm in Inverse Mathematical MISO Modelling Problem Solving, pp. 1-11, *V International Workshop on Mathematical Models and their Applications 2016*, 7–9 November, 2016, Krasnoyarsk, Russia.
41. Perrot, M. H., Cohen, R. J., 1996. An efficient approach to arma modelling of biological systems with multiple inputs and delays, **IEEE Transactions on Biomedical Engineering**, 43(1), 1-14
42. Özçelik Y., 2007 Farksal Evrim Algoritması Kullanarak Sistem Kimliklendirme, Yüksek Lisans Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Kayseri.
43. Eykhoff, P., 1974. System Identification, Parameter & State Estimation, **John Wiley & Sons** University of Southampton, UK
44. Close, C.M., Frederick, D.K., 1978. Modelling & Analysis of Dynamic Systems, Houghton Mifflin Co., London
45. Schwarzenbach, J., Gill, K.F., 1988. System Modelling & Control, Chapman & Hall, 2nd Ed., NY

46. Ogata, K., 1990. Modern Control Engineering, Prentice-Hall, 2nd Ed.
47. Golden, J., Verwer, A., 1991. Control System Design & Simulation, McGraw-Hill Book Co., UK
48. Ferguson, C., 1974. From beam to load cell, **Engineering**, pp.202-207.
49. **Brendal, A.,1979. High speed check weighing:part 1-4, Meas. and Cont. Vol.12**
50. Huggins, P., 1982. A high speed load cell, **Meas. & Cont.**, **15**:211-213.
51. Norden, K.E., Kersten, 1984. Electronic Weighing In Industrial Processes, Granada Technical Books, London.
52. King (Fellow), A., 1987. An introduction to weighing systems, **Electronic Technology**, **21**:25-29.
53. Allison, J.S. 1969. The estimation of parameters of dynamic systems, in least squares methods in data analysis, (Edited by Anderssen, R.S. & Osborne, M.R.), Univ. of Queensland Press.
54. Haberman, R., 1977. Mathematical Models, Mechanical Vibrations, Population Dynamics & Traffic Flow, Prentice-Hall Inc., Englewood Cliffs, London
55. Spriet, J.A., Vansteenkiste, G.C., 1982. Computer Aided Modeling And Simulation, **Academic Press**, London.
56. Ediger, M.N., 1990. A gauss-newton method for nonlinear regression, **Journal of The Mathematica**, **1**(2):42-44
57. Kupper, W.E., 1990. High accuracy mass measurement from micrograms to tons, **ISA Transactions**, **29**(4):11-20
58. Kersten, J. 1995. Weighing Systems. Control and Inst ., June. 23, 24, 26, 28.
59. Shi, W.J., Brignell, J.E., 1991. On-line optimisation in sensors frequency response, **Sensors & Actuators A**, **25**:37-41
60. Shi, W.J., 1992, Dynamic Frequency Compensation for Transducers, Ph.D. Thesis,
61. Alataş, B., 2007. Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları Geliştirme, Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.
62. Kennedy, J., Eberhart, R. C., 1995. Particle swarm optimization, pp. 1942-1948, *Proc. of IEEE International Conference on Neural Networks*, 27 Nov.-1 Dec., 1995, Perth, WA, Australia,
63. Sevkli M., 2005. Atölye Tipi Çizelgeleme Problemlerine Parçacık Sürü Optimizasyonu Yaklaşımı ve Genetik Algoritma Modeli ile Karşılaştırılması, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul.

64. Eberhart, R. C., Kennedy, J., 1995. A new optimizer using particle swarm theory, pp. 39 - 43, *Proc.of Sixth International Symposium on Micro Machine and Human Science*, 4-6 Oct., 1995, Nagoya, Japan.
65. X. S. Yang, 2009. Firefly algorithms for multimodal optimization, proceedings of the stochastic algorithms: foundations and applications, Lecture Notes in Computing Sciences, pp. 169-178, *5th International Symposium, SAGA 2009*, Springer, Sapporo, Japan.
66. Apostolopoulos, T., Vlachos, A., 2011. Application of the firefly algorithm for solving the economic emissions load dispatch problem, **Hindawi Publishing Corporation International Conference Journal of Combinatorics**, vol. 2011, Article ID 523806, 23 pages.
67. Yang X. S., 2010. Firefly algorithm, levy flights and global optimization, **Research and Development in Intelligent Systems XXVI**. 209-218, Springer, London, UK.
68. Atashpaz-Gargari E., Lucas C., 2007. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, pp. 4661–4667, *IEEE congress on evolutionary computation*, 25-28 Sept. 2007, Singapore, Singapore.
69. Ahmadi M.A., 2011. Prediction of asphaltene precipitation using artificial neural network optimized by imperialist competitive algorithm. **Journal of Petroleum Exploration and Production Technology**, 1(2-4): 99–106
70. Yazdipour A., Ghaderi M.R., 2014. Optimization of weld bead geometry in gtaw of cp titanium using imperialist competitive algorithm. **The International Journal of Advanced Manufacturing Technology**, 72(5-8):619-625.
71. Gargari A., 2007, The Development Of Optimization Algorithm Social And Review Of Performance, Master thesis, Faculty of Electrical Engineering and by computer, Tehran University, Iran.
72. Emami H., Derakhshan F., 2015. Integrating fuzzy k-means, particle swarm optimization, and imperialist competitive algorithm for data clustering. **Arabian Journal for Science and Engineering**, 40(12): 3545-3554.
73. Watson, W.A., Philipson, T. and Oates P.J. 1981. Numerical Analysis, the Mathematics of Computing. Edward Arnold Ltd. 2nd Ed., London.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Ahmet EMİN BAKTIR
Uyruğu: Türkiye (T.C)
Doğum Tarihi ve Yeri: 08.04.1991
Medeni Durum: EVLİ
e-mail: aebaktir@gmail.com
Yazışma Adresi: İlkyerleşim Mah. 1939 Sok. Meşale Batı Konutları No:4/4
Yenimahalle/Ankara

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi	-
Lisans	Erciyes Üniversitesi, Bilgisayar Mühendisliği	2015
Lisans	Erciyes Üniversitesi, Elektrik-Elektronik Mühendisliği	2016
Lise	Prof. Dr. Necati Erşen Anadolu Öğretmen Lisesi, Sivas	2009

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2017-Halen	Ankara Yıldırım Beyazıt Üniversitesi	Arş. Grv.
2016-2017	Turcosa Analitik Çözümler	Front-End geliştirici

YABANCI DİL

İngilizce(orta)

YAYINLAR

1. Kıvanç Ş.G., Baktır A.E., Şen B., 2018, Performance comparison of machine learning methods for solving handwriting character recognition problem, *International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES'18)*, May 11-13, 2018, Safranbolu, Turkey