

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

**KALMAN SÜZGEÇİ PARAMETRELERİNİN
ENİYİLENMESİNDE ZEKİ OPTİMİZASYON
TEKNİKLERİNİN BAŞARIMLARININ İNCELENMESİ**

**Hazırlayan
Tayyip ÖZCAN**

**Danışman
Doç. Dr. Alper BAŞTÜRK**

Yüksek Lisans Tezi

**Bu çalışma Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi
tarafından FYL-2016-6520 numaralı proje ile desteklenmiştir**

**Temmuz 2016
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.

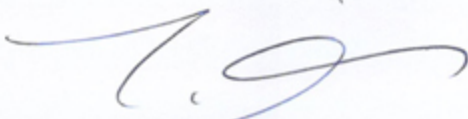
Tayyip ÖZCAN

İmza:



YÖNERGEYE UYGUNLUK

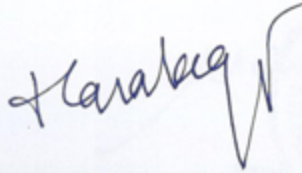
“Kalman Süzgeci Parametrelerinin Eniyilenmesinde Zeki Optimizasyon Tekniklerinin Başarımlarının İncelenmesi” adlı Yüksek Lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.



Tezi Hazırlayan
Tayyip ÖZCAN



Danışman
Doç. Dr. Alper BAŞTÜRK



Bilgisayar Mühendisliği ABD Başkanı
Prof. Dr. Derviş KARABOĞA

Doç. Dr. Alper BAŞTÜRK danışmanlığında Tayyip ÖZCAN tarafından hazırlanan “Kalman Süzgeci Parametrelerinin Eniyilenmesinde Zeki Optimizasyon Tekniklerinin Başarımlarının İncelenmesi” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

29/07/2016

JÜRİ :

Danışman : Doç. Dr. Alper BAŞTÜRK

Üye : Prof. Dr. Derviş KARABOĞA

Üye : Yrd. Doç. Dr. Asuman SAVAŞÇIHABEŞ

ONAY :

Bu tezin kabulü Enstitü Yönetim Kurulunun 09/05/2016 tarih ve 2016/35-17 sayılı kararı ile onaylanmıştır.

Prof. Dr. Mehmet AKKURT
Enstitü Müdürü

ÖNSÖZ / TEŞEKKÜR

"Kalman Süzgeci Parametrelerinin Eniyilenmesinde Zeki Optimizasyon Tekniklerinin Başarımlarının İncelenmesi" konulu tez çalışmamın gerçekleştirilmesinde bilgi ve tecrübelerini esirgemeyen saygıdeğer danışman hocam Doç. Dr. Alper BAŞTÜRK'e sonsuz saygı ve teşekkürlerimi sunarım.

Optimizasyon konusundaki başarı ve bilgisiyle bizlere örnek olan Prof. Dr. Derviş KARABOĞA hocama, tez yazım ve uygulama araçlarında yardımcı olan Yrd. Doç. Dr. Fehim KÖYLÜ, Yrd. Doç. Dr. Emre BENDEŞ ve Arş. Gör. Ömür ŞAHİN hocalarıma, desteklerinden dolayı araştırma görevlisi dostlarıma, hocalarıma çok teşekkür ederim.

Hayatını çocuklarına adayan, her türlü zorluklara rağmen bu günlere gelmemi sağlayan, hakkını asla ödeyemeyeceğim fedakar babam Mustan ÖZCAN'a, hep yanımda hissettiğim ve hissedeceğim annem Emine ÖZCAN'a, abim Nadir ÖZCAN'a, babaannem Alime ÖZCAN'a ve amcalarım Sami ÖZCAN ve Sadettin ÖZCAN'a sonsuz teşekkür ederim.

Her türlü desteğiyle yanımda olan, kıymetlim, sevgili eşim İclal ÖZCAN'a ve biricik kızım Emine Defne ÖZCAN'a sonsuz teşekkür ederim.

KALMAN SÜZGEÇİ PARAMETRELERİNİN ENİYİLENMESİNDE ZEKİ OPTİMİZASYON TEKNİKLERİNİN BAŞARIMLARININ İNCELENMESİ

Tayyip ÖZCAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Temmuz 2016

Danışman : Doç. Dr. Alper BAŞTÜRK

ÖZET

Rudolf Kalman tarafından geliştirilen Kalman Süzgeçleme (KS) gelecekteki değerlerin kestirimi için modelin geçmişteki verilerinden yola çıkarak bir sonraki iterasyonda girdi olarak çıktıları kullanması ve hata kovaryansını her iterasyonda azaltması prensibine göre çalışmaktadır. Gürültülü durumlarda süzgeçleme işlemini hem güncelleme hem tahmin adımlarıyla gerçekleştiren KS, sistemde ölçülemeyen durumları ise sadece tahmin adımını uygulayarak kestirebilmektedir. KS'nin daha doğru sonuçlar verebilmesi için parametrelerinin de uygun seçilmesi önemli bir faktördür. KS parametrelerinden ölçüm kovaryans matrisi (R), işlem kovaryans matrisi (Q) ve başlangıç hata kovaryansı (P) değerlerinin en uygun şekilde seçilmesi gerekmektedir. Klasik yaklaşımda, bu seçim kullanıcı deneyimine bırakılırken, literatür incelendiğinde bu seçimin optimizasyon algoritmalarıyla da yapılabildiği görülmüştür. Bu tez çalışmasında, Kalman Süzgeçleme parametrelerinin eniyilenmesi için henüz literatürde çalışılmamış zeki optimizasyon tekniklerinin tespit edilmesi ve hem bu tekniklerin başarımlarının hem de tekniklerin birbiriyle işlem süresi ve doğruluk kriterleri açısından kıyaslanması amaçlanmıştır. Gerçekleştirilen detaylı benzetim çalışmaları ile farklı senaryolar için farklı algoritmaların başarılı olduğu gözlemlenmiştir.

Anahtar Kelimeler: Kalman Filtreleme (KF), Eniyileme, Parametre Ayarı, Zeki Optimizasyon Algoritmaları.

ANALYZING THE SUCCESS OF INTELLIGENT OPTIMIZING TECHNIQUES ON OPTIMIZING KALMAN FILTER PARAMETERS

Tayyip ÖZCAN

Erciyes University, Graduate School of Natural and Applied Sciences

M.Sc. Thesis, July 2016

Supervisor: Assoc. Prof. Dr. Alper BAŞTÜRK

ABSTRACT

Kalman Filter (KF), developed by Rudolf Kalman, works on the principle of using previous outputs as input for next iteration and reducing error covariance in each iteration to predict future values. KF uses both update and prediction steps for noisy situations, whereas it uses only prediction step to estimate the unmeasurable status of the system. Selecting the KF parameters is an important factor to get more accurate results. Parameters of KF (measurement covariance matrix (R), process covariance matrix (Q), and initial error covariance (P)) should be optimally selected. In conventional approach, determination of these parameters is based on user experience but when the literature analyzed, R , Q and P parameters are also selected by intelligent optimization algorithms. This thesis proposes detecting unused intelligent optimization algorithms in literature for optimizing KF parameters and comparing elapsed time and accuracy values of them. It is observed that different algorithms give different performance values for different scenarios.

Keywords: Kalman Filtering (KF), Optimization, Parameter Tuning, ,Intelligent Optimization Algorithms.

İÇİNDEKİLER

KALMAN SÜZGEÇİ PARAMETRELERİNİN ENİYİLENMESİNDE ZEKİ OPTİMİZASYON TEKNİKLERİNİN BAŞARIMLARININ İNCELENMESİ

BİLİMSEL ETİĞE UYGUNLUK SAYFASI	i
YÖNERGEYE UYGUNLUK SAYFASI	ii
KABUL VE ONAY	iii
ÖNSÖZ / TEŞEKKÜR	iv
ÖZET	v
ABSTRACT	vi
İÇİNDEKİLER	vii
TABLolar LİSTESİ	xi
ŞEKİLLER LİSTESİ	xiv

GİRİŞ	1
-----------------	---

1. BÖLÜM KALMAN SÜZGEÇLEME

1.1. Kalman Süzgeçleme	8
1.2. Kalman Süzgeçleme Temel Denklemleri	9
1.3. Kalman Süzgeçleme Parametreleri	10
1.3.1. Girdi Değerleri	10
1.3.2. Çıktı Değerleri	10
1.3.3. Ara Değerler	10
1.3.4. Sabit Değerler	10
1.4. Kalman Süzgeçleme Temel Adımları	11

2. BÖLÜM PROBLEM ÇÖZÜMÜNDE KULLANILACAK OLAN GÜNCEL ZEKİ OPTİMİZASYON TEKNİKLERİ

2.1. Genetik Algoritma	14
----------------------------------	----

2.1.1. Genetik Algoritma (GA)	14
2.1.2. Genetik Algoritma Temel Adımları	14
2.1.3. Genetik Algoritma Kontrol Parametreleri	17
2.2. Yapay Arı Koloni Algoritması	18
2.2.1. Yapay Arı Koloni Algoritması (YAKA)	18
2.2.2. Yapay Arı Koloni Algoritması Temel Adımları	19
2.2.3. Yapay Arı Koloni Algoritması Kontrol Parametreleri	19
2.3. Diferansiyel Gelişim Algoritması	21
2.3.1. Diferansiyel Gelişim Algoritması (DGA)	21
2.3.2. Diferansiyel Gelişim Algoritması Temel Adımları	22
2.3.3. Diferansiyel Gelişim Algoritması Kontrol Parametreleri	22
2.4. Parçacık Sürü Optimizasyon Algoritması	24
2.4.1. Parçacık Sürü Optimizasyon Algoritması (PSOA)	24
2.4.2. Parçacık Sürü Optimizasyon Algoritması Temel Adımları	25
2.4.3. Parçacık Sürü Optimizasyon Algoritması Kontrol Parametreleri	25
2.5. Ateşböceği Algoritması	27
2.5.1. Ateşböceği Algoritması (ABA)	27
2.5.2. Ateşböceği Algoritması Temel Adımları	27
2.5.3. Ateşböceği Algoritması Kontrol Parametreleri	28

3. BÖLÜM

DENEYSEL ÇALIŞMALAR VE SONUÇLAR

3.1. Uygulama 1	31
3.1.1. Problem Tanımı	31
3.1.2. Problemin Kalman Süzgeçleme Modeline Uyarlanması	32
3.1.3. Uygulama 1 İçin Uygunluk Fonksiyonu Seçimi	33
3.1.4. Algoritma Temel Parametrelerinin Seçimi	33
3.1.4.1. GA Temel Parametrelerinin Seçimi	33
3.1.4.2. YAKA Temel Parametrelerinin Seçimi	34
3.1.4.3. DGA Temel Parametrelerinin Seçimi	34

3.1.4.4.	PSOA Temel Parametrelerinin Seçimi	35
3.1.4.5.	ABA Temel Parametrelerinin Seçimi	35
3.1.5.	Optimize Edilmiş KS'nin Uygulama Sonuçları	35
3.2.	Uygulama 2	45
3.2.1.	Problem Tanımı	45
3.2.2.	Problemın Kalman Süzgeçleme Modeline Uyarlanması	45
3.2.3.	Uygulama 2 İçin Uygunluk Fonksiyonu Seçimi	47
3.2.4.	Algoritma Temel Parametrelerinin Seçimi	48
3.2.4.1.	GA Temel Parametrelerinin Seçimi	48
3.2.4.2.	YAKA Temel Parametrelerinin Seçimi	48
3.2.4.3.	DGA Temel Parametrelerinin Seçimi	49
3.2.4.4.	PSOA Temel Parametrelerinin Seçimi	49
3.2.4.5.	ABA Temel Parametrelerinin Seçimi	49
3.2.5.	Optimize Edilmiş KS'nin Uygulama Sonuçları	50
3.3.	Uygulama 3	59
3.3.1.	Problem Tanımı	59
3.3.1.1.	Uygulama 3.1 İçin Problem Tanımı	59
3.3.1.2.	Uygulama 3.2 İçin Problem Tanımı	60
3.3.1.3.	Uygulama 3.3 İçin Problem Tanımı	60
3.3.2.	Uygulama 3 İçin Uygunluk Fonksiyonu Seçimi	60
3.3.3.	Algoritma Temel Parametrelerinin Seçimi	61
3.3.3.1.	GA Temel Parametrelerinin Seçimi	62
3.3.3.2.	YAKA Temel Parametrelerinin Seçimi	62
3.3.3.3.	DGA Temel Parametrelerinin Seçimi	62
3.3.3.4.	PSOA Temel Parametrelerinin Seçimi	63
3.3.3.5.	ABA Temel Parametrelerinin Seçimi	63
3.3.4.	Optimize Edilmiş KS'nin Uygulama Sonuçları	63
3.3.4.1.	Uygulama 3.1 Sonucu	64

3.3.4.2.	Uygulama 3.2 Sonucu	73
3.3.4.3.	Uygulama 3.3 Sonucu	82

4. BÖLÜM

TARTIŞMA, SONUÇ ve ÖNERİLER

4.1.	Tartışma, Sonuç ve Öneriler	91
KAYNAKLAR		93
ÖZGEÇMİŞ		97

TABLOLAR LİSTESİ

Tablo 1.	Kalman Süzgeçleme vs PSOA tabanlı Kalman Süzgeçleme hata karşılaştırılması	2
Tablo 2.	Standart KKS vs PSOA tabanlı KKS parametrelerinin karşılaştırılması	3
Tablo 3.	VR = 1M ve V=17MRAD değerleri için algoritmaların RMSE değerlerinin karşılaştırılması	3
Tablo 4.	VR = 0.5M ve V=17MRAD değerleri için algoritmaların RMSE değerlerinin karşılaştırılması	4
Tablo 5.	VR = 0.1M ve V=17MRAD değerleri için algoritmaların RMSE değerlerinin karşılaştırılması	4
Tablo 6.	Standart ve DGA tabanlı KKS'nin MSE ve işlem süresi açısından karşılaştırılması	4
Tablo 7.	Kalman Süzgeçleme Parametrelerinin GA ile Eniyilenmesiyle Elde Edilen Sonuçlar	5
Tablo 8.	Kullanıcı tabanlı GKS'nin parametre değerlerine göre MSE performansı	6
Tablo 9.	PSOA tabanlı GKF'nin farklı iterasyonlardaki parametre değerlerine göre MSE performansı	6
Tablo 10.	GA tabanlı GKF'nin farklı iterasyonlardaki parametre değerlerine göre MSE performansı	6
Tablo 1.1.	Kalman Süzgeçlemenin kullanılabileceği bazı tahmin problemleri . .	9
Tablo 3.1.	GA Temel Parametreleri ve Değerleri	33
Tablo 3.2.	YAKA Temel Parametreleri ve Değerleri	34
Tablo 3.3.	DGA Temel Parametreleri ve Değerleri	34
Tablo 3.4.	PSOA Temel Parametreleri ve Değerleri	35
Tablo 3.5.	ABA Temel Parametreleri ve Değerleri	35

Tablo 3.6.	Uygulama 1 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafından seçilen KS parametreleri	36
Tablo 3.7.	Kullanılan algoritmaların uygulama 1 için hata dağılımı	36
Tablo 3.8.	Uygulama 1 için tüm algoritmalarda Kruskal Wallis test sonucu . . .	37
Tablo 3.9.	Uygulama 1 için ikili algoritmalarda Mann-Whitney U test sonucu .	37
Tablo 3.10.	GA Temel Parametreleri ve Değerleri	48
Tablo 3.11.	YAKA Temel Parametreleri ve Değerleri	48
Tablo 3.12.	DGA Temel Parametreleri ve Değerleri	49
Tablo 3.13.	PSOA Temel Parametreleri ve Değerleri	49
Tablo 3.14.	ABA Temel Parametreleri ve Değerleri	49
Tablo 3.15.	Uygulama 2 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafında seçilen KS parametreleri	50
Tablo 3.16.	Kullanılan algoritmaların uygulama 2 için hata dağılımı	51
Tablo 3.17.	Uygulama 2 için tüm algoritmalarda Kruskal Wallis test sonucu . . .	51
Tablo 3.18.	Uygulama 2 için ikili algoritmalarda Mann-Whitney U test sonucu .	51
Tablo 3.19.	GA Temel Parametreleri ve Değerleri	62
Tablo 3.20.	YAKA Temel Parametreleri ve Değerleri	62
Tablo 3.21.	DGA Temel Parametreleri ve Değerleri	62
Tablo 3.22.	PSOA Temel Parametreleri ve Değerleri	63
Tablo 3.23.	ABA Temel Parametreleri ve Değerleri	63
Tablo 3.24.	Uygulama 3.1 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafında seçilen KS parametreleri	64
Tablo 3.25.	Kullanılan algoritmaların uygulama 3.1 için hata dağılımları	65
Tablo 3.26.	Uygulama 3.1 için tüm algoritmalarda Kruskal Wallis test sonucu . .	65
Tablo 3.27.	Uygulama 3.1 için ikili algoritmalarda Mann-Whitney U test sonucu	65

Tablo 3.28.	Uygulama 3.2 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafında seçilen KS parametreleri	73
Tablo 3.29.	Kullanılan algoritmaların uygulama 3.2 için hata dağılımı	73
Tablo 3.30.	Uygulama 3.2 için tüm algoritmalarda Kruskal Wallis test sonucu . .	74
Tablo 3.31.	Uygulama 3.2 için ikili algoritmalarda Mann-Whitney U test sonucu	74
Tablo 3.32.	Uygulama 3.3 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafından seçilen KS parametreleri	82
Tablo 3.33.	Kullanılan algoritmaların uygulama 3.3 için hata dağılımı	83
Tablo 3.34.	Uygulama 3.3 için tüm algoritmalarda Kruskal Wallis test sonucu . .	83
Tablo 3.35.	Uygulama 3.3 için ikili algoritmalarda Mann-Whitney U test sonucu	83

ŞEKİLLER LİSTESİ

Şekil 1.1.	Kalman Süzgecinin Şematize Edilmiş Hali	13
Şekil 2.1.	GA Akış Şeması	17
Şekil 2.2.	YAKA Akış Şeması	20
Şekil 2.3.	DGA Akış Şeması	23
Şekil 2.4.	PSOA Akış Şeması	26
Şekil 2.5.	ABA Akış Şeması	29
Şekil 3.1.	Uygulama 1 için yöntemlerin iterasyon-MSE değişimi	37
Şekil 3.2.	Uygulama 1, kullanıcı deneyimi tabanlı KS (D1)	38
Şekil 3.3.	Uygulama 1, kullanıcı deneyimi tabanlı KS (D2)	38
Şekil 3.4.	Uygulama 1, kullanıcı deneyimi tabanlı KS (D3)	39
Şekil 3.5.	Uygulama 1, GA tabanlı KS için 30 koşmanın en iyisi	40
Şekil 3.6.	Uygulama 1, GA tabanlı KS için 30 koşmanın en kötüsü	40
Şekil 3.7.	Uygulama 1, YAKA tabanlı KS için 30 koşmanın en iyisi	41
Şekil 3.8.	Uygulama 1, YAKA tabanlı KS için 30 koşmanın en kötüsü	41
Şekil 3.9.	Uygulama 1, DGA tabanlı KS için 30 koşmanın en iyisi	42
Şekil 3.10.	Uygulama 1, DGA tabanlı KS için 30 koşmanın en kötüsü	42
Şekil 3.11.	Uygulama 1, PSOA tabanlı KS için 30 koşmanın en iyisi	43
Şekil 3.12.	Uygulama 1, PSOA tabanlı KS için 30 koşmanın en kötüsü	43
Şekil 3.13.	Uygulama 1, ABA tabanlı KS için 30 koşmanın en iyisi	44
Şekil 3.14.	Uygulama 1, ABA tabanlı KS için 30 koşmanın en kötüsü	44
Şekil 3.15.	Uygulama 2 için yöntemlerin iterasyon-ortalama norm(P) değişimi	52
Şekil 3.16.	Uygulama 2, kullanıcı deneyimi tabanlı KS (D1)	52

Şekil 3.17.	Uygulama 2, kullanıcı deneyimi tabanlı KS (D2)	53
Şekil 3.18.	Uygulama 2, kullanıcı deneyimi tabanlı KS (D3)	53
Şekil 3.19.	Uygulama 2, GA tabanlı KS için 30 koşmanın en iyisi	54
Şekil 3.20.	Uygulama 2, GA tabanlı KS için 30 koşmanın en kötüsü	54
Şekil 3.21.	Uygulama 2, YAKA tabanlı KS için 30 koşmanın en iyisi	55
Şekil 3.22.	Uygulama 2, YAKA tabanlı KS için 30 koşmanın en kötüsü	55
Şekil 3.23.	Uygulama 2, DGA tabanlı KS için 30 koşmanın en iyisi	56
Şekil 3.24.	Uygulama 2, DGA tabanlı KS için 30 koşmanın en kötüsü	56
Şekil 3.25.	Uygulama 2, PSOA tabanlı KS için 30 koşmanın en iyisi	57
Şekil 3.26.	Uygulama 2, PSOA tabanlı KS için 30 koşmanın en kötüsü	57
Şekil 3.27.	Uygulama 2, ABA tabanlı KS için 30 koşmanın en iyisi	58
Şekil 3.28.	Uygulama 2, ABA tabanlı KS için 30 koşmanın en kötüsü	58
Şekil 3.29.	Uygulama 3.1, tünel konumu 43-57. çerçeveler	59
Şekil 3.30.	Uygulama 3.2, tünel konumu 67-81. çerçeveler	60
Şekil 3.31.	Uygulama 3.3, tünel konumu 86-100. çerçeveler	61
Şekil 3.32.	Uygulama 3.1 için yöntemlerin iterasyon-MSE değişimi	66
Şekil 3.33.	Uygulama 3.1, kullanıcı deneyimi tabanlı KS (D1)	66
Şekil 3.34.	Uygulama 3.1, kullanıcı deneyimi tabanlı KS (D2)	67
Şekil 3.35.	Uygulama 3.1, kullanıcı deneyimi tabanlı KS (D3)	67
Şekil 3.36.	Uygulama 3.1, GA tabanlı KS için 30 koşmanın en iyisi	68
Şekil 3.37.	Uygulama 3.1, GA tabanlı KS için 30 koşmanın en kötüsü	68
Şekil 3.38.	Uygulama 3.1, YAKA tabanlı KS için 30 koşmanın en iyisi	69
Şekil 3.39.	Uygulama 3.1, YAKA tabanlı KS için 30 koşmanın en kötüsü	69
Şekil 3.40.	Uygulama 3.1, DGA tabanlı KS için 30 koşmanın en iyisi	70

Şekil 3.41.	Uygulama 3.1, DGA tabanlı KS için 30 koşmanın en kötüsü	70
Şekil 3.42.	Uygulama 3.1, PSOA tabanlı KS için 30 koşmanın en iyisi	71
Şekil 3.43.	Uygulama 3.1, PSOA tabanlı KS için 30 koşmanın en kötüsü	71
Şekil 3.44.	Uygulama 3.1, ABA tabanlı KS için 30 koşmanın en iyisi	72
Şekil 3.45.	Uygulama 3.1, ABA tabanlı KS için 30 koşmanın en kötüsü	72
Şekil 3.46.	Uygulama 3.2 için yöntemlerin iterasyon-MSE değişimi	75
Şekil 3.47.	Uygulama 3.2, kullanıcı deneyimi tabanlı KS (D1)	75
Şekil 3.48.	Uygulama 3.2, kullanıcı deneyimi tabanlı KS (D2)	76
Şekil 3.49.	Uygulama 3.2, kullanıcı deneyimi tabanlı KS (D3)	76
Şekil 3.50.	Uygulama 3.2, GA tabanlı KS için 30 koşmanın en iyisi	77
Şekil 3.51.	Uygulama 3.2, GA tabanlı KS için 30 koşmanın en kötüsü	77
Şekil 3.52.	Uygulama 3.2, YAKA tabanlı KS için 30 koşmanın en iyisi	78
Şekil 3.53.	Uygulama 3.2, YAKA tabanlı KS için 30 koşmanın en kötüsü	78
Şekil 3.54.	Uygulama 3.2, DGA tabanlı KS için 30 koşmanın en iyisi	79
Şekil 3.55.	Uygulama 3.2, DGA tabanlı KS için 30 koşmanın en kötüsü	79
Şekil 3.56.	Uygulama 3.2, PSOA tabanlı KS için 30 koşmanın en iyisi	80
Şekil 3.57.	Uygulama 3.2, PSOA tabanlı KS için 30 koşmanın en kötüsü	80
Şekil 3.58.	Uygulama 3.2, ABA tabanlı KS için 30 koşmanın en iyisi	81
Şekil 3.59.	Uygulama 3.2, ABA tabanlı KS için 30 koşmanın en kötüsü	81
Şekil 3.60.	Uygulama 3.3 için yöntemlerin iterasyon-MSE değişimi	84
Şekil 3.61.	Uygulama 3.3, kullanıcı deneyimi tabanlı KS (D1)	84
Şekil 3.62.	Uygulama 3.3, kullanıcı deneyimi tabanlı KS (D2)	85
Şekil 3.63.	Uygulama 3.3, kullanıcı deneyimi tabanlı KS (D3)	85
Şekil 3.64.	Uygulama 3.3, GA tabanlı KS için 30 koşmanın en iyisi	86

Şekil 3.65.	Uygulama 3.3, GA tabanlı KS için 30 koşmanın en kötüsü	86
Şekil 3.66.	Uygulama 3.3, YAKA tabanlı KS için 30 koşmanın en iyisi	87
Şekil 3.67.	Uygulama 3.3, YAKA tabanlı KS için 30 koşmanın en kötüsü	87
Şekil 3.68.	Uygulama 3.3, DGA tabanlı KS için 30 koşmanın en iyisi	88
Şekil 3.69.	Uygulama 3.3, DGA tabanlı KS için 30 koşmanın en kötüsü	88
Şekil 3.70.	Uygulama 3.3, PSOA tabanlı KS için 30 koşmanın en iyisi	89
Şekil 3.71.	Uygulama 3.3, PSOA tabanlı KS için 30 koşmanın en kötüsü	89
Şekil 3.72.	Uygulama 3.3, ABA tabanlı KS için 30 koşmanın en iyisi	90
Şekil 3.73.	Uygulama 3.3, ABA tabanlı KS için 30 koşmanın en kötüsü	90

GİRİŞ

Amerikan sistem teoristi Rudolf Kalman tarafından geliştirilen Kalman Süzgeci (KS), 1950'li yıllarda bulunmasına rağmen çalışma prensibinin geliştirilmesiyle 1960'lı yıllarda aktif olarak kullanılmaya başlanmıştır. Etkili bir tahmin edici olan bu algoritma, ilk olarak araç navigasyon sistemlerinde ve ardından uzay programlarında kullanılarak yaygınlaşmıştır. KS gürültülü durumlarda süzgeçleme yaparken hem tahmin hem güncelleme adımlarını uygular. Buna karşılık sistemde ölçülemeyen bir durumla karşılaşıldığı zaman, sadece tahmin adımını uygulayarak kestirim yapabilmektedir. [1].

KS girdi, çıktı, ara ve sabit değerler olmak üzere 4 farklı grupta toplayabileceğimiz parametrelerden oluşmaktadır. Bu parametrelerden özellikle, başlangıç hata kovaryansı P , işlem hata kovaryansı Q ve ölçüm hata kovaryansı R parametrelerinin seçimi, sistemin daha doğru çalışması bakımından önemlidir. Bu parametrelerin seçimi kullanıcı deneyimine bırakıldığında optimum sonuca ulaşmak oldukça güçtür. Bu zor süreç güncel zeki optimizasyon algoritmaları kullanılarak kolaylaştırılabilir. Optimizasyon algoritması tabanlı KS ile Q , R ve P parametrelerinin eniyilenmesi sağlanmıştır.

Oldukça geniş bir kullanım alanına sahip olan KS, biyomedikal, bilgisayar bilimleri, ekonomi vb alanlarda yaygınca kullanılmaktadır. Bu tez çalışmasında KS parametreleri sinyalden gürültü arındırma ve popüler bir konu olan nesne takibi uygulamaları için kullanılmış ve zeki optimizasyon algoritmalarıyla desteklenen KS ile daha doğru sonuçlar elde edilmiştir.

Yapılan nesne takibi uygulamasında ayrıca KS'nin çok iyi bir kestirim aracı olduğu görülmüştür. Gelecek kestirimi yapılmasına ihtiyaç duyulan durumlarda KS'nin kullanılması gayet uygundur. Daha önceden topladığı veriler vasıtasıyla kestirim yapabilen KS'nin daha doğru sonuçlar verebilmesi için de her senaryonun uygun bir optimizasyon algoritmasıyla desteklenmesi gerekmektedir.

Literatür Özeti

Mevcut çalışmalar incelendiğinde, Q, R ve P KS parametrelerinin zeki optimizasyon algoritmalarıyla eniyilenebildikleri tespit edilmiştir.

Parametrelerin Genetik Algoritma (GA) kullanarak belirlenmesi üzerine bir çalışma yapan Jianguo Yan ve arkadaşları, başlangıç durumundaki belirsizlik varyansı, ölçüm gürültüsü varyansı, tahmin hatası kovaryansı gibi terimlerin bulunduğu objektif fonksiyonlarını minimize etme yoluyla KS parametrelerinin iyileştirilmesi üzerine başarı sağlamışlardır [2]. Dikey jiroskop, üç eksenli jiroskop oranı, başlık sensörü, dinamik ve statik basınç sensörlerinden oluşan insansız hava aracı navigasyon sistemi kaçınılmaz bir şekilde gürültüye maruz kalmaktadır. Çalışmada gürültü GA ile optimize edilen KS kullanılmıştır.

Nesne takibi, nesnelerin rastgele gerçekleştirdiği hareketin rotasını yakalamayı amaçlamaktadır. Füze takibi, güvenlik sistemleri, sanal gerçeklik arayüzleri gibi ticari alanlar araştırmacıların fazlaca dikkatini çeken savunma uygulamalarından bazılarıdır. KS, varsayılan başlangıç durumu ve gürültü kovaryanslarıyla birlikte nesnenin takibini sağlamaktadır. Nimmakayala Ramakoti ve arkadaşları daha etkili takip yapılabilmesi için KS gürültü kovaryansları parametrelerinin optimize edilmesi gerektiğini saptamıştır [3]. Yaptıkları çalışmada gürültü kovaryanslarının ayarlanmasında Parçacık Sürüşü Optimizasyon algoritmasını (PSOA) kullanmışlardır. Tablo 1’de Standart KS ile PSOA tabanlı KS’nin hata bazında farkı görülmektedir.

Tablo 1. Kalman Süzgeçleme vs PSOA tabanlı Kalman Süzgeçleme hata karşılaştırılması

Algoritma	Parametre Ayarlama Süresi	Hata
Klasik Kalman Süzgeçleme	-	0.3051
PSOA’lı Kalman Süzgeçleme	0.646 sn	0.06

KS, genellikle hedef takibi uygulamalarında kullanılan adaptif bir süzgeçleme algoritmasıdır. Model ve ölçümlerin doğrusal olmadığı durumlarda Genişletilmiş Kalman Süzgeçleme (GKS) veya Kokusuz Kalman Süzgeçleme (KKS) kullanılmaktadır. Hedef durumunun tahmininin daha iyi yapılabilmesi için KS algoritmasının parametre ayarının doğru bir şekilde yapılması gerekmektedir. Ravi Kumar ve arkadaşları hedef takibinde daha doğru sonuçlar elde edebilmek için PSOA tabanlı KKS algoritmasını kullanmış, bu

yöntemin standart KKS'ye göre daha başarılı olduğunu belirlemişlerdir [4]. Tablo 2'de çalışma ile elde edilen parametre değerleri verilmiştir.

Tablo 2. Standart KKS vs PSOA tabanlı KKS parametrelerinin karşılaştırılması

Parametre	Standart KKS	PSOA Tabanlı KKS
Ölçüm Gürültü Standart Sapması	0.05	0.0002
İşlem Hata Kovaryansı q_x	0.01	0.0119
İşlem Hata Kovaryansı q_y	0.01	0.0339
RMSE (m)	0.104	0.033
Hesaplama Karmaşıklığı (sn)	0.00156	0.424

Ravi Kumar Jatoth ve Gogulamudi Anudeep Reddy, PSOA ile hibritleştirilen GA tabanlı KKS kullanarak, güç sistemi gerilim/akım işaretlerinin mevcut harmonik bileşenlerinin tahminini gerçekleştirmişlerdir [5]. Başlangıç işlem ve ölçüm hata kovaryanslarının seçimi, gürültünün arındırılmasında hayati rol oynamaktadır. Bundan dolayı, hibrit GA-PSOA yaklaşımı, KKS RMSE hatasının minimize edilmesini sağlamak amacıyla hata kovaryans matrislerinin tahmini için kullanılmıştır. Simülasyon sonuçları hibrit GA-PSOA tabanlı KKS'nin standart KKS'ye göre daha doğru sonuçlar verdiğini göstermektedir [5].

Balistik bir hedefin takibinde, doğrusal olmayan süzgeçleme için radar ölçümleri hayli karmaşık bir problemdir. Ölçüm değerleri gürültülü olan hedef konumunun tahmininde KS kullanılmaktadır. Ölçüm değerlerinin doğrusal olmaması da KKS'nin kullanılmasını gerektirmektedir. Daha doğru sonuçların elde edilmesi için GA ve PSOA tabanlı KKS kullanan Ravi Kumar ve arkadaşları, yaptıkları simülasyon sonucunda standart KKS'ye karşı daha doğru sonuçlar aldığını belirtmişlerdir [6]. Elde edilen sonuçlar Tablo 3-5'te verilmektedir.

Tablo 3. VR = 1M ve V=17MRAD değerleri için algoritmaların RMSE değerlerinin karşılaştırılması

Algoritma	RMSE (m)	
	X Düzlemi	Y Düzlemi
Standart UKF	6.87	3.213
GA	6.859	3.21
PSOA	6.778	3.202

Tablo 4. VR = 0.5M ve V=17MRAD değerleri için algoritmaların RMSE değerlerinin karşılaştırılması

Algoritma	RMSE (m)	
	X Düzlemi	Y Düzlemi
Standart UKF	6.864	3.153
GA	6.832	3.141
PSOA	6.724	3.128

Tablo 5. VR = 0.1M ve V=17MRAD değerleri için algoritmaların RMSE değerlerinin karşılaştırılması

Algoritma	RMSE (m)	
	X Düzlemi	Y Düzlemi
Standart UKF	6.863	3.126
GA	6.843	3.124
PSOA	6.820	3.110

Kokusuz dönüşüm için ölçekleme parametre seçimi çalışması yapan Jin Yao [7], genel metodun analizi ve karşılaştırılmasıyla ölçekleme parametresinin seçimini bir optimizasyon hedefi olarak görmüştür. KKS'nin çevrimiçi ve çevrimdışı modeline Diferansiyel Gelişim algoritmasını (DGA) uygulamış ve çalışmanın sonuçları Tablo 6'da verilmiştir. Bu çalışmada, DGA tabanlı KKS'nin daha doğru sonuçlar verdiği tespit edilmiştir.

Tablo 6. Standart ve DGA tabanlı KKS'nin MSE ve işlem süresi açısından karşılaştırılması

	Standart KKS		DGA Tabanlı KKS	
	Uygulama 1	Uygulama 2	Çevrimdışı Model	Çevrimiçi Model
MSE	46.1613	44.4866	7.2151	6.3411
Zaman[sn]	0.0026	0.0026	1.0699	1.0234

T.O. Ting ve arkadaşlarının çalışmasında pil sistemindeki şarj durumunun tahmini için KS kullanılmıştır. RMS hata hesaplamasıyla birlikte en küçük hata hesaplaması yapılarak en iyi sonuç elde edilmeye çalışılmıştır. Ölçüm hata kovaryansı (R) ve işlem hata kovaryansına (Q) farklı değerler verilerek RMS değeri gözlemlenebilir. Bu durum tamamiyle kullanıcının deneyimine veya deneme yanılma yoluna bağlıdır. Bu çalışmada GA kullanılarak KS parametrelerinden Q ve R değerleri eniyilenmiştir [8]. Çalışma

sonucunda KS parametreleri (Q, R) eniyilenmiş ve RMS değerinde düşüş gözlenmiştir. Sonuçlar Tablo 7’de verilmiştir Bu tablodaki y değeri doğru değeri, y_v değeri ölçülen değeri, y_e süzgeçlenen değeri ifade etmektedir..

Tablo 7. Kalman Süzgeçleme Parametrelerinin GA ile Eniyilenmesiyle Elde Edilen Sonuçlar

Çıkış	Q,R	RMS Hata Değeri
$y - y_v$	1, 1	1.0013
Tahmin Edilen(KF), $y - y_e$	1, 1	0.00019185
Tahmin Edilen(GA tabanlı KF), $y - y_e$	0.126973, 2.3039	0.0000023282

Bir diğer çalışmada ise Guohan Lin ve arkadaşları, sensörsüz bir indüksiyon motorunun kontrolünde GA’nın avantajlarıyla hibritlenen bir algoritma olan Gelişmiş Parçacık Sürü Optimizasyon algoritmasını (GPSOA) GKS eniyilenmesinde kullanarak uygulamada büyük bir sorun olarak gördükleri işlem ve ölçüm kovaryans matrislerinin seçiminde çözümü bulmak için çalışmışlardır [9]. Simülasyon sonuçları GPSOA, hız tahmininde GA, PSO ve DGA’ya göre daha doğru sonuçlar verdiğini göstermektedir.

Bir indüksiyon motorunun hızı ve rotor akımını tahmin etmek için GKS ve PSO’nun birleştirildiği yeni bir yöntem Yahia Laamari ve Kheireddine Chafaa tarafından geliştirilmiştir [10]. Kovaryans matrislerinin çevrimdışı bir sistemde PSO ile eniyilenmesi ve daha sonra eniyilenmiş değerlerin modele eklenmesi bu çalışmanın temel adımlarını oluşturmaktadır. Tablo 8’de kullanıcı deneyimine bırakılan parametre seçimine göre alınan MSE değerleri verilmiştir. Tablo 9’da PSO tabanlı olan GKS’nin MSE performansının kullanıcı deneyimli olandan çok daha iyi olduğu gözlemlenmektedir. Bu çalışmada ayrıca GA tabanlı GKS de PSO tabanlı GKS ile aynı koşullarda çalıştırılmış ve PSO tabanlı GKS’nin daha başarılı olduğu tespit edilmiştir. GA tabanlı GKS çalışmasına ait sonuç verileri Tablo 10’da yer almaktadır [10].

Bu tez dönemi içerisinde, Kalman Süzgeçleme parametrelerinin zeki optimizasyon algoritmalarıyla eniyilendiği 2 bildirimimiz, ICETAS2016 [11] ve IC-SMHD2016 [12] konferanslarında kabul edilmiş ve sunulmuştur.

Tablo 8. Kullanıcı tabanlı GKS'nin parametre değerlerine göre MSE performansı

Durum	Q ve R Değerleri	MSE	Tahmin Kalitesi
1	$q_i = q\psi = q\Omega = r_i = 0.1$	2376.1	Çok kötü
2	$q_i = q\psi = 1e-3, q\Omega = r_i = 1e-2$	136.8057	Çok kötü
3	$q_i = q\psi = q\Omega = 0.01, r_i = 0.1$	72.2788	Kötü
4	$q_i = q\psi = 1e-7, q\Omega = 1e-2, r_i = 1e-1$	33.1022	Kötü
5	$q_i = q\psi = 1e-6, q\Omega = r_i = 1e-2$	19.1381	Kötü
6	$q_i = q\psi = 1e-4, q\Omega = 1e-2, r_i = 1e-1$	2.2877	İyi
7	$q_i = q\psi = 1e-4, q\Omega = 1e-3, r_i = 1e-2$	1.0220	İyi
8	$q_i = q\psi = 1e-3, q\Omega = 1e-3, r_i = 1e-1$	0.3866	Çok iyi

Tablo 9. PSOA tabanlı GKF'nin farklı iterasyonlardaki parametre değerlerine göre MSE performansı

İterasyon Sayısı	Q ve R Değerleri	MSE
5	$r_i = 0.1818; q_i = 0.0838; q\psi = 0.0038; q\Omega = 0.0027$	0.7406
10	$r_i = 0.205; q_i = 0.3424; q\psi = 0.0054; q\Omega = 0.0117$	0.3897
20	$r_i = 1.1597; q_i = 0.2512; q\psi = 0.0152; q\Omega = 0.0279$	0.2619
30	$r_i = 15.2927; q_i = 0.4156; q\psi = 0.2518; q\Omega = 0.0823$	0.0848

Tablo 10. GA tabanlı GKF'nin farklı iterasyonlardaki parametre değerlerine göre MSE performansı

İterasyon Sayısı	Q ve R Değerleri	MSE
5	$r_i = 0.929; q_i = 0.947; q\psi = 0.024; q\Omega = 0.07$	1.1620
10	$r_i = 0.95; q_i = 0.906; q\psi = 0.033; q\Omega = 0.035$	0.7176
20	$r_i = 0.707; q_i = 1.435; q\psi = 0.021; q\Omega = 0.028$	0.3105
30	$r_i = 15.8244; q_i = 3.2488; q\psi = 0.3083; q\Omega = 0.0341$	0.1487

Tezin Amacı

Bu çalışmasının amaçları şu şekilde sıralanabilir:

- KS parametrelerini farklı uygulamalarda, farklı optimizasyon algoritmalarıyla belirleyip optimum sonuçlara ulaşmak.
- Ölçülemeyen durumlar için, KS yardımıyla kestirim yapabilmek.
- Tünel vb engeller ardında kalan nesnelerin takibinin ve pozisyon kestirimlerinin yapılabilmesi.

Tezin Organizasyonu

KS parametrelerinin güncel zeki optimizasyon algoritmalarıyla eniyilenmesini amaçlayan bu tez şu şekilde organize edilmiştir:

- Birinci bölümde, geniş bir kullanım alanına sahip olan KS hakkında bilgi verilmiştir.
- İkinci bölümde, KS parametrelerinin eniyilenmesi için kullanılacak olan güncel zeki optimizasyon algoritmaları anlatılmıştır.
- Üçüncü bölümde, 3 farklı uygulamada KS parametrelerinin eniyilenmesi çalışması yapıp sonuçlar değerlendirilmiştir.
- Dördüncü bölümde, tez çalışmasından çıkan sonuçlar, gelecekte yapılması planlanan çalışmalar belirtilmiştir.

1. BÖLÜM

KALMAN SÜZGEÇLEME

KS, modelin geçmişteki verilerinden yararlanarak izleyen iterasyonlarda girdi olarak çıktıları kullanma ve hata kovaryansını her iterasyonda azaltma prensibine göre çalışmaktadır [1].

Algoritma, gürültülü verilerde özyinelemeli bir biçimde gerçek zamanlı çalışarak hataları en küçük kareler yöntemi kullanarak azaltır ve sistem modeli ile üretilen gelecek durumun kestiriminde iyileştirme yapar [13].

Bu bölümde ilk olarak KS hakkında bilgi verilecek, daha sonra KS'nin parametreleri ele alınacak ve son olarak KS'nin temel adımları detaylı bir şekilde incelenecektir.

1.1. Kalman Süzgeçleme

Macar asıllı Amerikan matematiksel sistem teoristi Rudolf Kalman tarafından geliştirilen KS, modelin geçmişteki verilerden yararlanarak izleyen iterasyonlarda girdi olarak çıktıları kullanma ve hata kovaryansını her iterasyonda azaltma prensibine göre çalışmaktadır [1]. KS, etkili bir tahmin edici ve sistemin ölçülemeyen durumlarını kestirebilen yetenekli bir algoritmadır. Bu algoritma, ölçülebilen durumların gürültüden arındırılması işlemini gerçekleştirebilirken, ayrıca ölçülemeyen durumlarda da önceki verileri kullanmak gibi dolaylı yollardan ölçüm sağlayabilmektedir. Dolayısıyla bu güçlü ve etkili algoritma, geçmiş verilerden yola çıkarak gelecekteki muhtemel durumlar için de kestirim yapabilmektedir [1] [14].

Optimizasyon algoritmalarıyla desteklenmeye elverişli olan KS, 20. yüzyıldaki önemli buluşlardan birisidir. Bu algoritma aslında bir süzgeçlemeden daha çok bir tahmin edici olarak kabul edilebilmektedir.

Radyo ve radar sinyalleri tahmini, görüntü işleme, sismik veri analizi, hava durumu

tahmini, tarımsal ürünler için uygun hasat zamanı tahmini, navigasyon sistemleri, nesne takibi, deprem kestirimi, ekonomi vb birçok alanda kullanılan KS'nin kullanım alanı oldukça geniştir. Tablo 1.1'de KS'nin kullanıldığı bazı uygulama alanları ve algılayıcı tipleri sunulmuştur [15].

Tablo 1.1. Kalman Süzgeçlemenin kullanılabileceği bazı tahmin problemleri

Uygulama	Uygulama Alanı	Algılayıcı Tipi
İşlem Kontrol	Kimya Tesisi	Basınç Sıcaklık Akış hızı Gaz analizi
Su Baskını Kestirimi	Nehir	Su seviyesi Yağmur göstergesi Hava durumu radarı
Takip	Uzay aracı	Radar Görüntü sistemi
Navigasyon	Gemi	Gemi hız ölçer Jiroskop İvme ölçer GPS Sekstant

1.2. Kalman Süzgeçleme Temel Denklemleri

Kalman Süzgeçlemenin temel iki denklemi aşağıdaki gibidir [16]:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_{k-1} \quad (1.1)$$

$$\mathbf{Z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (1.2)$$

Eşitlik (1.1) ile hesaplanan $\mathbf{x}_k$ değeri, önceki durum değerine, kontrol vektörü $\mathbf{u}_k$ ile önceki işlem gürültüsü $\mathbf{w}_{k-1}$ değerlerinin eklenmesiyle bulunur. $\mathbf{u}_k$ değeri genellikle sistemde kullanılmaz. Diğer bir temel denklem olan Eşitlik (1.2)'de; ölçüm değeri $\mathbf{Z}_k$, gözlem vektörü ile durum tahmininin çarpımı ve ölçüm gürültüsü değerlerinin toplamına eşittir.

1.3. Kalman Süzgeçleme Parametreleri

KS temel denklemlerinde ve temel adımlarında kullanılan parametrelerin bir kısmı ($\mathbf{u}_k$, $\mathbf{Z}_k$) girdi olarak, bir kısmı ($\hat{\mathbf{x}}_k$, $\mathbf{P}_k$) çıktı olarak, bir kısmı ($\hat{\mathbf{x}}_k^-$, $\mathbf{P}_k^-$, $\mathbf{K}_k$, $\mathbf{v}_k$, $\mathbf{w}_{k-1}$) ara değerler olarak, bir kısmı da ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{R}$) sabit değerler olarak kullanılmaktadır [1].

1.3.1. Girdi Değerleri

$\mathbf{u}_k$: Kontrol vektörüdür. Modelde, kontrol sisteminin veya kullanıcı kontrolünün büyüklüğünü ifade eder. $u \times 1$ boyutunda bir matristir.

$\mathbf{Z}_k$: Ölçüm vektörüdür. Her bir zaman adımında ölçülen gerçek değerdir. $m \times 1$ boyutunda bir matristir.

1.3.2. Çıktı Değerleri

$\hat{\mathbf{x}}_k$: Durum vektörüdür. Hesaplanan yeni tahmindir. $n \times 1$ boyutunda bir matristir.

$\mathbf{P}_k$: Güncel hata kovaryansını verir. $n \times n$ boyutunda bir matristir.

1.3.3. Ara Değerler

$\hat{\mathbf{x}}_k^-$: Hesaplanan önceki tahmindir. $n \times 1$ boyutunda bir matristir.

$\mathbf{P}_k^-$: Hesaplanan önceki hata kovaryansıdır. $n \times n$ boyutunda bir matristir.

$\mathbf{K}_k$: Kalman kazancıdır. Her bir iterasyonda tekrar hesaplanır. $n \times m$ boyutunda bir matristir.

$\mathbf{v}_k$: Ölçüm gürültüsüdür. $m \times 1$ boyutunda bir matristir.

$\mathbf{w}_{k-1}$: Önceki işlem gürültüsüdür. $q \times 1$ boyutunda bir matristir.

1.3.4. Sabit Değerler

$\mathbf{A}$: Durum geçiş matrisidir. $n \times n$ boyutunda bir matristir.

$\mathbf{B}$: Giriş vektörü, kontrol matrisidir. Her kontrol sinyali için etki derecesini gösterir. $n \times u$ boyutunda bir matristir.

H : Gözlem vektörü, çıkış matrisidir. $m \times n$ boyutunda bir matristir.

Q : Tahmini işlem hatası kovaryansıdır. $q \times q$ boyutunda bir matristir.

R : Tahmini ölçüm hatası kovaryansıdır. $m \times m$ boyutunda bir matristir.

KS parametrelerinden başlangıç hata kovaryansı **P**, işlem hata kovaryansı **Q** ve ölçüm hata kovaryansı **R** değerlerinin doğru seçilmesi, daha iyi tahmin sonuçlarının elde edilmesine olanak sağlamaktadır. Bu parametrelerin seçimi, geçmişteki çalışmalarda kullanıcı deneyimine bırakılmışken, son yıllardaki çalışmalarda bu parametreler için değer seçme işlemi güncel zeki optimizasyon algoritmalarıyla yapılabilmektedir [1].

1.4. Kalman Süzgeçleme Temel Adımları

KS'nin iki temel adımı aşağıdaki gibidir [17]:

1. Zaman Güncelleme (Kestirim):

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k \quad (1.3)$$

Eşitlik (1.3)'de $\hat{\mathbf{x}}_k^-$, hesaplanan tahmini değeri ifade etmektedir. Bu değer ayrıca ölçüm güncelleme adımında da kullanılacak olan tahmini durum değeridir. Sistemin durum geçiş matrisini ifade eden **A** ile kontrol matrisini ifade eden **B** değerleri, var ise sistem modellemesine dahil edilmelidir. $\hat{\mathbf{x}}_{k-1}$ başlangıç durum değerinin uygun bir değer seçilmesi bir sonraki adım için daha doğru sonuçlar vermesi açısından önem arz etmektedir. Bu değer hiçbir şekilde bilinmemesi durumunda ise 0 değerini alması uygundur. Bu bilgilerden yola çıkarak hesaplanan tahmini durum değeri; geçiş matrisi ile önceki durum değeri çarpımı ve giriş matrisi ile kontrol vektörünün çarpım değerlerinin toplamına eşittir.

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (1.4)$$

Eşitlik (1.4)'de $\mathbf{P}_k^-$, hesaplanan tahmin hata kovaryansını ifade etmektedir. Bu değer; durum geçiş matrisi ve bu matrisin transpozu ile önceki hata kovaryansın çarpımı ve işlem hata kovaryansı değerlerinin toplamına eşittir. Eğer başlangıç hata kovaryansı bilinmiyorsa başlangıç hata kovaryansı olarak 1 alınması uygundur.

2. Ölçüm Güncelleme (Doğrulama):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (1.5)$$

Eşitlik (1.5)'te yer alan Kalman Kazancı $\mathbf{K}_k$, durum tahmininin bir çarpanıdır ve durum tahmininin değişim oranını ifade etmektedir. Bu parametreye en fazla etki eden parametre $\mathbf{P}_k^-$ dır. $\mathbf{P}_k^-$ değerinin büyük veya küçük olması aynı şekilde Kalman Kazancının da büyük veya küçük olmasına sebep olmaktadır. Kalman Kazancını etkileyen bir diğer parametre de ölçüm hata kovaryansı $\mathbf{R}$ değeridir. Kalman Kazancı $\mathbf{R}$ ile ters orantılı olarak değer almaktadır.

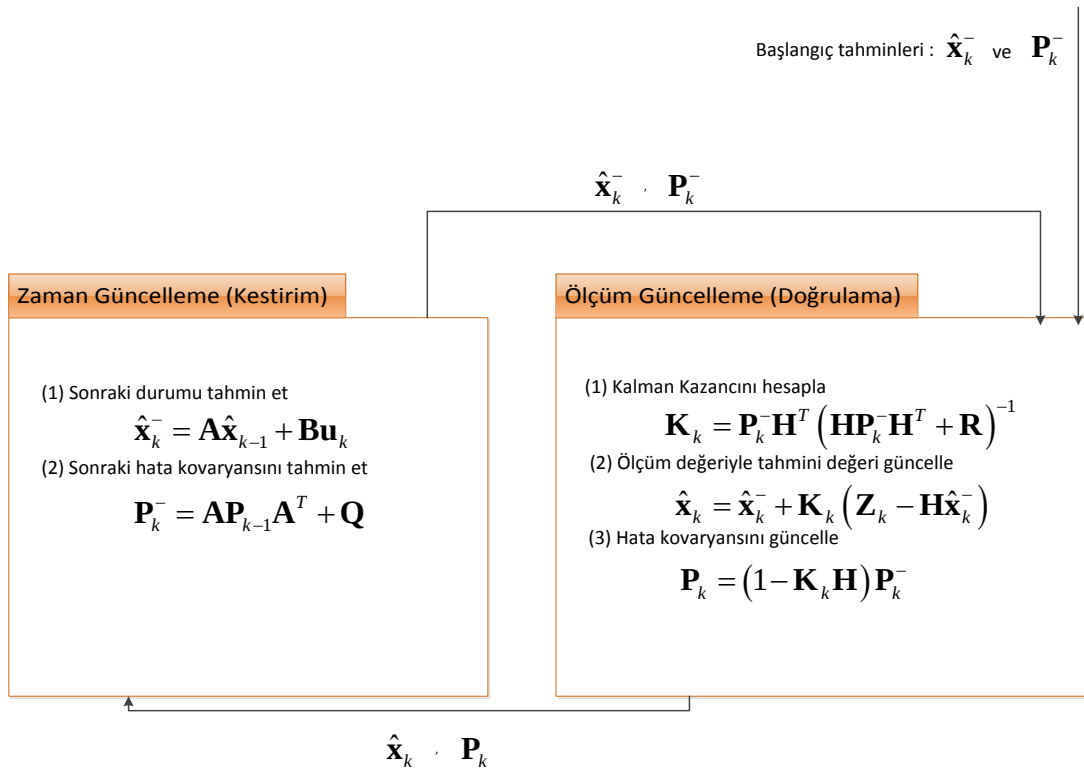
$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{Z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \quad (1.6)$$

Eşitlik 1.6, durum tahmininin yapıldığı kısımdır. Bu değer zaman güncelleme adımından gelen, hesaplanan tahmini durum değeriyle Eşitlik (1.5)'te elde edilen Kalman Kazancı değerinin işleme girmesiyle elde edilir. Bulunan bu değer bir sonraki iterasyonda zaman güncelleme adımı girdi olarak kullanılacaktır.

$$\mathbf{P}_k = (1 - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \quad (1.7)$$

Kalman Kazancı ve hesaplanan önceki hata kovaryansı bilgilerinin işlenmesiyle elde edilen güncel hata kovaryansı Eşitlik (1.7) ile ifade edilmektedir. Bu değer de durum tahmini değeri gibi bir sonraki iterasyonun zaman güncelleme adımı girdi olarak kullanılacaktır.

KS'nin şematize edilmiş bir diğer gösterimi Şekil 1.1'deki gibidir. Bu şemadan, başlangıç durum ve hata kovaryansı değerinin sisteme girdi olarak verildiği görülmektedir. Zaman güncelleme adımı hesaplanan önceki hata kovaryansı ve durum tahmini değerleri ölçüm güncelleme adımı girdi olarak gönderilirken, ölçüm güncelleme adımı hesaplanan güncel hata kovaryansı ve durum değeri bir sonraki iterasyonda zaman güncelleme adımı girdi olarak gönderilmektedir [18].



Şekil 1.1. Kalman Süzgecinin Şematize Edilmiş Hali

2. BÖLÜM

PROBLEM ÇÖZÜMÜNDE KULLANILACAK OLAN GÜNCEL ZEKİ OPTİMİZASYON TEKNİKLERİ

KS parametrelerinin doğru seçimi, daha doğru sonuçlar elde edilmesine olanak sağlamaktadır. Bu parametrelerin seçiminin eniyilenmesi için kullanılacak olan güncel zeki optimizasyon teknikleri bu bölümde incelenecektir.

2.1. Genetik Algoritma

2.1.1. Genetik Algoritma (GA)

1975'te J.Holland ve arkadaşları tarafından geliştirilen GA, biyolojik evrimin doğal süreçlerinden esinlenerek oluşturulmuş çözüm algoritmalarıdır [19]. Genel çözümün kromozomlar şeklinde gösterimini takiben, farklı genetik operatörlerle yapılan işlemler sonucunda, en iyi uygunluk değerini veren gen dizilişine ulaşılması hedeflenmektedir. Kullanıcı tarafından belirtilen durdurma kriteri koşulu sağlanıncaya kadar, algoritma çözüm alternatiflerini denemeye ve en uygununu aramaya devam etmektedir.

Evrimsel algoritmanın yarışma yardımlaşma ve değişim olan 3 temel özelliğini taşımaktadır. GA'daki yarışma seleksiyon adımına, yardımlaşma çaprazlama adımına, değişim ise mutasyon adımına karşılık gelmektedir.

2.1.2. Genetik Algoritma Temel Adımları

GA'ların ilk aşamasında, incelenilen problemin parametreleri uygun şekilde kodlanarak kromozomlar oluşturulur. Daha sonra bu kromozomlardan oluşan bir başlangıç popülasyonu meydana getirilir. Popülasyonda yer alan her bir kromozom, problemin

çözümü için bir adaydır. Ardından popülasyondan bazı kromozomlar seçilir ve doğadaki genetik süreçlere dayanan “çaprazlama” ve “mutasyon” genetik operatörleri uygulanarak daha iyi kromozomlar elde edilmeye çalışılır. Ardı ardına gerçekleştirilen iterasyonlar ile yeni nesiller oluşturulur ve bu yeni nesillerin verdiği uygunluk değerleri popülasyondakilerle karşılaştırılır [19].

```

Başlangıç popülasyonunu oluştur;
Popülasyondaki her bir çözümün uygunluk değerini hesapla;
if Durdurma kriteri sağlanmışsa then
    | Araştırmayı sonlandır;
else
    | Seleksiyon;
    | Çaprazlama;
    | Mutasyon;
end
Adım 2'ye git;

```

Algoritma 2.1: Genetik Algoritma Temel Adımları

Algoritma 2.1’de temel adımları verilen GA’nın adım adım incelenmesi:

Başlangıç popülasyonunu oluştur

Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur (çözüm grubu, biyolojideki benzerliği nedeniyle popülasyon, çözümlerin kodları (string) da kromozom olarak adlandırılır). Popülasyon büyüklüğü olması gerekenden küçük seçilirse çözüm uzayının yeterince araştırılmamasına sebep olurken olması gerekenden büyük seçilmesi durumunda da çözüme ulaşmak uzun zaman almaktadır.

Uygunluk değeri hesaplama

Uygunluk fonksiyonu yardımı ile her kromozomun çözüm kalitesinin ne kadar yüksek olduğu bulunur.

Seleksiyon

Popülasyonda, kaliteli çözümlerin artması, kalitesiz çözümlerin azalmasını sağlayan operatördür. Seçimin yapılabilmesi için kullanılan en temel yöntemler :

- **Rulet Tekeri:**

Popülasyondaki tüm bireylerin uygunluk değerleri toplanır. Burada her bir

kromozomun seçilme şansı, kendi uygunluk değerinin tüm bireylerin uygunluk değerlerinin toplamına oranı kadardır.

- Sıralama:

Popülasyondaki bireyler, uygunluk değerlerine göre küçükten büyüğe sıralanır. Bu sıraya göre her bir bireye sıra numarası verilir ve bu numaralara rulet tekeri yöntemi uygulanır. Bu sayede bütün kromozomlara seçilme şansı verilir.

- Turnuva:

Popülasyondaki kromozomlardan belirli bir sayıda birey rastgele olarak seçilir ve bunlardan uygunluk değeri en iyi olan kromozom ilk ebeveyn olarak alınır. Benzer şekilde bu işlem tekrar uygulanarak ikinci ebeveyn seçilir.

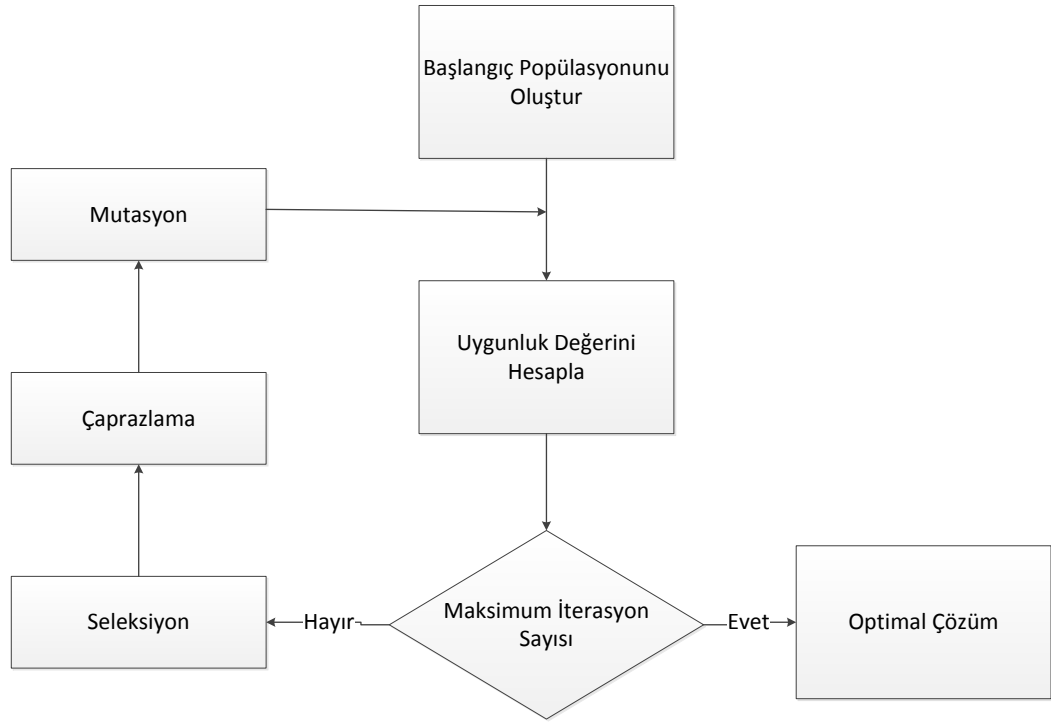
Çaprazlama

Yüksek kaliteye sahip bireyler çaprazlama yapılarak daha yüksek kaliteli bireylerin üretilmesi hedeflenmektedir. Düşük çaprazlama oranı yakınsama hızını azaltırken yüksek çaprazlama oranı çözüm uzayının hızlı araştırılmasını dolayısıyla kaliteli çözümün çabuk bozulmasına neden olarak algoritmanın performansını düşürür. En önemli adımlardan birisi olan çaprazlamada, bir önceki adım olan seleksiyon aşamasında seçim süreci sonunda oluşan ebeveynler çaprazlama yapılarak yeni bireyler oluşturulur. Burada amaç; ebeveyn kromozom genlerinden uygunluk değeri daha yüksek olan çocuk kromozomlar meydana getirmeye çalışmaktır. En çok kullanılan çaprazlama yöntemi ise tek noktalı çaprazlamadır.

Mutasyon

Seçme ve çaprazlama işlemleriyle birbirine benzer çözümler elde edilir. Çaprazlama sonucu oluşan çocuğun kromozomunda genlerden bir ya da daha fazlası yer değiştirilerek genetik çeşitliliği sağlamak ve devam ettirmek için kromozom elde edilir. Mutasyon olasılığı genellikle düşük tutulur ve mutasyon bütün çocuklara uygulanmaz, bu yüzden mutasyon etkileri kromozomlarda az görülür ve kromozomların sayısı değişmez.

GA'nın akış şeması Şekil 2.1'de verilmiştir.



Şekil 2.1. GA Akış Şeması

2.1.3. Genetik Algoritma Kontrol Parametreleri

GA'nın performansının değişken olmasındaki en büyük etkenlerden birisi de kontrol parametrelerinin seçimidir. Basit GA'nın başlıca temel parametreleri: popülasyon büyüklüğü, iterasyon sayısı, çaprazlama oranı ve mutasyon oranıdır.

Popülasyon Büyüklüğü

Popülasyonu oluşturan her birey için hesaplanan uygunluk değeri GA'nın performansı açısından önemlidir. Bu değerın aşırı küçük seçilmesi, iterasyonların daha hızlı tamamlanmasına ve yerel maksimuma takılma riskinin artmasına sebebiyet verecektir. Ayrıca araştırma uzayının yeteri kadar incelenememesine sebep olacaktır. Popülasyon büyüklüğü değerinin aşırı büyük olması da her bir iterasyonun işlem süresini artıracak ve özellikle çevrimiçi uygulamalar için istenmeyen bir durumun ortaya çıkmasına sebep olacaktır [19].

İterasyon Sayısı

Maksimum iterasyon sayısı GA için bir durdurma kriteridir. Popülasyon büyüklüğü

gibi iterasyon sayısının olması gerekenden çok büyük seçilmesi çevrimiçi uygulamalar için istenmeyen bir durumdur. Aşırı küçük seçilmesi de optimum sonuca ulaşmayı engelleyebilir.

Çaprazlama Oranı

Çaprazlama oranı, aynı araştırma uzayında farklı kısımları da tarayabilme yeteneğini sağlayan yardımlaşma operatörüdür. Bu değerin çok düşük seçilmesi, yeni nesle aktarılan yeni yapının küçük oranda girmesine sebep olur ve bundan dolayı araştırmanın yakınsama hızı düşer. Çaprazlama oranının, olması gerekenden çok yüksek seçilmesi araştırma uzayının çok hızlı araştırılmasına neden olur [20]. Bu parametre genellikle 0.6-0.9 arasında bir değer alır.

Mutasyon Oranı

Değişim operatörü olan mutasyon işlemi belli oranda gerçekleştirilir. Bu oranın çok yüksek seçilmesi kromozomun yapısının bozulmasına dolayısıyla da faydalı genlerin kaybolmasına sebep olacaktır. Oranın olması gerekenden düşük seçilmesi de araştırma uzayının tamamen araştırılmasına engel olacak ve ıraksamayı düşürecektir.

2.2. Yapay Arı Koloni Algoritması

2.2.1. Yapay Arı Koloni Algoritması (YAKA)

YAKA, Derviş Karaboğa tarafından bal arılarının doğal yiyecek arama davranışlarından esinlenerek ortaya çıkarılmıştır. YAKA, arılar arasındaki görev paylaşımı ve kendi kendilerine organize olabildikleri için sürü zekâsına dayalı, kontrol parametresi az olan basit ve esnek bir algoritmadır [20]. Kolonide var olan her arı kendi görevini gerçekleştirmektedir. YAKA için kolonide üç farklı türde arı vardır. Bunlar: kaşif (scout) arı, görevli (employed) arı ve gözcü (onlooker) arıdır. Kaşif arılar küresel araştırma kabiliyetine sahipken, gözcü ve görevli arılar bölgesel araştırma kabiliyetine sahiptir. Görevli ve gözcü arıların sayısı eşittir ve her bir görevli arı ayrı bir yiyecek kaynağından sorumludur.

Yiyecek kaynağı yeri olası çözümleri ifade ederken, bu kaynaklara ait nektar miktarı

kaynakların kalite değerini ifade etmektedir. Geniş bir kullanım alanına sahip olan YAKA genellikle GA, PSO ve Evrimsel Strateji algoritmalarına göre daha başarılıdır. Hem numerik hem ayrık problemler için kullanılabilen YAKA, oldukça sade ve esnektir.

2.2.2. Yapay Arı Koloni Algoritması Temel Adımları

YAKA'da ilk olarak yiyecek kaynakları ve bu yiyecek kaynaklarına uygun yerlerin üretilmesi sağlanır. Yiyecek kaynaklarının üretiminden sonra görevli arılar kendi sorumluluklarındaki yiyecek kaynaklarına gönderilir. Yeni kaynaklar, işçi arılar tarafından yiyecek kaynaklarının komşuluğunda belirlenir. Yeni bulunan yiyecek kaynağının kalitesi önceki kaynaktan daha iyiye yeni yiyecek kaynağı hafızaya alınır. Aksi halde, önceki kaynağın deneme sayacı bir artırılarak araştırmaya devam edilir. Araştırma işleminin bitmesiyle görevli arılar kaynaklarla ilgili pozisyon ve kalite bilgisini gözcü arılara aktarır. Gözcü arılar bu bilgileri değerlendirir ve kaynakların nektar miktarına bağlı olarak olasılık değerlerini hesaplar. Şayet bir i . kaynak için üretilen olasılık değeri p_i 0 ve 1 arasında rastgele üretilen sayıdan küçükse, o kaynak gözcü arı tarafından incelemeye alınır. Kalitesi yüksek olan yiyecek kaynakları daha fazla seçilme eğilimindedir. İyi değerlerin seçimi rastgeleliğe bağlı olduğu için her zaman iyi kaynakların seçilmesi garanti değildir.

YAKA'ya ait temel adımlar Algoritma 2.2 ile gösterilebilir.

YAKA'nın akış şeması ise Şekil 2.2 ile gösterilebilir.

```

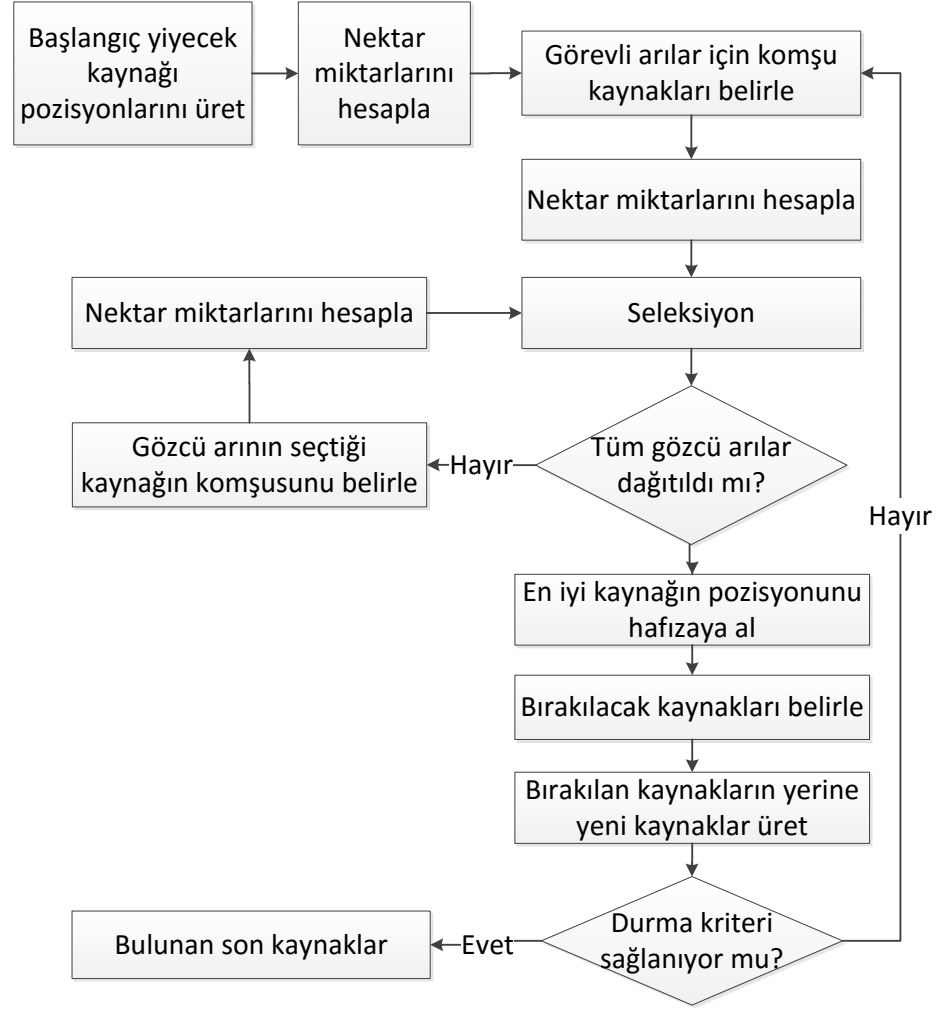
Başlangıç yiyecek kaynağı bölgelerinin oluşturulması;
while Çevrim sayısı <= Max İterasyon Sayısı do
    Görevli arıların yiyecek kaynağı bölgelerine gönderilmesi;
    Olasılıksal seleksiyonda kullanılacak olasılık değerlerinin görevli arılardan
    gelen bilgiye göre hesaplanması;
    Gözcü arıların olasılık değerlerine göre yiyecek kaynağı bölgesi seçmeleri;
    Bırakılacak kaynakların bırakılışı ve kâşif arı üretimi;
end

```

Algoritma 2.2: Yapay Arı Koloni Algoritması Temel Adımları

2.2.3. Yapay Arı Koloni Algoritması Kontrol Parametreleri

YAKA'nın temel kontrol parametreleri koloni büyüklüğü, parametre sayısı, yiyecek



Şekil 2.2. YAKA Akış Şeması

kaynağı, maximum döngü sayısı ve limittir.

Koloni Büyüklüğü

Görevli ve gözücü arıların toplam sayısından oluşmaktadır. Optimum sonuca optimum zamanda ulaşmak için bu parametrenin uygun seçimi önemlidir.

Yiyecek Kaynağı Sayısı

Koloni büyüklüğünün yarı değerine eşit olan yiyecek kaynağı sayısı, optimum sonuca ulaşma açısından önemli bir parametredir.

Maximum Döngü Sayısı

Bir durdurma kriteri olan bu değer, uygunluk değerinin artmayı veya azalmayı bıraktığı iterasyon olarak seçilebilir. Bu değerın olması gerekenden büyük seçilmesi işlem süresi açısından maliyetli ve istenmeyen bir durumdur. Bu değerın çok küçük seçilmesi ise optimum değere ulaşamamaya sebep olabilir.

Limit

Görevli arıların yiyecek kaynağını terk etmesi için belirlenen deneme sayısıdır. Deneme sayacı 'limit' değerinin üzerine çıkarsa görevli kaynağı terk eder ve kaşif arı haline gelerek yeni kaynak arayışına girer [20].

2.3. Diferansiyel Gelişim Algoritması

2.3.1. Diferansiyel Gelişim Algoritması (DGA)

Price ve Storn tarafından geliştirilen DGA, operatörlerin benzerliği açısından genetik algoritmaya dayanan, popülasyon tabanlı sezgisel bir optimizasyon tekniğidir [21]. Temel olarak GA'ya dayanan bu algoritmada bir çözüm, adım adım ilerler ve bireyler birbirini beklemez. Dolayısıyla bu durum DGA'ya hız kazandırır.

GA'da var olan seçim, çaprazlama ve mutasyon adımları DGA'da da bulunmaktadır. GA'dan farklı olarak mutasyon işlemi diğer operatörlerden daha önce gelmektedir. GA'dan bir diğer farkı ise DGA'nın nümerik olmasıdır.

Daha iyi sonuçların elde edilmesi için DGA, bir veya daha fazla yöntemle birleştirilerek hibritleşme sağlanmıştır. Başta YAKA olmak üzere, PSO, ısıtım işlem algoritması gibi

farklı algoritmalarla faydalı kısımlar birleştirilerek yeni yöntemler sunulmuştur [22–26].

2.3.2. Diferansiyel Gelişim Algoritması Temel Adımları

DGA temel adımlarından ilki, kontrol parametrelerinin belirlenmesidir. Popülasyon büyüklüğü, iterasyon sayısı, alpha, betamin ve gammadan oluşan DGA kontrol parametrelerinin değerleri belirlenir. Daha sonra başlangıç popülasyonu oluşturulur. Kullanıcı tarafından belirlenen kromozom sayısı (NP), yeni kromozomların üretilmesi için kendi haricinde üç adet kromozoma daha ihtiyaç duyduğu için en az 4 olmalıdır [21]. Kromozom sayısı (NP) ile problem boyutu (D)’den başlangıç popülasyonu meydana getirilir. Bu aşamadan sonra maximum iterasyon sayısı sağlanana kadar sırasıyla mutasyon, çaprazlama ve seleksiyon adımları gerçekleştirilir. Mevcut kromozomun bir kısım genleri üzerinde bir kısım rastgele değişim yapılması ve bu sayede araştırma uzayında aranan optimum çözüm için kromozomda farklılık oluşturulması mutasyon adımıyla gerçekleştirilmektedir. Tamamlayıcı bir işlem olan çaprazlama adımının amacı, var olan genlerden faydalanılarak yeni genleri oluşturmak suretiyle problem çözümünde başarılı bir sonuca ulaşmakta yardımcı etken olmaktır. Seleksiyon adımı ile, mevcut nesil ve üretilen yeni nesil kromozomlar değerlendirilip karşılaştırılarak yeni neslin oluşması sağlanır [20]. Seçme işlemindeki faktör, kromozomların uygunluk değerlerine bağlıdır. DGA’nın algoritma ve akış şeması sırasıyla Algoritma 2.3 ve Şekil 2.3 ile gösterilebilir.

```

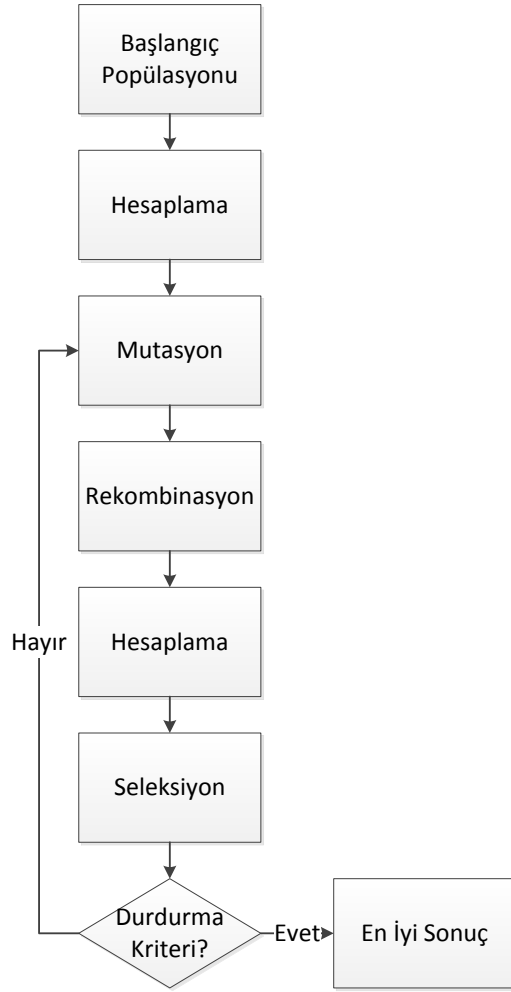
Kontrol parametrelerini belirle;
Başlangıç popülasyonunu oluştur;
if Durdurma kriteri sağlanıyorsa then
    | Araştırmayı durdur;
else
    | Mutasyon;
    | Rekombinasyon;
    | Değerlendirme;
    | Seleksiyon;
end

```

Algoritma 2.3: Diferansiyel Gelişim Algoritması Temel Adımları

2.3.3. Diferansiyel Gelişim Algoritması Kontrol Parametreleri

Popülasyon büyüklüğü, iterasyon sayısı, çaprazlama sabiti ve adım boyutu DGA’nın



Şekil 2.3. DGA Akış Şeması

temel kontrol parametreleridir.

Popülasyon Büyüklüğü

Problemin çözümü için oluşturulan kromozom sayısını ifade etmektedir. Her bir kromozom bir çözüme karşılık gelir. Bu değerin çok büyük seçilmesi, işlem süresini uzatırken, çok küçük bir değer seçilmesi de optimum sonuca ulaşamama riskini artırır.

İterasyon Sayısı

Durdurma kriteri olan bu parametrenin doğru sonuca ulaşmadaki etkisi fazladır. Bu parametrenin alacağı değerin çok büyük olması, işlem süresini uzatacağı için özellikle çevrimiçi uygulamalarda istenmeyen bir durum oluşturur. Seçilen değerin çok küçük

olması durumunda araştırma uzayı yeteri kadar incelenemez.

Çaprazlama Sabiti

Evrimsel süreçteki yardımlaşma olarak kabul edilir. Bu oran olması gerekenden küçük seçilirse, yeni nesle az sayıda yeni yapının girmesine neden olur. Oran olması gerekenden büyük seçilirse araştırma uzayı çok hızlı bir şekilde araştırılır ve bireyler aynı bilgiyi taşımaya başlar.

Adım Boyutu

DGA için bir ölçekleme faktörü olan bu değer 0 ile 2 arasında bir değer alabilir. Bu parametre, algoritmanın uygulanacağı probleme göre farklı değerlerde optimum sonucu verebilir.

2.4. Parçacık Sürü Optimizasyon Algoritması

2.4.1. Parçacık Sürü Optimizasyon Algoritması (PSOA)

Sürü zekası temelli algoritmalarından birisi olan PSOA Eberhart ve Kennedy tarafından 1995 yılında önerilmiştir. Balık ve kuş sürülerinin sosyal ve bireysel davranışlarından esinlenerek ortaya çıkan bu algoritmanın uygulanabilirliği gayet kolaydır. Biyoteknoloji ve bilgisayar bilimleri başta olmak üzere birçok alanda kullanımı gerçekleştirilmektedir [27]. PSOA’da her parçacık, uygulandığı problem için ayrı ayrı olası çözümleri ifade eder. Amaç fonksiyonu ile de her parçacığın performansı hesaplanır.

Araştırma uzayında akış halinde bulunan her bir parçacığın pozisyonu kendi komşuluğunda bulunan parçacıkların ve kendi pozisyon bilgilerine göre etkilenir.

PSOA, optimal çözümler elde etmede başarılı bir algoritmadır. PSOA’nın en büyük dezavantajlarından biri yakınsama problemidir. Parçacıklar arasında bilgi akışının hızlı olması benzer parçacıkların ortaya çıkmasına, dolayısıyla da lokal yakınsama riskinin artmasına sebep olmaktadır. Hibrit PSOA çalışmalarında bu problemin aşımına yönelik çalışmalar yapılmıştır [28–30].

2.4.2. Parçacık Sürü Optimizasyon Algoritması Temel Adımları

PSOA'da öncelikle başlangıç parçacıkları oluşturulur. Her bir parçacık olası bir çözümü ifade eder. Daha sonra durdurma kriteri sağlanana kadar döngü devam ettirilir. Bu döngüde, öncelikle her bir parçanın uygunluk değeri hesaplanır. Sürüdeki en iyi parçacıklar belirlenir ve bunlar içerisindeki en iyi parçacık seçilir. Parçacıkların her biri, parçacığın en iyi kendi çözümü p_{best} ve tüm parçacıkların en iyi çözümü g_{best} kullanılarak yeniden hesaplanır [27]. Algoritma 2.4'te temel adımları işlenen PSOA'nın

```

Başlangıç parçacıklarını oluştur;
if Durdurma kriteri sağlanıyorsa then
    Araştırmayı durdur;
else
    Her bir parçanın uygunluk değerini hesapla;
    Sürüdeki en iyi parçacıkları belirle;
    En iyi parçacığı seç;
    Her bir parçacığın hızını hesapla;
    Hesaplanmış hızı kullanarak parçacığın veri değerlerini güncelle;
end

```

Algoritma 2.4: Parçacık Sürü Optimizasyon Algoritması Temel Adımları

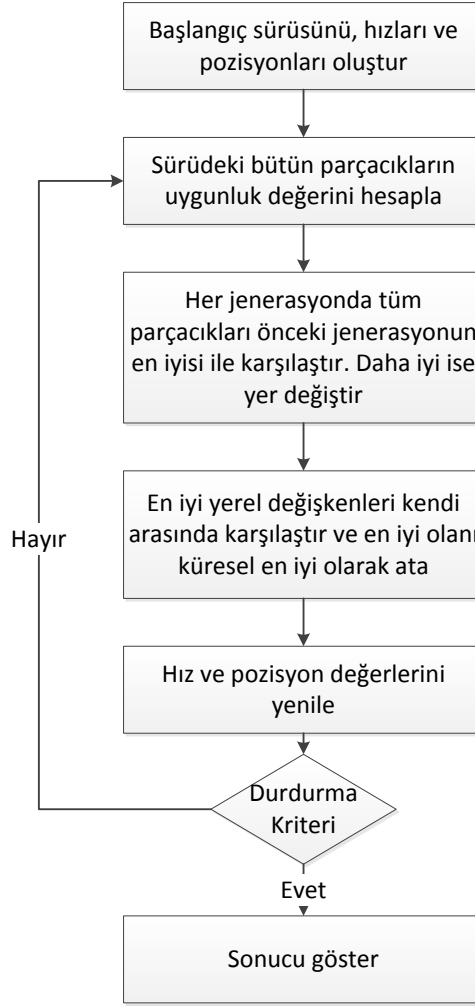
akış şeması Şekil 2.4'te verilmektedir.

2.4.3. Parçacık Sürü Optimizasyon Algoritması Kontrol Parametreleri

PSOA'nın temel kontrol parametreleri popülasyon büyüklüğü, iterasyon sayısı, parçacık boyutu, maximum değişiklik V_{max} , öğrenme faktörleri (C_1 , C_2) ve eylemsizlik ağırlığı (w)'dir.

Popülasyon Büyüklüğü

PSOA için popülasyon büyüklüğü değeri sürüdeki parçacık sayısına eşittir. Bu değer olması gereken değerden çok daha küçük olması araştırma uzayının yetersiz örneklenmesine sebep olur ve araştırma belli bir alt optimal noktaya doğru sürüklenir [20]. Değer olması gereken değerden çok büyük seçilmesi ise özellikle çevrimiçi uygulamalarda gecikmeye neden olacağı için istenmeyen bir durumun ortaya çıkmasına sebep olur.



Şekil 2.4. PSO Akış Şeması

İterasyon Sayısı

Diğer algoritmalarda olduğu gibi bu algorithmada da bir durdurma kriteri olan iterasyon sayısının seçimi optimum sonucu bulma ve işlem süresi açısından önemlidir.

Parçacık Boyutu

Optimize edilecek probleme göre değişmektedir.

$$V_{\max}$$

Bir iterasyondaki bir parçacıkta meydana gelebilecek maksimum değişikliği belirler. Genellikle parçacığın alabileceği alt ve üst sınırlara göre belirlenir.

Öğrenme Faktörleri

C_1 , C_2 değerleridir ve $[0, 4]$ aralığında belirlenir. Genellikle iki değer birbirine eşittir ve 2 olarak seçilir.

Eylemsizlik Ağırlığı

Global ve yerel arama yeteneğini dengelemek için kullanılmaktadır. Bu değer büyük olması global arama, küçük olması da yerel arama yapılmasını kolaylaştırır.

2.5. Ateşböceği Algoritması

2.5.1. Ateşböceği Algoritması (ABA)

ABA, 2007 yılında Xin She Yang tarafından geliştirilen metasezgisel bir optimizasyon algoritmasıdır. Bu algortmada, doğadaki ateş böceklerinin sosyal davranışlarından esinlenilmiştir [31]. Ateş böceklerinin tek cinsiyetli olması ve birbirlerini çekme özellikleri, ABA'nın temelini oluşturmaktadır. Ateş böceklerinin parlaklıkları ABA'da uygunluk değerine karşılık gelmektedir. Bir ateş böceğinin ne kadar parlak olduğu o ateş böceğinin çekiciliğini ifade eder ve ateş böcekleri kendilerinden daha parlak bir ateş böceği görürse, ona doğru yönlenecektir. Ateş böceğinin yanıp sönen ışığı, diğer ateş böceklerini çekmek için kullandığı bir sinyal sistemi gibidir [32]. Ateş böcekleri, bu ışıkları sayesinde, arkadaşlarını bulabilir, olası avlarını çekebilir ve avcılardan kendilerini koruyabilmektedirler [33].

2.5.2. Ateşböceği Algoritması Temel Adımları

Işık yoğunluğunun değişimi ve popülasyondaki ateş böceklerinin çekiciliği, bu algoritmanın iki önemli noktasıdır. Ateş böceklerinin çekiciliği, parlaklığı ile doğru orantılıdır ve ABA'nın amaç fonksiyonu değeri ile ilişkilidir. ABA'da bir optimizasyon probleminde parlaklığı I ile ifade edilir ve Eşitlik 2.1 ile hesaplanır.

$$I = I_0 e^{-\gamma r} \quad (2.1)$$

Bu eşitlikteki I_0 değeri başlangıç ışık yoğunluğunu, γ değeri ışık soğurma katsayısını, r

değeri iki ateş böceği arasındaki uzaklığı ifade etmektedir. Parlaklığa ve iki ateş böceği arasındaki mesafeye bağlı olan ateş böceği çekicilik değeri Eşitlik 2.2 ile hesaplanır.

$$\beta = \beta_0 e^{-\gamma r^2} \quad (2.2)$$

β ifadesindeki değere göre az çekici olan ateş böceği daha çekici olan ateş böceğine doğru hareket eder. Bu hareketle birlikte yeni çözüm Eşitlik 2.3 ile hesaplanır.

$$x_i^{t+1} = x_i + \beta_0 e^{(-\gamma r_{ij}^2)} (x_j - x_i) + \alpha \varepsilon_i^t \quad (2.3)$$

Bu eşitlikteki r_{ij} değeri iki ateş böceği arasındaki uzaklıktır ve Eşitlik 2.4 uzaklık formülüyle belirlenir [31].

$$r_{ij} = |x_i - x_j|^2 = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (2.4)$$

```

Başlangıç popülasyonunu oluştur;
if Durdurma kriteri sağlanıyorsa then
    | Araştırmayı durdur;
else
    | Herbir ateşböceğinin uygunluk değerini hesapla;
    | Işık şiddeti değerini güncelle;
    | Durumu güncelle;
    | Yeni çözümleri değerlendir;
end

```

Algoritma 2.5: Ateşböceği Algoritması Temel Adımları

Algoritma 2.5 ile temel adımları işlenen PSOA'nın akış şeması Şekil 2.5'te verilmiştir.

2.5.3. Ateşböceği Algoritması Kontrol Parametreleri

ABA'nın 5 adet temel kontrol parametresi vardır. Bunlar: popülasyon büyüklüğü, iterasyon sayısı, adım büyüklüğü (alpha), çekicilik değeri (betamin) ve çekicilik değişim parametresi (gamma)'dir.

Popülasyon Büyüklüğü

Popülasyondaki ateş böceği sayısını ifade eder. Her bir ateş böceği, olası bir çözüme karşılık gelmektedir.



Şekil 2.5. ABA Akış Şeması

İterasyon Sayısı

Optimum sonuca ulaşmada önemli bir parametre olan iterasyon sayısı, ayrıca bir durdurma kriteridir. Bu parametre, alması gereken değerin çok üstünde bir değer alırsa, işlem süresini uzatacağı için özellikle çevrimiçi uygulamalarda istenmeyen bir durum oluşturur. Bu parametre, alması gereken değerden çok küçük değer alması durumunda ise araştırma uzayı yeteri kadar araştırılmaz ve optimum sonuca ulaşamama riski doğurur.

Alpha

Adım büyüklüğü veya bir diğer adıyla ölçekleme faktörü olarak bilinir. Genellikle 0.5 değerini almaktadır.

Betamin

Ateş böceklerinin parlaklıklarına ve iki ateş böceği arasındaki mesafeye bağlı olan parametredir.

Gamma

Çekicilik değişim değeri olan bu parametre genellikle 1 değerini almaktadır.

3. BÖLÜM

DENEYSEL ÇALIŞMALAR VE SONUÇLAR

Bu bölümde, KS parametrelerinin eniyilenmesi için yapılan farklı uygulamalar incelenip, uygulamalar için zaman ve doğruluk kriterlerine göre en ideal zeki optimizasyon algoritması belirlenecektir. Uygulama 1’de sabit bir voltaj okuyucusundan alınan değerlere gürültü eklenecek ve sadece gürültülü değerler KS algoritmasına verilerek gürültüden arındırılması sağlanacaktır. Uygulama 2’de eğik atış uygulaması yapılacak ve atılan nesnenin x ve y koordinatları, KS yardımıyla gürültüden arındırılacak ve gerçek değere ulaşmaya çalışılacaktır. Son uygulama olan Uygulama 3’de ise nesne takibi işlemi gerçekleştirilecektir. Bu uygulamada diğer uygulamalardan farklı olarak nesnenin görünmediği durumlarda da KS’nin kestirim adımı kullanılarak nesnenin konumu tahmin edilecektir. Bu üç uygulamada, daha doğru sonuçlar elde etmek için uygun seçilmesi şart olan KS parametrelerinden Q , R , P parametreleri güncel zeki optimizasyon teknikleri kullanılarak eniyilenmeye çalışılacaktır. Doğruluk ve işlem süresi kriterleri yanısıra bu 3 uygulama için istatistiksel analiz yapılacaktır. Algoritmaların dağılım analizinde homojen dağılmadığı görülmüş olup bu nedenden dolayı algoritmaların analizi yapılırken Kruskal Wallis ve Mann-Whitney U nonparametrik testleri uygulanacaktır [34].

3.1. Uygulama 1

3.1.1. Problem Tanımı

Sabit bir kaynaktan 0.5 V voltaj değeri üretildiği varsayıp, 200 iterasyonluk bir çalışma yapılacaktır. Her ölçümde aslında var olan gerçek değer 0.5 V iken bu değerlere rastgele gürültüler eklenecek olup, bu gürültülü değerler KS’ye verilerek gerçek değere ne derece yaklaşılabileceği test edilecektir. Uygulamada ölçüm hata kovaryansı değeri de

değiştirilerek bu değerin sonuca etkisi de gözlemlenecektir.

3.1.2. Problemin Kalman Süzgeçleme Modeline Uyarlanması

KS'nin temel iki denklemi aşağıdaki gibidir:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_{k-1} \quad (3.1)$$

$$\mathbf{Z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (3.2)$$

Uygulamanın çözümü için öncelikle model oluşturulacaktır. 1 boyutlu bir sinyal problemi olduğundan dolayı modeldeki her bir eleman matris formatında olmayıp sayısal değere sahiptir. Herhangi bir kontrol sinyali ($\mathbf{u}_k$) bulunmadığı için bu değer 0'a eşittir. Sinyal sabit bir değere sahip olduğundan dolayı durum geçiş değeri ($\mathbf{A}$) 1 değerini alır. Ölçüm değeri; durum değeri ile bir miktar gürültünün birleşiminden oluşacağından gözlem değeri ($\mathbf{H}$) 1'e eşittir. Bu bilgilerden sonra (3.1) ve (3.2)'deki denklemlerin sadeleştirilmiş hali aşağıdaki gibidir:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (3.3)$$

$$\mathbf{Z}_k = \mathbf{x}_k + \mathbf{v}_k \quad (3.4)$$

Uygulamada (k) değeri iterasyon adımını belirtmektedir. $\mathbf{x}_0$ başlangıç ölçüm değeri 0 olarak ve $\mathbf{P}_0$ başlangıç hata kovaryansı daha iyi bir durum bilinmediğinden dolayı 1 olarak alınacaktır. $\mathbf{Q}$ tahmini işlem hatası yapılmayacağından dolayı 0 olarak alınacaktır.

Bu bilgilerden sonra KS algoritmasının 2 adımı aşağıdaki gibi oluşur:

1. Zaman Güncelleme (Kestirim):

$$\hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1} \quad (3.5)$$

$$\mathbf{P}_k^- = \mathbf{P}_{k-1} \quad (3.6)$$

2. Ölçüm Güncelleme (Doğrulama):

$$\mathbf{K}_k = \mathbf{P}_k^- (\mathbf{P}_k^- + \mathbf{R})^{-1} \quad (3.7)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{Z}_k - \hat{\mathbf{x}}_k^-) \quad (3.8)$$

$$\mathbf{P}_k = (1 - \mathbf{K}_k) \mathbf{P}_k^- \quad (3.9)$$

3.1.3. Uygulama 1 İçin Uygunluk Fonksiyonu Seçimi

Optimizasyon algoritmalarında her bir olası çözümün kalitesini belirleyen uygunluk fonksiyonu, bu problem için ortalama karesel hata (MSE) Eşitlik 3.10 ile belirlenmiştir.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{x}_i - V_i)^2 \quad (3.10)$$

Bu eşitlikte n değeri yapılan toplam voltaj ölçümü sayısıdır ve 200'e eşittir. $\hat{x}_i$ değeri o anki tahmin edilen voltaj değerini ifade etmektedir. V_i değeri ise o anki gerçek voltaj değerini ifade etmektedir.

3.1.4. Algoritma Temel Parametrelerinin Seçimi

Uygulama 1 için, KS parametrelerinin eniyilenmesinde kullanılacak olan güncel zeki optimizasyon algoritmalarının temel parametrelerinin seçimi yapılmıştır. Uygulamanın daha doğru sonuçlar verebilmesi için nasıl ki KS parametrelerini eniyilemek için optimizasyon algoritmalarını kullanıyorsak, kullandığımız zeki optimizasyon algoritmalarının genel kontrol parametrelerinin de doğru bir şekilde seçilmesi gerekmektedir. Farklı kontrol parametreleri seçimi yapılarak 10'ar adet koşma yapıp ortalamaları alınarak en iyi sonucu veren kontrol parametreleri aşağıdaki gibi belirlenmiştir.

3.1.4.1. GA Temel Parametrelerinin Seçimi

Tablo 3.1. GA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Çaprazlama Sabiti	Mutasyon Oranı
20	20	0.8	0.2

GA temel parametrelerinin seçimi Tablo 3.1'deki gibi gerçekleştirilmiştir. Farklı popülasyon büyüklüğü seçimiyle yapılan denemelerde bu değerin 20 olmasının yeterli olduğu tespit edilmiştir. Ayrıca 20'ye yakın iterasyon sayısından itibaren hata oranında

azalma görülmemiştir. Diğer algoritmalarda da popülasyon büyüklüğü ve iterasyon sayısı 20 olarak alınmıştır. Genetik çeşitliliğin sağlandığı kısım olan çaprazlama işlemi için çaprazlama sabiti 0.8 olarak kabul edilmiştir. Bir diğer GA kontrol parametresi olan mutasyon oranı ise değişimin ne oranda olacağını belirler. Bu oran yüksek seçilirse iyi olan bireyler rastgele yapılan bu değişimle daha kötü bireyler haline gelebilir. Mutasyon oranı bu problem için 0.2 olarak seçilmiştir.

3.1.4.2. YAKA Temel Parametrelerinin Seçimi

Tablo 3.2. YAKA Temel Parametreleri ve Değerleri

Koloni Büyüklüğü	<i>limit</i>	İterasyon Sayısı	Yiyecek Kaynağı
20	20	20	10

YAKA temel parametrelerinin seçimi Tablo 3.2'deki gibi gerçekleştirilmiştir. Koloni büyüklüğünü ifade eden toplam arı sayısı 20 olarak alınmıştır. Sadece ölçüm hata kovaryansı R optimize edileceği için optimize edilecek parametre sayısı 1'dir. 20 seçilen 'limit' sayısı kadar denemeden sonra işçi arılar bir iyileşme görmediği zaman yiyecek kaynağını değiştirir. Döngü sayısı bu algoritma için de 20 olarak belirlenmiştir. Son YAKA kontrol parametresi olan yiyecek kaynağı sayısı, koloni sayısının yarısı kadardır ve 10 olarak belirlenmiştir.

3.1.4.3. DGA Temel Parametrelerinin Seçimi

Tablo 3.3. DGA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Çaprazlama Sabiti	Adım Boyutu
20	20	0.8	1.75

DGA temel parametrelerinin seçimi Tablo 3.3'deki gibi gerçekleştirilmiştir. Bu algoritmada da popülasyon büyüklüğü ve iterasyon sayısı 20 olarak alınmıştır. Yardımlaşmayı ifade eden çaprazlama sabiti 0.8, ölçekleme faktörü 1.75 olarak belirlenmiştir. Yapılan bu uygulamada algoritma statejisi olarak, hızlı yakınsayan küçük popülasyonlu uygulamalar için kullanılan $DGA/best/1$ stratejisi kullanılmıştır.

3.1.4.4. PSO Temel Parametrelerinin Seçimi

Tablo 3.4. PSOA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Parçacık Boyutu	V_{max}	C_1, C_2	w
20	20	1	1000	1.49	1.1

PSOA temel parametrelerinin seçimi Tablo 3.4'deki gibi gerçekleştirilmiştir. Popülasyon büyüklüğü ve iterasyon sayısı diğer algoritmalarda olduğu gibi bu algoritma için de 20 olarak belirlenmiştir. Bir iterasyondaki bir parçacıkta meydana gelebilecek maksimum değışikliği belirten V_{max} 1000 değerini, öğrenme faktörleri C_1, C_2 1.49 değerini ve eylemsizlik ağırlığı w 1.1 değerini almıştır.

3.1.4.5. ABA Temel Parametrelerinin Seçimi

Tablo 3.5. ABA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Alpha	Betamin	Gamma
20	20	0.5	0.2	1

ABA temel parametrelerinin seçimi Tablo 3.5'deki gibi gerçekleştirilmiştir. Popülasyon ve iterasyon sayısı 20, adım büyüklüğü Alpha 0.5, ateş böceklerinin çekicilik oranı Betamin 0.2 ve ateş böceklerinin arasındaki çekicilik değışimini belirleyen parametre Gamma 1 olarak seçilmiştir.

3.1.5. Optimize Edilmiş KS'nin Uygulama Sonuçları

Bu uygulamada, KS gürültülü voltaj değerlerini girdi olarak alarak, büyük oranda gürültüden arındırılmış bir şekilde çıktı olarak vermiştir. Daha doğru sonuçların elde edilmesi için KS parametrelerinden ölçüm hata kovaryansı R değeri optimum şekilde seçilmelidir. R değerinin optimum seçilmesi için KS, zeki optimizasyon teknikleriyle desteklenmiş ve 5 farklı algoritma ile KS parametresi R değeri optimal değerleri almıştır. Kullanıcı deneyimine bırakılan parametre seçiminde doğru sonuca ulaşmak oldukça zordur. Tablo 3.6 ile gösterildiği gibi 3 farklı R değeriyle deneme yapılmış ve oluşan sonuç grafikleri Şekil 3.2, 3.3 ve 3.4 ile verilmiştir. Tablo 3.6 ile kullanılan güncel zeki optimizasyon algoritmalarının her biri için yapılan 30 koşmada hata oranı en yüksek ve en düşük olanların koşma sonunda almış oldukları R parametre değerleri verilmektedir. Tablo 3.7 ile seçilen R parametrelerine karşılık gelen hata dağılımları verilmiştir.

Tablo 3.6. Uygulama 1 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafından seçilen KS parametreleri

Algoritma	R
Standart-KS (Deneme 1)	0.2
Standart-KS (Deneme 2)	1
Standart-KS (Deneme 3)	0.5
GA-KS (En iyi)	0.031426176
GA-KS (En kötü)	0.031717707
YAKA-KS (En iyi)	0.031510357
YAKA-KS (En kötü)	0.031510356
DGA-KS(En iyi)	0.03152541
DGA-KS(En kötü)	0.03154322
PSOA-KS(En iyi)	0.031510795
PSOA-KS(En kötü)	0.031510553
ABA-KS(En iyi)	0.031511831
ABA-KS(En kötü)	0.189165975

Tablo 3.7 incelendiğinde bu uygulama için standart sapması en küçük olan YAKA, kararlılığı en yüksek olan algoritmadır. 9.662519E-03 değeri ile YAKA en düşük MSE değerine sahiptir. Bir diğer performans kriteri olan işlem süresi karşılaştırması yapıldığında ise YAKA'nın diğer 4 algoritmaya göre daha hızlı çalıştığı gözlemlenmiştir.

Tablo 3.7. Kullanılan algoritmaların uygulama 1 için hata dağılımı

Uygulama 1 için Algoritmaların Hata Dağılımı						
Algoritmalar	En İyi	En Kötü	Medyan	Ortalama	Standart Sapma	İşlem Süresi (sn)
GA	9.662519E-03	9.662568E-03	9.662521E-03	9.662527E-03	1.119608E-08	0.2886
YAKA	9.662519E-03	9.662519E-03	9.662519E-03	9.662519E-03	1.080102E-12	0.1799
DGA	9.662519E-03	9.662527E-03	9.662519E-03	9.662520E-03	1.570894E-09	0.2184
PSOA	9.662519E-03	9.901138E-03	9.662519E-03	9.670474E-03	4.356559E-05	0.2782
ABA	9.662519E-03	1.388682E-02	9.662519E-03	9.854601E-03	7.974297E-04	0.1940

Uygulama 1 için yapılan istatistiksel analizde, Tablo 3.8 ile Kruskal Wallis test sonucunda anlamlılık değerinin (p), 0.00000000082659 değerine eşit olması bu algoritmalar arasında anlamlı farkın olduğunu göstermektedir. Tablo 3.9 incelendiğinde Mann-Whitney U testinin tüm algoritmaların ikili kombinasyonları için uygulandığı ve bu algoritmaların birbirlerine göre anlamlı farka sahip olup olmadıkları gözlemlenmiştir.

Şekil 3.1'e göre DGA, optimum sonuca yakınsaması en hızlı olan algoritmadır. Yaklaşık 10. iterasyondan sonra YAKA, GA, PSO ve DGA aynı derecede ortalama hata değerine ulaşmıştır. ABA ise geç yakınsaması ve hata oranının daha yüksek olmasından dolayı bu

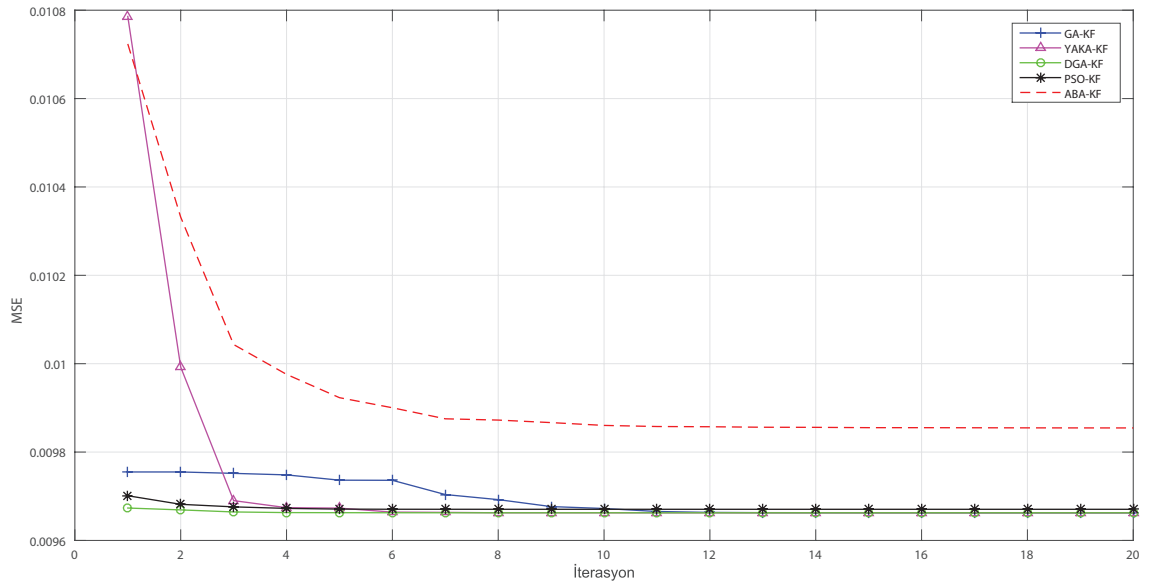
Tablo 3.8. Uygulama 1 için tüm algoritmalarda Kruskal Wallis test sonucu

Algoritma	N	df	Chi_Square	p
GA	30	4	48.276	0.00000000082659
YAKA	30			
DGA	30			
PSOA	30			
ABA	30			

Tablo 3.9. Uygulama 1 için ikili algoritmalarda Mann-Whitney U test sonucu

Algoritmalar	Z	Ortalamaların Farkı	p	Anlamlı Fark
GA-YAKA	-5.651	8.13090469E-09	1.59350E-08	YAKA Lehine
GA-DGA	-4.402	7.57560362E-09	1.10000E-05	DGA Lehine
GA-PSOA	-4.303	-7.94631779E-06	1.70000E-05	GA Lehine
GA-ABA	-3.572	-1.92074016E-04	3.54000E-04	GA Lehine
YAKA-DGA	-2.313	-5.55301069E-10	2.07140E-02	YAKA Lehine
YAKA-PSOA	-2.313	-7.95444869E-06	2.07070E-02	YAKA Lehine
YAKA-ABA	-2.313	-1.92082146E-04	2.07140E-02	YAKA Lehine
DGA-PSOA	-0.498	-7.95389339E-06	6.18411E-01	Yok
DGA-ABA	-0.36	-1.92081591E-04	7.18761E-01	Yok
PSOA-ABA	-0.148	-1.84127698E-04	8.82294E-01	Yok

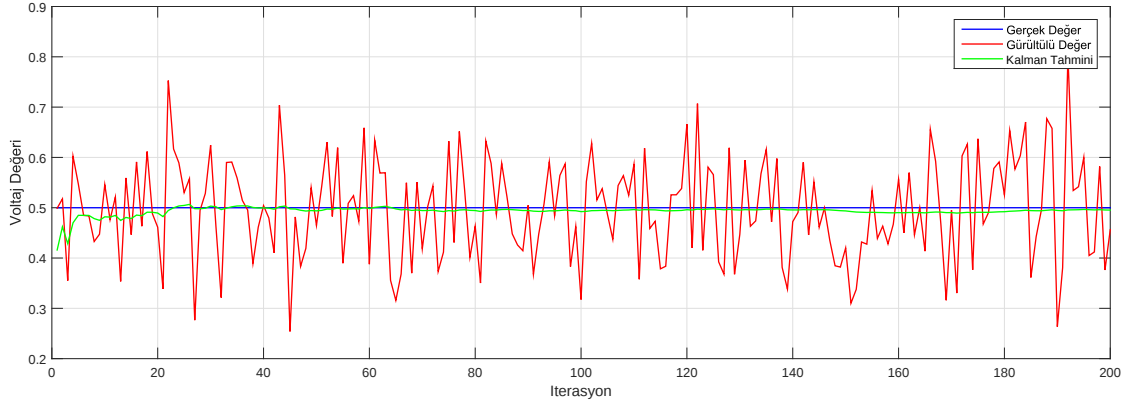
uygulama için doğruluk kriterine göre en son tercih edilen zeki optimizasyon algoritması olarak kabul edilebilir.



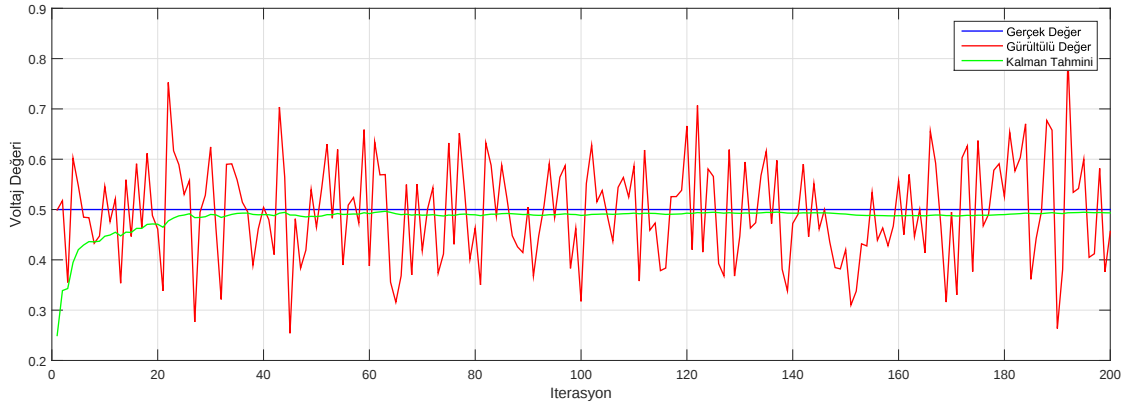
Şekil 3.1. Uygulama 1 için yöntemlerin iterasyon-MSE değişimi

Şekil 3.2, 3.3 ve 3.4 bu uygulama için uygulanan deneme 1 (D1), deneme 2 (D2),

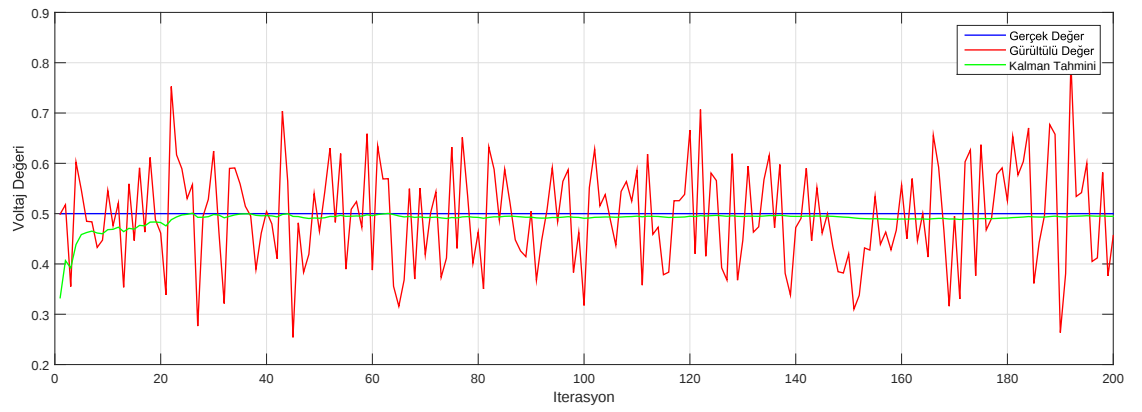
deneme 3 (D3) kullanıcı deneyimi tabanlı KS algoritmasının aldığı R parametresiyle oluşan sonuçları vermektedir. Kullanıcı deneyimine bağlı olarak seçilen R değeriyle elde edilen sonuçlar incelendiğinde doğru sonuçların elde edilmesi için KS parametreleri seçiminin kullanıcı deneyimine bırakılması yerine zeki optimizasyon algoritmalarıyla belirlenmesinin daha uygun olduğu görülmüştür.



Şekil 3.2. Uygulama 1, kullanıcı deneyimi tabanlı KS (D1)

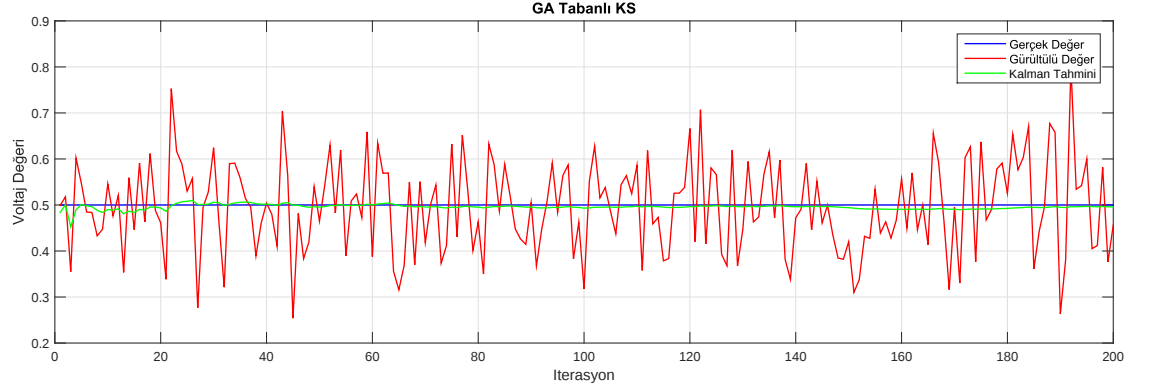


Şekil 3.3. Uygulama 1, kullanıcı deneyimi tabanlı KS (D2)

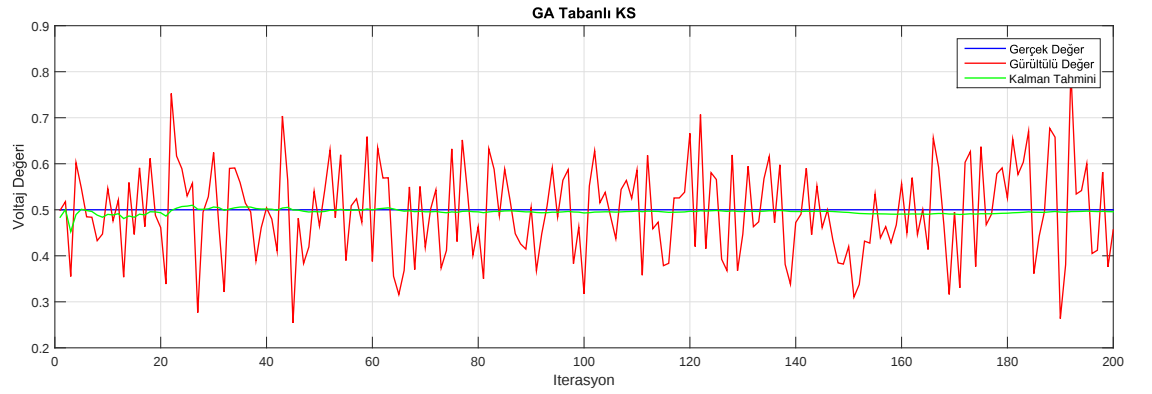


Şekil 3.4. Uygulama 1, kullanıcı deneyimi tabanlı KS (D3)

Uygulama 1 için yapılan 30 koşmalık GA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.5 ile, alınan en büyük hatalı sonuç Şekil 3.6 ile gösterilmiştir.

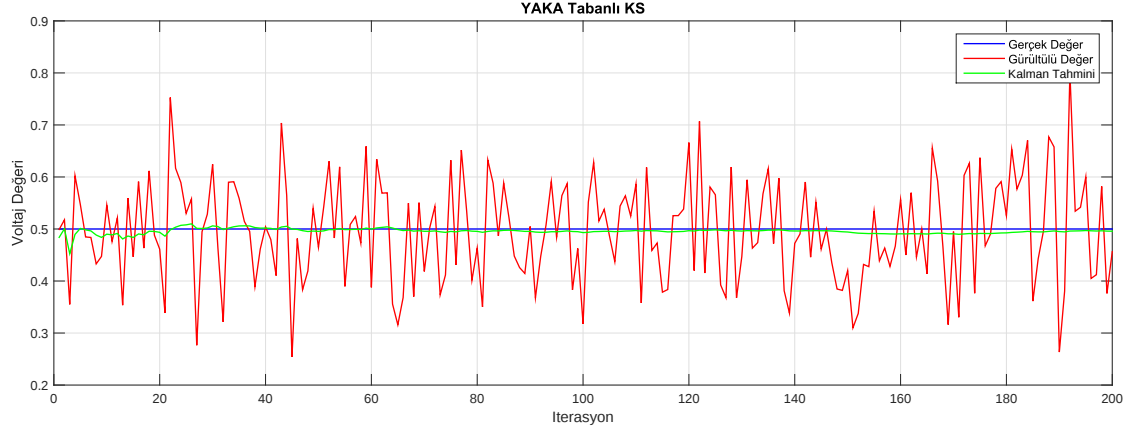


Şekil 3.5. Uygulama 1, GA tabanlı KS için 30 koşmanın en iyisi

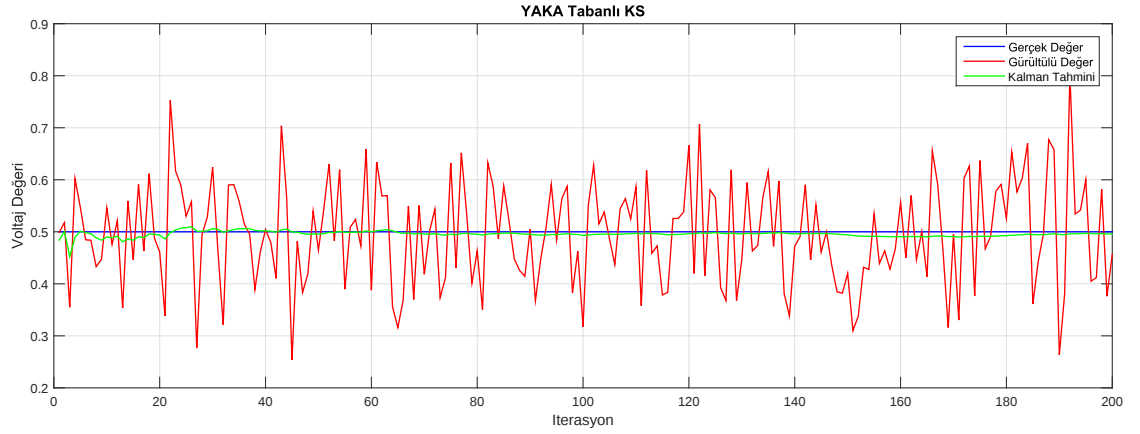


Şekil 3.6. Uygulama 1, GA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 1 için yapılan 30 koşmalık YAKA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.7 ile, alınan en büyük hatalı sonuç Şekil 3.8 ile gösterilmiştir.

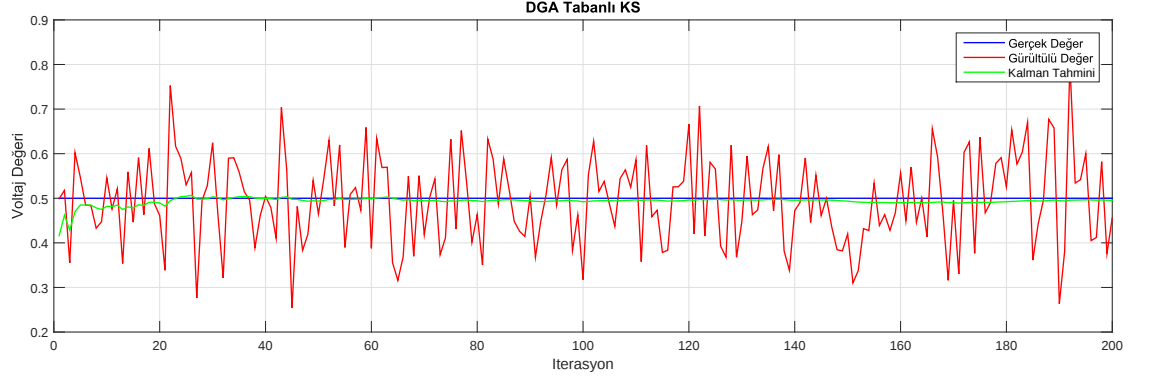


Şekil 3.7. Uygulama 1, YAKA tabanlı KS için 30 koşmanın en iyisi

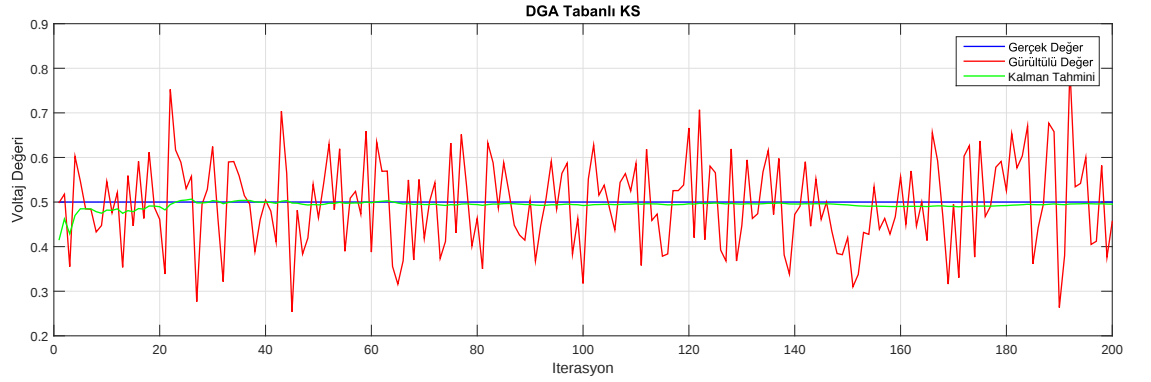


Şekil 3.8. Uygulama 1, YAKA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 1 için yapılan 30 koşmalık DGA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.9 ile, alınan en büyük hatalı sonuç Şekil 3.10 ile gösterilmiştir.

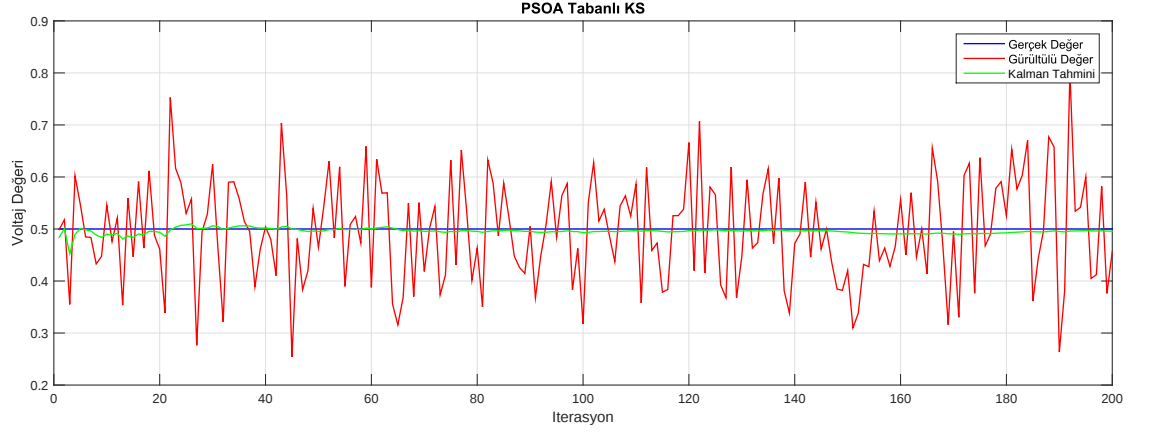


Şekil 3.9. Uygulama 1, DGA tabanlı KS için 30 koşmanın en iyisi

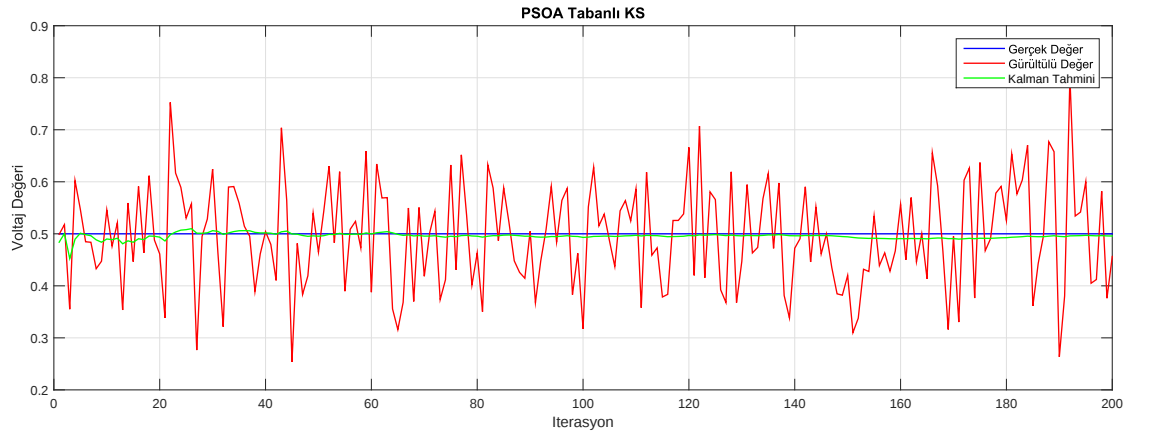


Şekil 3.10. Uygulama 1, DGA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 1 için yapılan 30 koşmalık PSOA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.11 ile, alınan en büyük hatalı sonuç Şekil 3.12 ile gösterilmiştir.

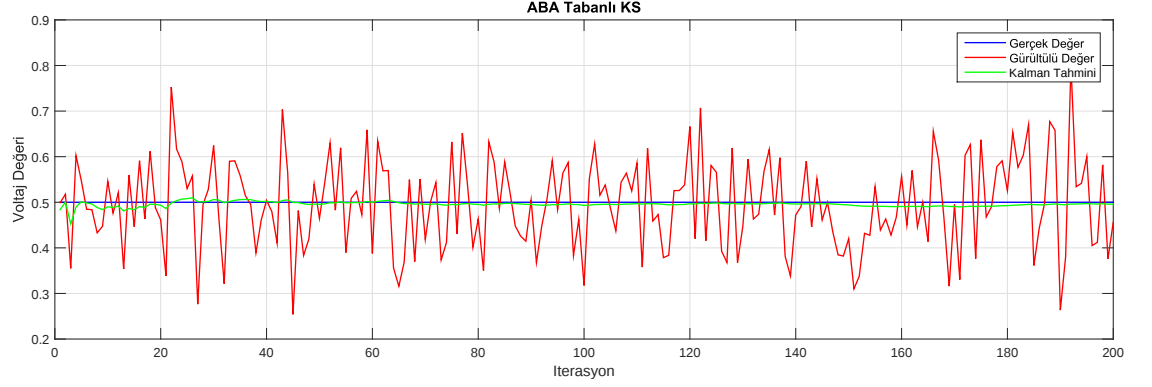


Şekil 3.11. Uygulama 1, PSOA tabanlı KS için 30 koşmanın en iyisi

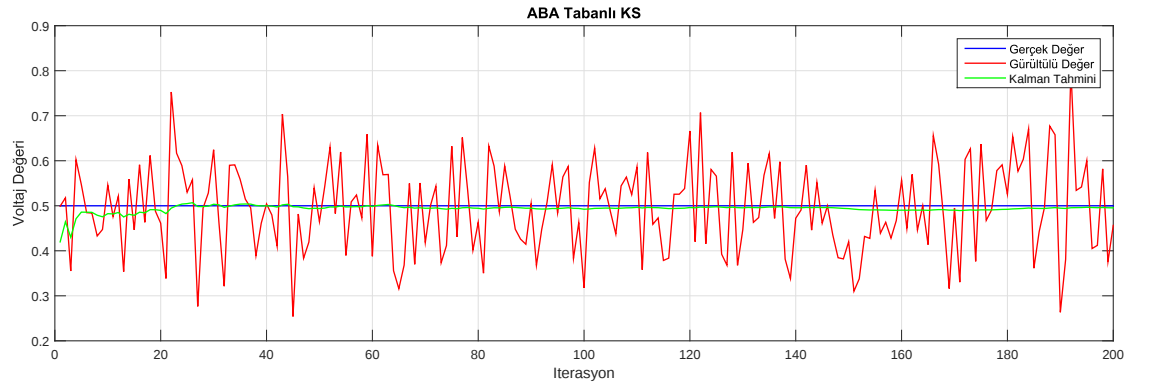


Şekil 3.12. Uygulama 1, PSOA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 1 için yapılan 30 koşmalık ABA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.13 ile, alınan en büyük hatalı sonuç Şekil 3.14 ile gösterilmiştir.



Şekil 3.13. Uygulama 1, ABA tabanlı KS için 30 koşmanın en iyisi



Şekil 3.14. Uygulama 1, ABA tabanlı KS için 30 koşmanın en kötüsü

3.2. Uygulama 2

3.2.1. Problem Tanımı

45 derecelik açı ve 100 birimlik hızla yatay bir nesne atışı yapılacaktır. Her 0.1 saniyede hem dış kamerayla gürültülü bir ölçüm yapılacak hem de nesnenin içerisindeki hassas sensörlerle gerçek konum değerleri ölçülecektir. Sadece gürültülü ölçüm değerlerini KS'ye vererek Kalman tahmini değerleri hesaplanacak ve gerçek konum, gürültülü konum, tahmin edilen konum grafik üzerinde incelenecektir.

3.2.2. Problemin Kalman Süzgeçleme Modeline Uyarlanması

Eğik harekete ait konum hız denklemleri aşağıdaki gibidir:

$$X(t) = X_0 + V_{0x}t \quad (3.11)$$

$$V_x(t) = V_{0x} \quad (3.12)$$

$$Y(t) = Y_0 + V_{0y}t - \frac{1}{2}gt^2 \quad (3.13)$$

$$V_y(t) = V_{0y} - gt \quad (3.14)$$

KS ayrık yapıda olduğundan dolayı, yukarıdaki denklemlerin ayrık biçimde düzenlenmesi aşağıdaki gibidir:

$$X_n = X_{n-1} + V_{xn-1}\Delta t \quad (3.15)$$

$$V_{xn} = V_{xn-1} \quad (3.16)$$

$$Y_n = Y_{n-1} + V_{yn-1}\Delta t - \frac{1}{2}g\Delta t^2 \quad (3.17)$$

$$V_{yn} = V_{yn-1} - g\Delta t \quad (3.18)$$

Bu denklemlerin farklı bir formatta toparlanmış hali aşağıdaki gibidir:

$$\begin{bmatrix} X_n \\ V_{xn} \\ Y_n \\ V_{yn} \end{bmatrix} = \begin{bmatrix} X_{n-1} + V_{xn-1}\Delta t \\ V_{xn-1} \\ Y_{n-1} + V_{yn-1}\Delta t \\ V_{yn-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{2}g\Delta t^2 \\ -g\Delta t \end{bmatrix} \quad (3.19)$$

Eşitlik (3.19)'den görüldüğü üzere her satıra ait kolonların birleşimi eğik atışın ayrık zamanlı denklemini vermektedir. Üstteki ifade, KS'ye ait olan durum geçiş matrisini,

$$\text{kontrol matrisini ve kontrol sinyali bulmak için } X_n = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{n-1} +$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{2}g\Delta t^2 \\ -g\Delta t \end{bmatrix} \text{ gibi gösterilebilir:}$$

Kalman süzgeçlemenin temel iki denklemi aşağıdaki gibidir:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{w}_{k-1} \quad (3.20)$$

$$\mathbf{Z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \quad (3.21)$$

Eşitlik (3.20) ve Eşitlik (3.21) modele uygun olarak düzenlenmiştir. Dolayısıyla

$$\text{Eşitlik (3.20)'daki durum geçiş matrisi } \mathbf{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ kontrol matrisi } \mathbf{B} =$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ ve kontrol sinyali } \mathbf{u}_k = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{2}g\Delta t^2 \\ -g\Delta t \end{bmatrix} \text{ olarak bulunmaktadır. Bir}$$

durum, gözlem matrisi $\mathbf{H}$ ile çarpıldığında ölçüm gösterimine dönüşmektedir. Ölçüm gösterimine dönüştürmek için durumdaki tüm değerler 1 ile çarpılır. Dolayısıyla $\mathbf{H} =$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ olarak verilmektedir.}$$

Güncel hata kovaryansı $\mathbf{P}$, daha iyi bir durum bilinmediği için başlangıç değeri olarak 1

$$\text{alınıyor. Dolayısıyla } \mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ olarak verilmektedir.}$$

Tahmini işlem hata kovaryansı $\mathbf{Q}$, işlem hatası yapılmadığı varsayılarak bütün değerler

için 0 olarak alınıyor. Dolayısıyla $\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ olarak verilmektedir.

Tahmini ölçüm hata kovaryansı $\mathbf{R}$, tam olarak bilinmemektedir. Ama bu uygulama için bütün değerlere 0.2 verilebilir. İlgili kodda bu değer değiştirilerek sonuçlar incelenebilir.

Dolayısıyla $\mathbf{R} = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix}$ olarak verilmektedir.

Problem tanımında 45 derecelik açı ve 100 birimlik hızla atılan bir nesne atışı yapılmaktaydı. Konum ve hız durumu için KS'ye ilk tahmin(X_0) gönderilmesi

gerekmektedir. Bu problem için verilen ilk tahmin $X_0 = \begin{bmatrix} X \\ V_x \\ Y \\ V_y \end{bmatrix} = \begin{bmatrix} 0 \\ 100 \cos\left(\frac{\pi}{4}\right) \\ 0 \\ 100 \sin\left(\frac{\pi}{4}\right) \end{bmatrix}$

olarak verilsin.

Bu bilgilerden sonra KS algoritmasının 2 adımı aşağıdaki gibi oluşur:

1. Zaman Güncelleme (Kestirim):

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} \quad (3.22)$$

$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q} \quad (3.23)$$

2. Ölçüm Güncelleme (Doğrulama):

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.24)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{Z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \quad (3.25)$$

$$\mathbf{P}_k = (1 - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \quad (3.26)$$

3.2.3. Uygulama 2 İçin Uygunluk Fonksiyonu Seçimi

Uygulama 2 için belirlenen uygunluk fonksiyonu Eşitlik 3.27 ile ifade edilmiştir.

$$OrtalamaNorm(P) = \frac{1}{n} \sum_{i=1}^n norm(P_i) \quad (3.27)$$

Bu eşitlikte n değeri, eğik atış uygulamasında yapılan toplam ölçüm sayısıdır. $norm$ fonksiyonu bir matrisin büyüklüğünü ifade etmektedir. P_i değeri ise güncel hata kovaryansını ifade etmektedir.

3.2.4. Algoritma Temel Parametrelerinin Seçimi

Bu bölümde uygulama 2 için, KS parametrelerinin eniyilenmesinde kullanılacak olan güncel zeki optimizasyon algoritmalarının temel parametrelerinin seçimi yapılmıştır. Farklı kontrol parametreleri seçimi yapılarak 10'ar adet koşma yapıp ortalamaları alınarak en iyi sonucu veren kontrol parametreleri aşağıdaki gibi belirlenmiştir.

3.2.4.1. GA Temel Parametrelerinin Seçimi

Tablo 3.10. GA Temel Parametreleri ve Değerleri

Koloni Sayısı	İterasyon	Çaprazlama Sabiti	Mutasyon Oranı
50	250	0.8	0.2

GA temel parametrelerinin seçimi Tablo 3.10'deki gibi gerçekleştirilmiştir. Algoritmalar arasında karşılaştırma yapılabilmesi için ortak değer alması gereken popülasyon büyüklüğü ve iterasyon sayısının yapılan koşmalarla birlikte sırasıyla 50 ve 20 olarak seçilmesi gerektiği saptanmıştır. İyi genlerin kaybolmaması için büyük seçilen çaprazlama sabiti 0.8, değişim olarak farklılığı sağlayan ama küçük değer seçilmesi daha uygun olan mutasyon oranı için de 0.2 değeri belirlenmiştir.

3.2.4.2. YAKA Temel Parametrelerinin Seçimi

Tablo 3.11. YAKA Temel Parametreleri ve Değerleri

Koloni Büyüklüğü	<i>limit</i>	İterasyon Sayısı	Yiyecek Kaynağı
50	20	250	25

YAKA temel parametrelerinin seçimi Tablo 3.11'deki gibi gerçekleştirilmiştir. Koloni büyüklüğü 50 olarak seçilmiştir. Ölçüm hata kovaryansı parametrelerinden R_1 ve R_2 değerleri optimize edileceği için optimize edilecek parametre sayısı 2'dir. 20 seçilen 'limit' sayısı kadar denemeden sonra işçi arılar gelişim sağlanmazsa yiyecek kaynağını değiştirir. Döngü sayısı bu algoritma için de 20 olarak belirlenmiştir. Son YAKA kontrol parametresi olan yiyecek kaynağı sayısı, koloni sayısının yarısı kadardır ve 25 olarak hesaplanmıştır.

3.2.4.3. DGA Temel Parametrelerinin Seçimi

Tablo 3.12. DGA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Çaprazlama Sabiti	Adım Boyutu
50	250	0.8	1.75

DGA temel parametrelerinin seçimi Tablo 3.12'deki gibi gerçekleştirilmiştir. Diğer algoritmalarla ortak değer olan popülasyon büyüklüğü ve iterasyon sayısı sırasıyla 50 ve 20 olarak alınmıştır. Yardımlaşma değeri olan çaprazlama sabiti için genel kabul görmüş olan 0.8 değeri, ölçekleme faktörü olan adım boyutu 1.75 olarak belirlenmiştir. Bu uygulamada da voltaj uygulamasında olduğu gibi *DGA/best/1* algoritma stratejisi kullanılmıştır.

3.2.4.4. PSOA Temel Parametrelerinin Seçimi

Tablo 3.13. PSOA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Parçacık Boyutu	V_{max}	C_1, C_2	w
50	250	2	[1000 1000]	1.49	1.1

PSOA temel parametrelerinin seçimi Tablo 3.13'deki gibi gerçekleştirilmiştir. Popülasyon büyüklüğü ve iterasyon sayısı diğer algoritmalarda olduğu gibi bu algoritma için de sırasıyla 50 ve 20 olarak belirlenmiştir. Bir iterasyondaki bir parçacıkta meydana gelebilecek maksimum değişikliği belirten V_{max} [1000 1000] değerini, öğrenme faktörleri C_1, C_2 1.49 değerini ve eylemsizlik ağırlığı w 1.1 değerini almıştır.

3.2.4.5. ABA Temel Parametrelerinin Seçimi

Tablo 3.14. ABA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Alpha	Betamin	Gamma
50	250	0.5	0.2	1

ABA temel parametrelerinin seçimi Tablo 3.14'deki gibi gerçekleştirilmiştir. Popülasyon sayısı 50, iterasyon sayısı 20, adım büyüklüğü Alpha 0.5, ateş böceklerinin çekicilik oranı Betamin 0.2 ve ateş böceklerinin arasındaki çekicilik değişimini belirleyen parametre

Gamma 1 olarak seçilmiştir.

3.2.5. Optimize Edilmiş KS'nin Uygulama Sonuçları

Bu uygulamada, KS eğik atışta gürültülü konum değerlerinin gürültüden arındırılması için kullanılmıştır. KS parametrelerinden ölçüm hata kovaryansı R_1 ve R_2 değerlerinin optimal seçilmesi daha doğru sonuçlar alınabilmesi açısından önemlidir. Bu uygulamada KS parametrelerinden R_1 ve R_2 değerleri öncelikle kullanıcı deneyimine bırakılarak Tablo 3.15 ile verilen değerleri almıştır. Aldığı bu değerlerle ortaya çıkan sonuç grafikleri Şekil 3.16, 3.17 ve 3.18 ile verilmiştir. 30'ar koşma yapılan 5 farklı optimizasyon algoritması için koşmalar sonunda belirlenen en iyi ve en kötü sonuca karşılık gelen R_1 ve R_2 parametre değerleri Tablo 3.15 ile sunulmuştur. 30'ar koşma sonrası algoritmaların hata dağılımları Tablo 3.16 ile verilmiştir.

Tablo 3.15. Uygulama 2 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafında seçilen KS parametreleri

Algoritma	R1	R2
Standart-KS (V1)	0.2	0.2
Standart-KS (V2)	0.01	1
Standart-KS (V3)	0.5	0.01
GA-KS (En iyi)	0.00001	0.00001
GA-KS (En kötü)	0.0079	0.0055
YAKA-KS (En iyi)	0.0001	0.0001
YAKA-KS (En kötü)	0.0001	0.0001
DGA-KS(En iyi)	0.19897	0.18992
DGA-KS(En kötü)	0.2101	0.19635
PSOA-KS(En iyi)	0.00001	0.00001
PSOA-KS(En kötü)	0.00001	0.00001
ABA-KS(En iyi)	0.001	0.001
ABA-KS(En kötü)	0.0308	0.1643

Tablo 3.16 incelendiğinde standart sapması en küçük olan YAKA ve PSOA kararlılığı en yüksek olan algoritmalar. 2.215384E-16 değeri ile YAKA ve PSOA en düşük ortalama norm(P) değerine sahiptir. Bir diğer performans kriteri olan işlem süresi karşılaştırması yapıldığında ABA'nın diğer algoritmalarla göre daha hızlı çalıştığı gözlemlenmiştir.

Uygulama 2 için yapılan istatistiksel analizde, Tablo 3.17 ile Kruskal Wallis test sonucunda anlamlılık değerinin (p), 2.52E-24 değerine eşit olması bu algoritmalar arasında anlamlı

Tablo 3.16. Kullanılan algoritmaların uygulama 2 için hata dağılımı

Uygulama 2 İçin Algoritmaların Hata Dağılımı						
Algoritmalar	En İyi	En Kötü	Medyan	Ortalama	Standart Sapma	İşlem Süresi (sn)
GA	9.999991E-07	1.264826E-05	3.732849E-06	4.071810E-06	3.073522E-06	8.9435
YAKA	2.215384E-16	2.215384E-16	2.215384E-16	2.215384E-16	0.000000E+00	8.5983
DGA	3.730578E-08	2.003408E-04	6.549620E-06	2.535909E-05	4.731570E-05	11.0672
PSOA	2.215384E-16	2.215384E-16	2.215384E-16	2.215384E-16	0.000000E+00	10.8389
ABA	4.047695E-07	8.136607E-06	2.924351E-06	3.072231E-06	1.724239E-06	8.4199

farkın olduğunu göstermektedir. Tablo 3.18 incelendiğinde Mann-Whitney U testinin tüm algoritmaların ikili kombinasyonları için uygulandığı ve bu algoritmaların birbirlerine göre anlamlı farka sahip olup olmadıkları gözlemlenmiştir.

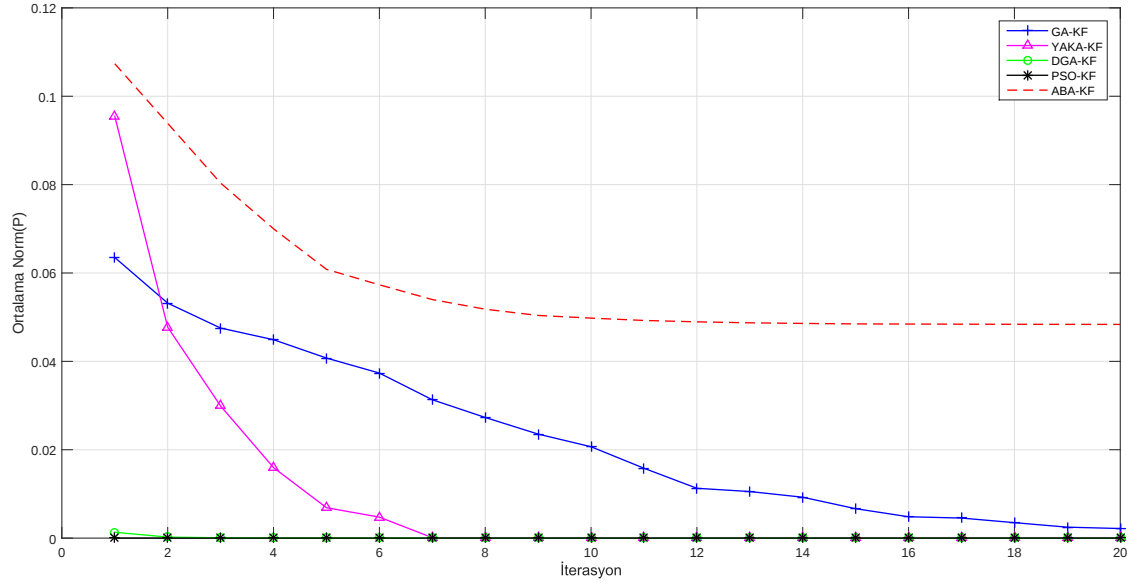
Tablo 3.17. Uygulama 2 için tüm algoritmalarda Kruskal Wallis test sonucu

Algoritma	N	df	Chi_Square	p
GA	30	4	116.846	2.52E-24
YAKA	30			
DGA	30			
PSOA	30			
ABA	30			

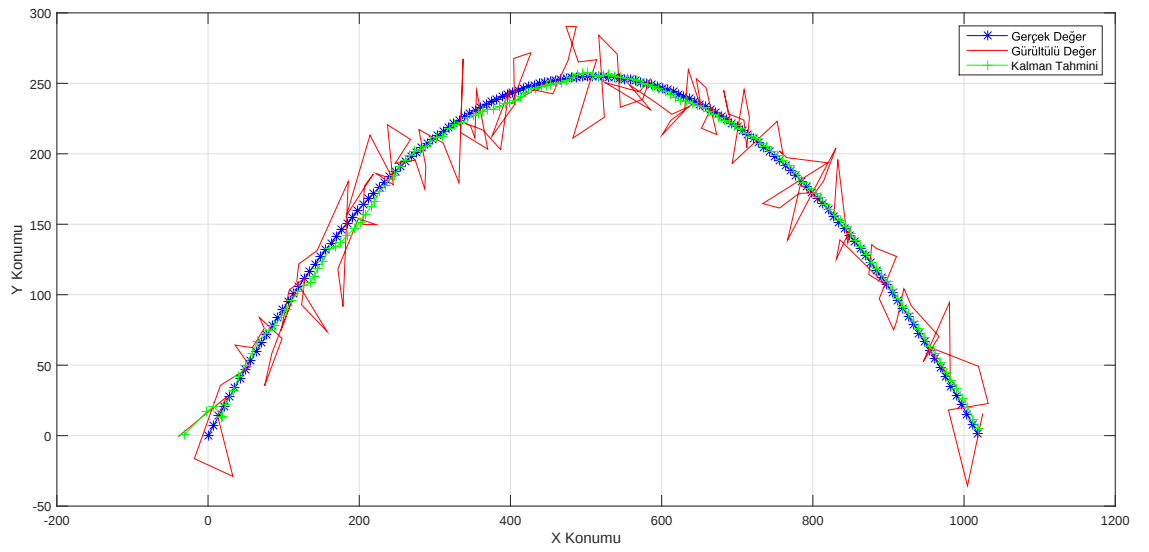
Tablo 3.18. Uygulama 2 için ikili algoritmalarda Mann-Whitney U test sonucu

Algoritmalar	Z	Ortalamaların Farkı	p	Anlamlı Fark
GA-YAKA	-7.131	4.07181034E-06	9.99E-13	YAKA Lehine
GA-DGA	-1.467	-2.12872814E-05	0.142	Yok
GA-PSOA	-7.131	4.07181034E-06	9.99E-13	PSOA Lehine
GA-ABA	-1.215	9.99579481E-07	0.224	Yok
YAKA-DGA	-7.112	-2.53590918E-05	1.14E-12	YAKA Lehine
YAKA-PSOA	0	0.00000000E+00	1	Yok
YAKA-ABA	-7.112	-3.07223086E-06	1.14E-12	YAKA Lehine
DGA-PSOA	-7.112	2.53590918E-05	1.14E-12	PSOA Lehine
DGA-ABA	-2.218	2.22868609E-05	0.026578	ABA Lehine
PSOA-ABA	-7.112	-3.07223086E-06	1.14E-12	PSOA Lehine

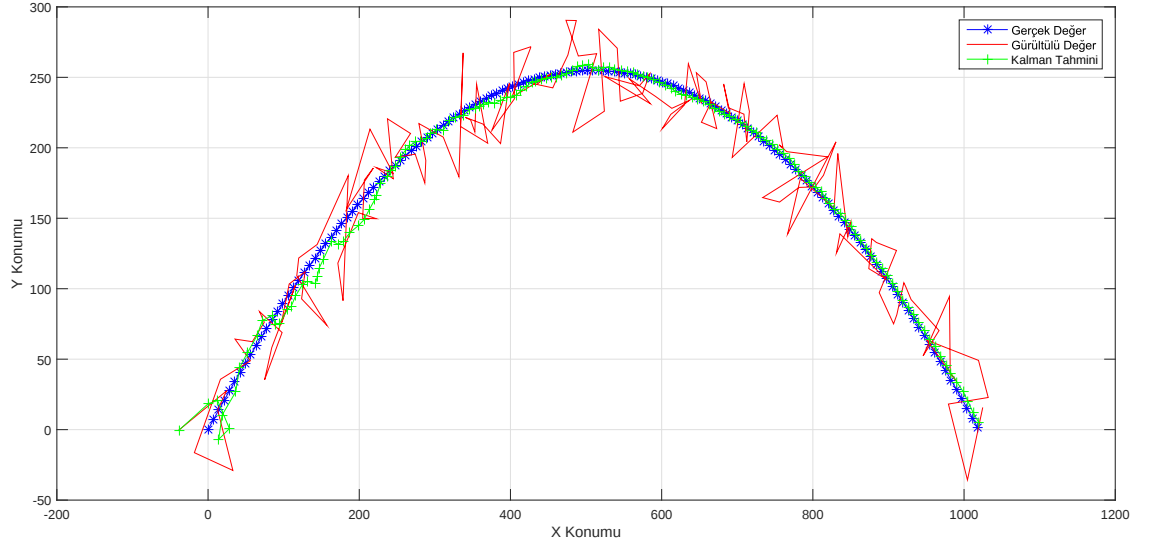
Uygulama 2 için yapılan kullanıcı deneyimi tabanlı KS çalışmasında, ortaya çıkan sonuçlar Şekil 3.16, 3.17 ve 3.18’de verilmiştir.



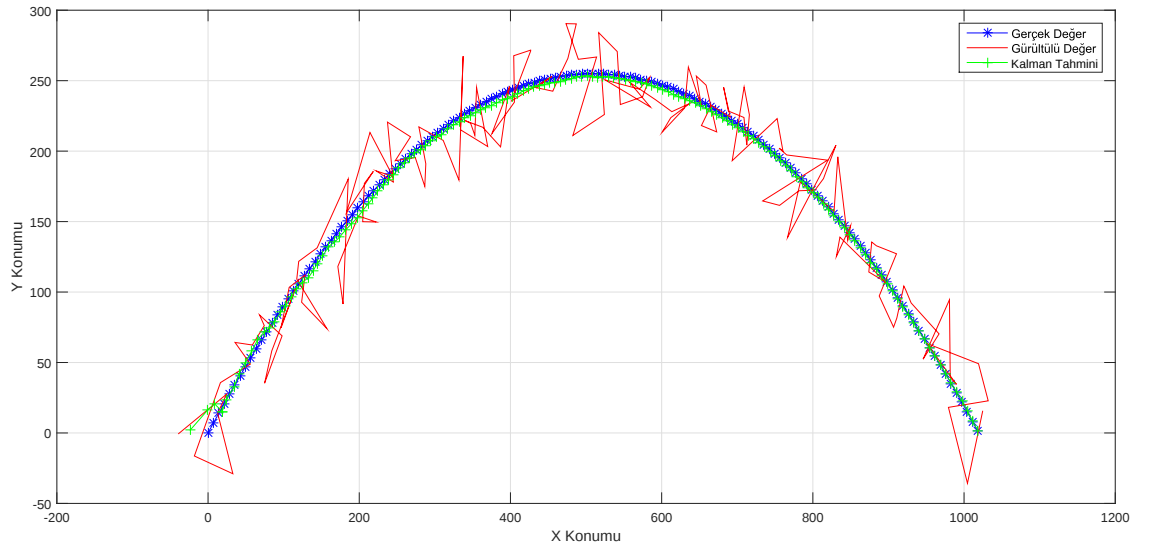
Şekil 3.15. Uygulama 2 için yöntemlerin iterasyon-ortalama norm(P) değişimi



Şekil 3.16. Uygulama 2, kullanıcı deneyimi tabanlı KS (D1)

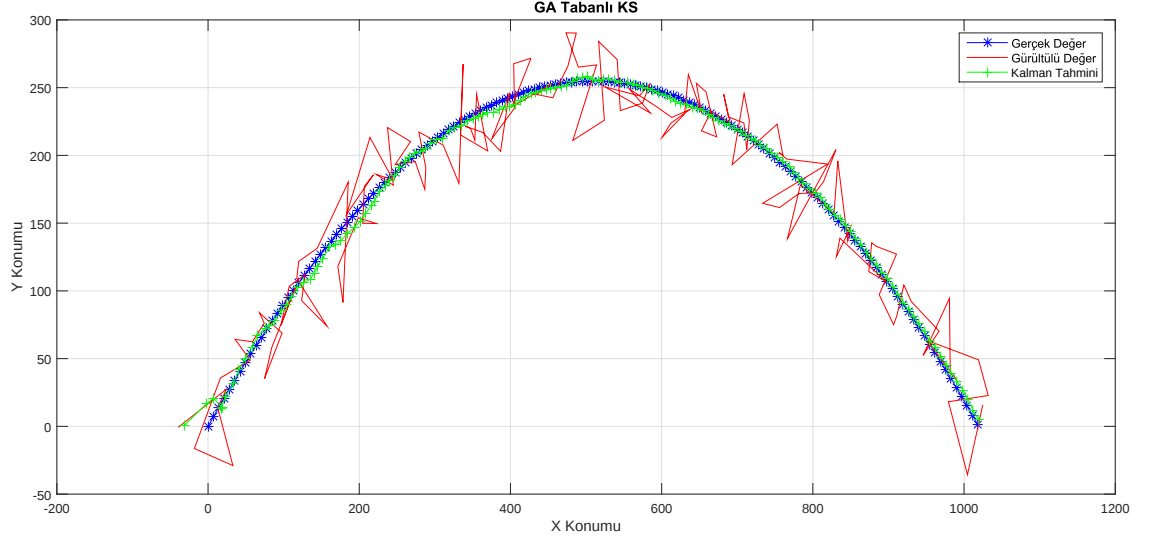


Şekil 3.17. Uygulama 2, kullanıcı deneyimi tabanlı KS (D2)

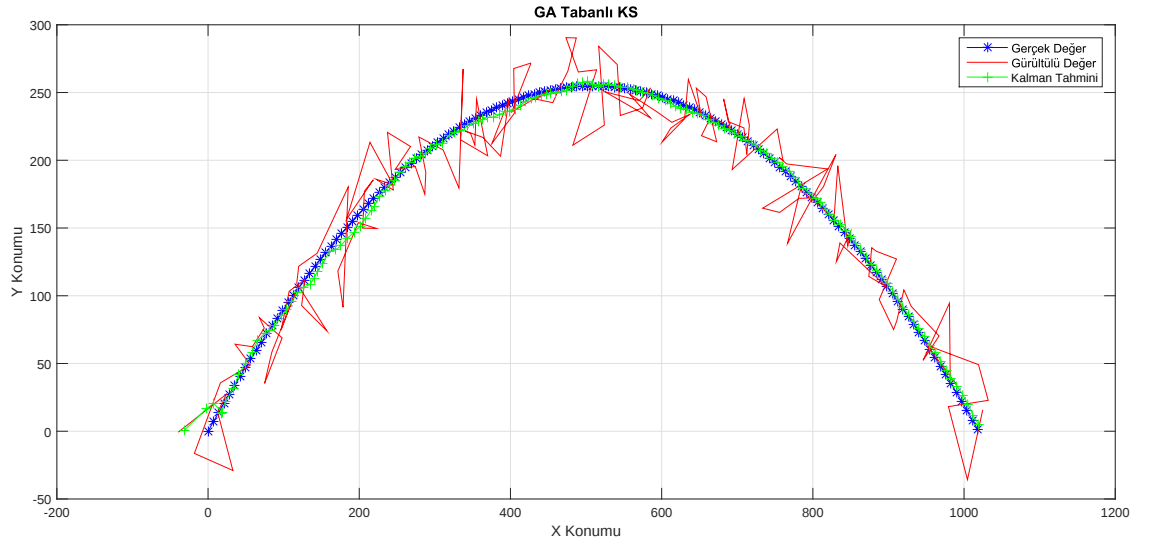


Şekil 3.18. Uygulama 2, kullanıcı deneyimi tabanlı KS (D3)

Uygulama 2 için yapılan 30 koşmalık GA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.19 ile, alınan en büyük hatalı sonuç Şekil 3.20 ile gösterilmiştir.

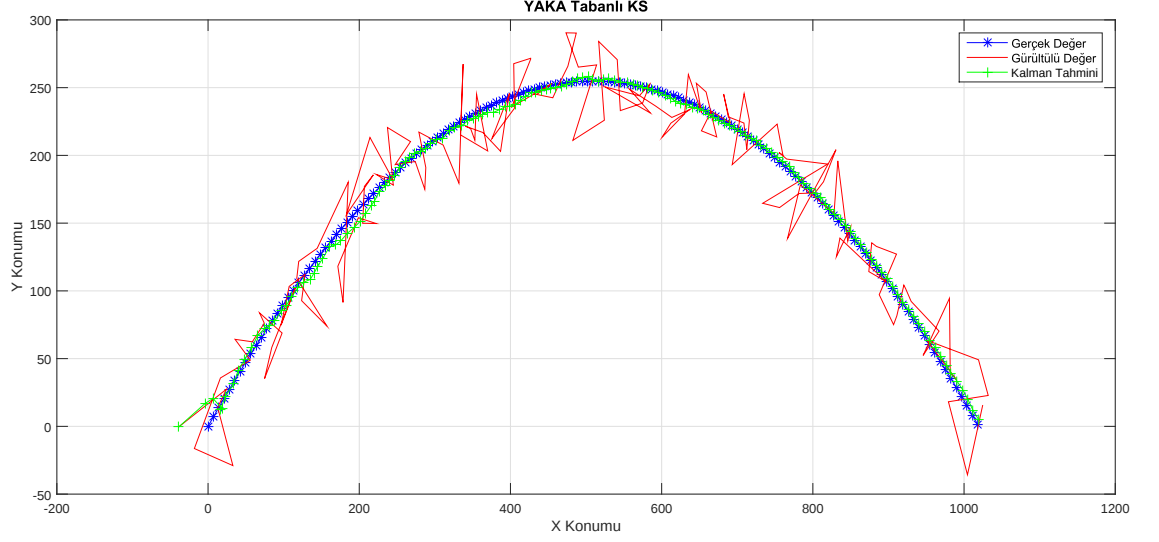


Şekil 3.19. Uygulama 2, GA tabanlı KS için 30 koşmanın en iyisi

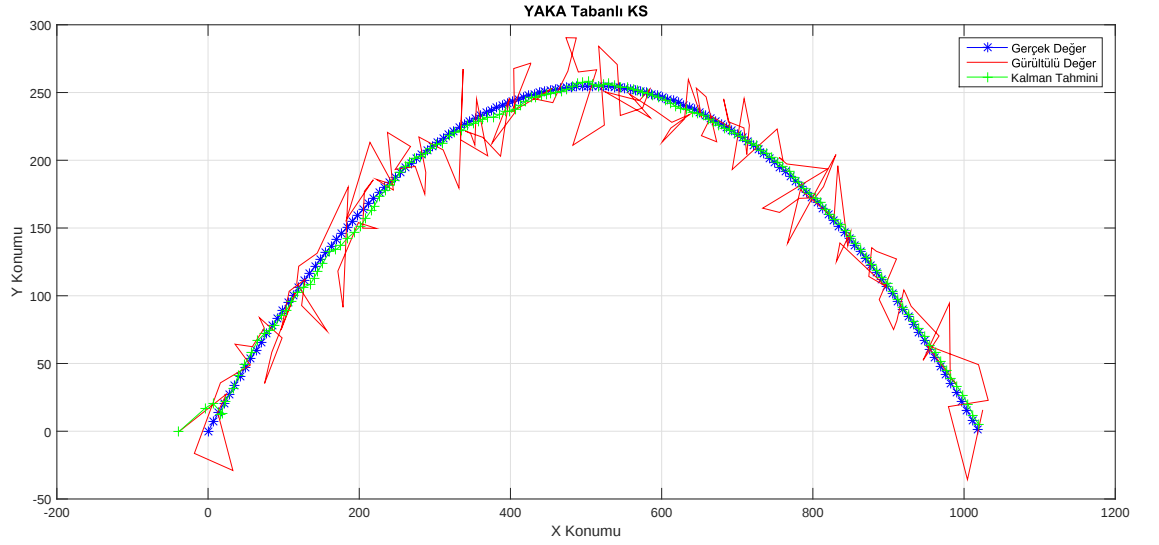


Şekil 3.20. Uygulama 2, GA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 2 için yapılan 30 koşmalık YAKA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.21 ile, alınan en büyük hatalı sonuç Şekil 3.22 ile gösterilmiştir.

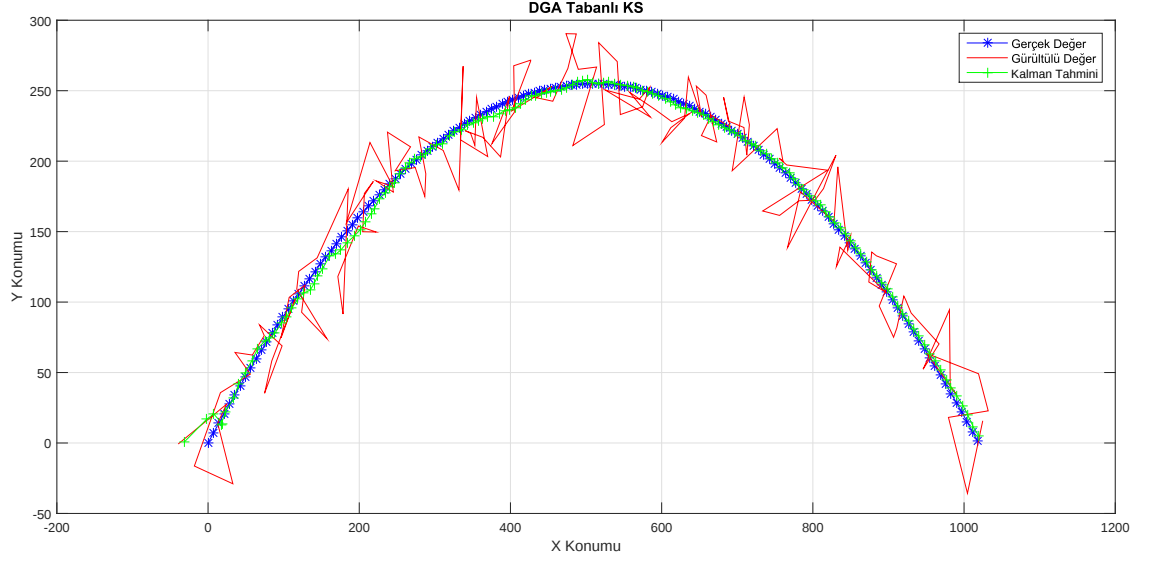


Şekil 3.21. Uygulama 2, YAKA tabanlı KS için 30 koşmanın en iyisi

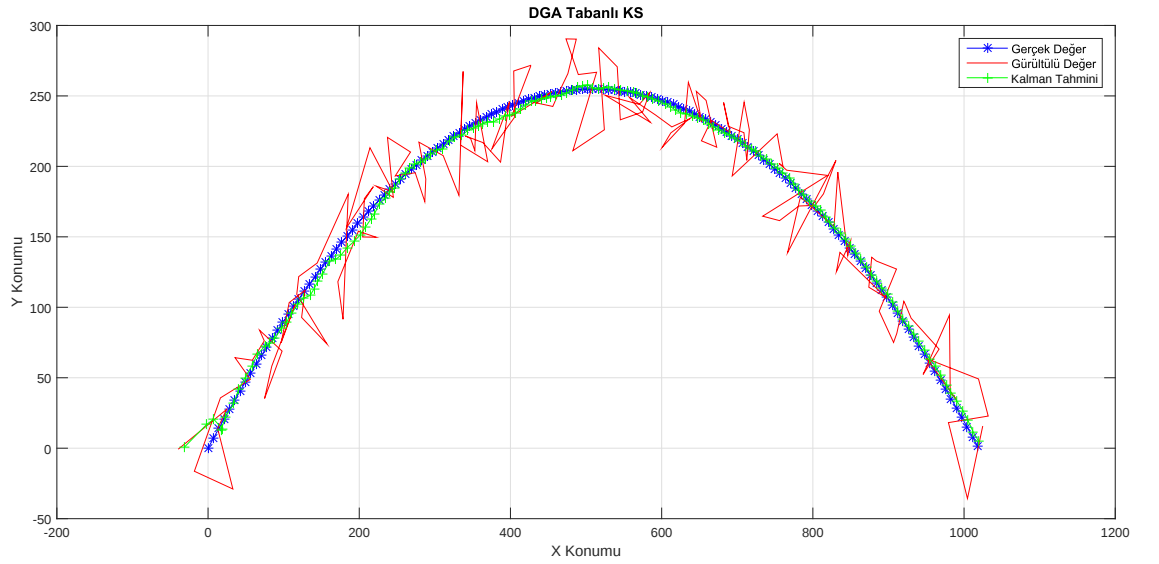


Şekil 3.22. Uygulama 2, YAKA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 2 için yapılan 30 koşmalık DGA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.23 ile, alınan en büyük hatalı sonuç Şekil 3.24 ile gösterilmiştir.

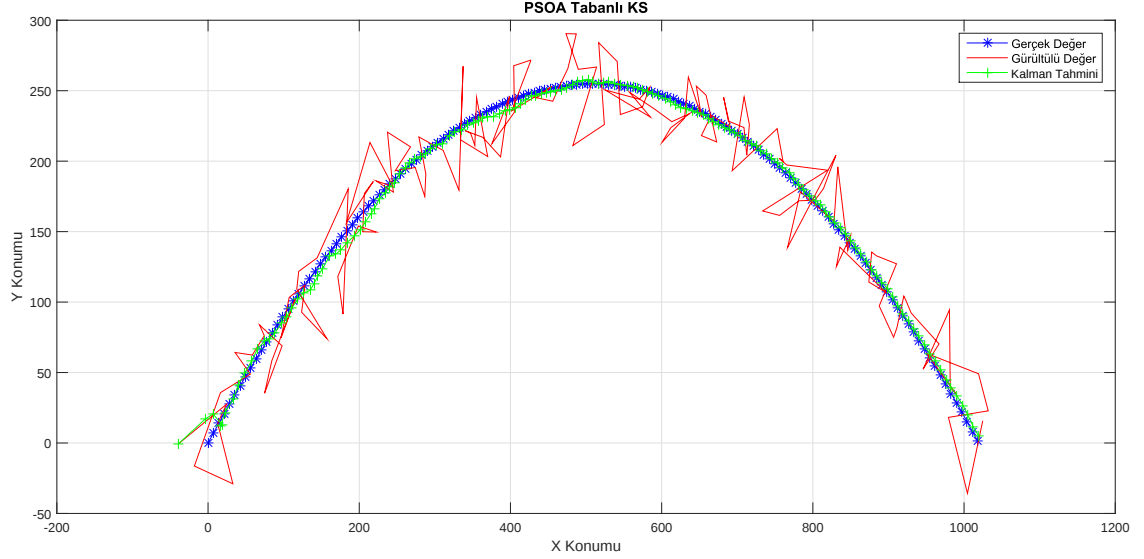


Şekil 3.23. Uygulama 2, DGA tabanlı KS için 30 koşmanın en iyisi

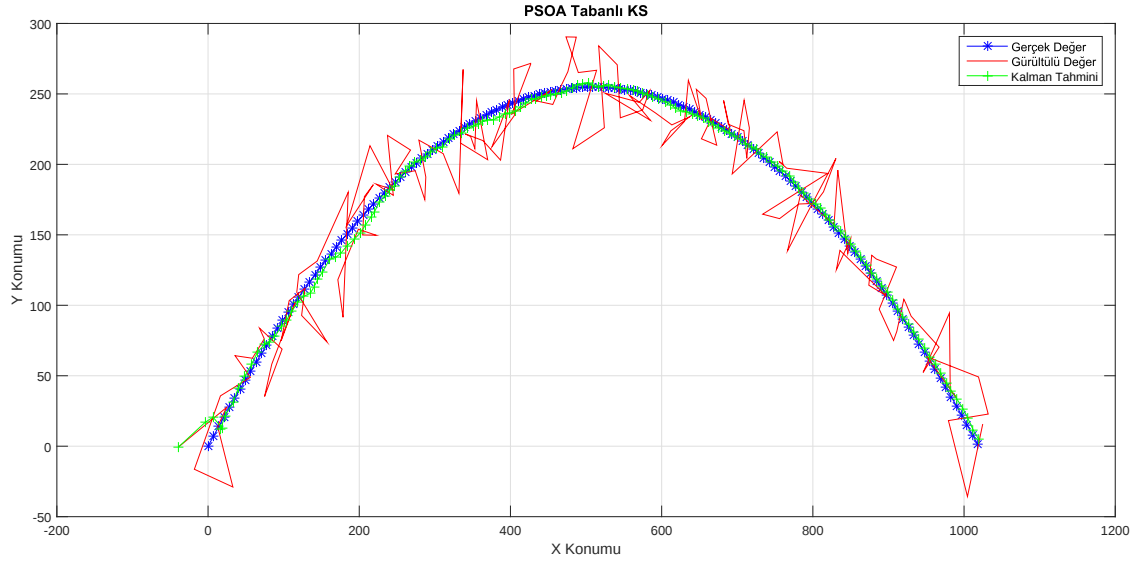


Şekil 3.24. Uygulama 2, DGA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 2 için yapılan 30 koşmalık PSOA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.25 ile, alınan en büyük hatalı sonuç Şekil 3.26 ile gösterilmiştir.

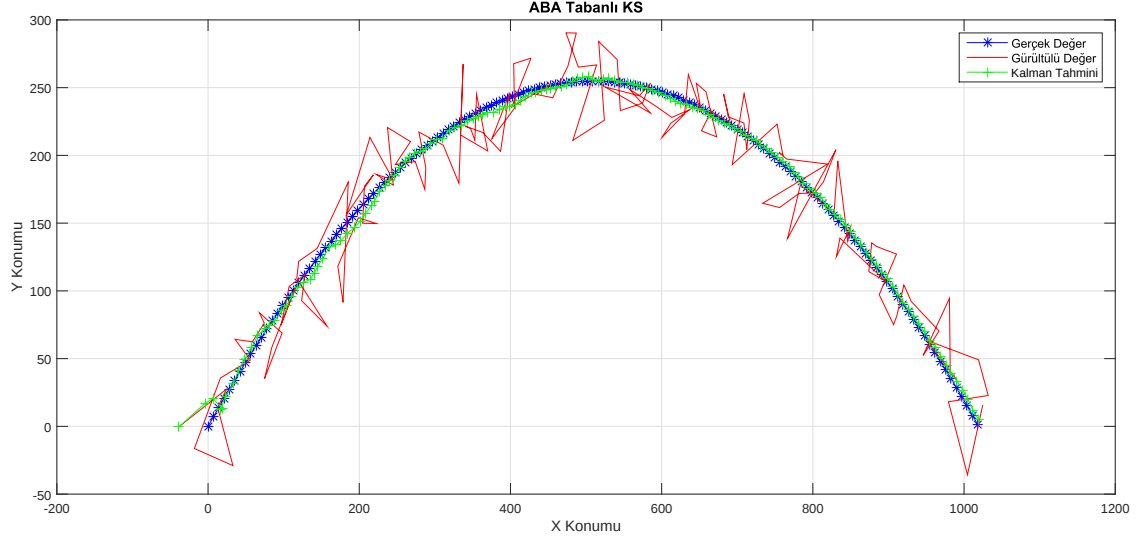


Şekil 3.25. Uygulama 2, PSOA tabanlı KS için 30 koşmanın en iyisi

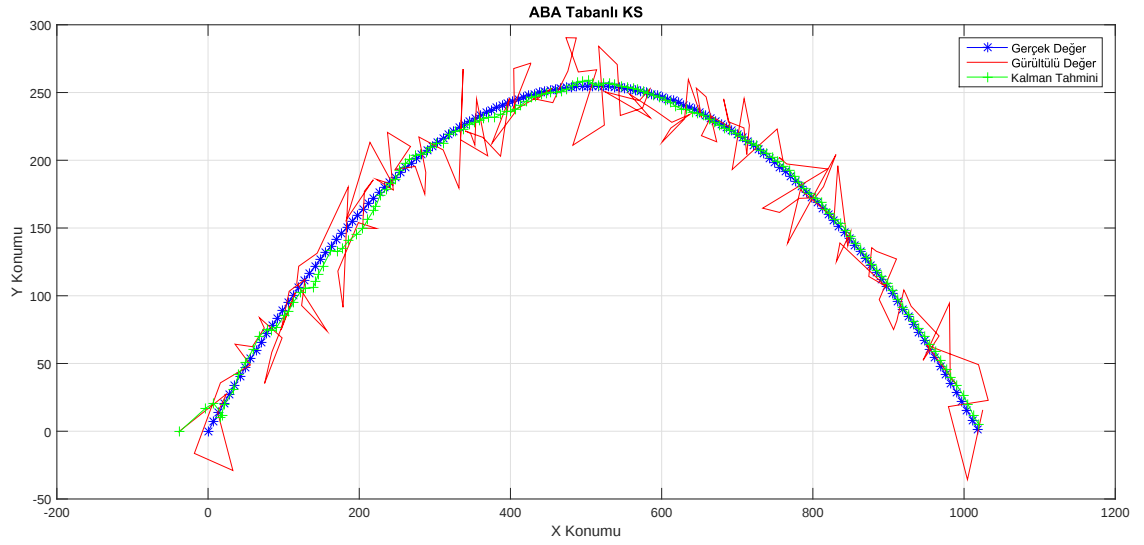


Şekil 3.26. Uygulama 2, PSOA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 2 için yapılan 30 koşmalık ABA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.27 ile, alınan en büyük hatalı sonuç Şekil 3.28 ile gösterilmiştir.



Şekil 3.27. Uygulama 2, ABA tabanlı KS için 30 koşmanın en iyisi



Şekil 3.28. Uygulama 2, ABA tabanlı KS için 30 koşmanın en kötüsü

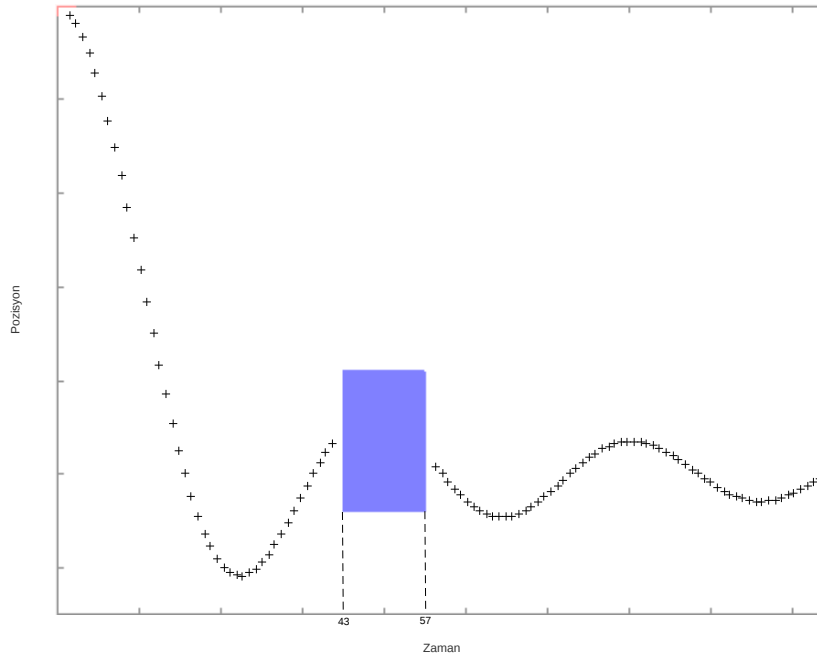
3.3. Uygulama 3

3.3.1. Problem Tanımı

İzdüşüm koordinatları '+' ile ifade edilen bir nesnenin 3 ayrı hareket güzergahı elde edilmiştir. 121 farklı çerçeveden oluşan toplu hareketlerde, nesne tünelin altından geçerken gözden kaybolduğu düşünülmektedir. KS ile hem nesnenin görüldüğü koordinatlar kestirim ve doğrulama ile belirlenecek hem de nesnenin görüldüğü durumlardaki koordinatlardan yola çıkarak gözden kaybolduğu durumlarda da konum kestirimi yapılacaktır.

3.3.1.1. Uygulama 3.1 İçin Problem Tanımı

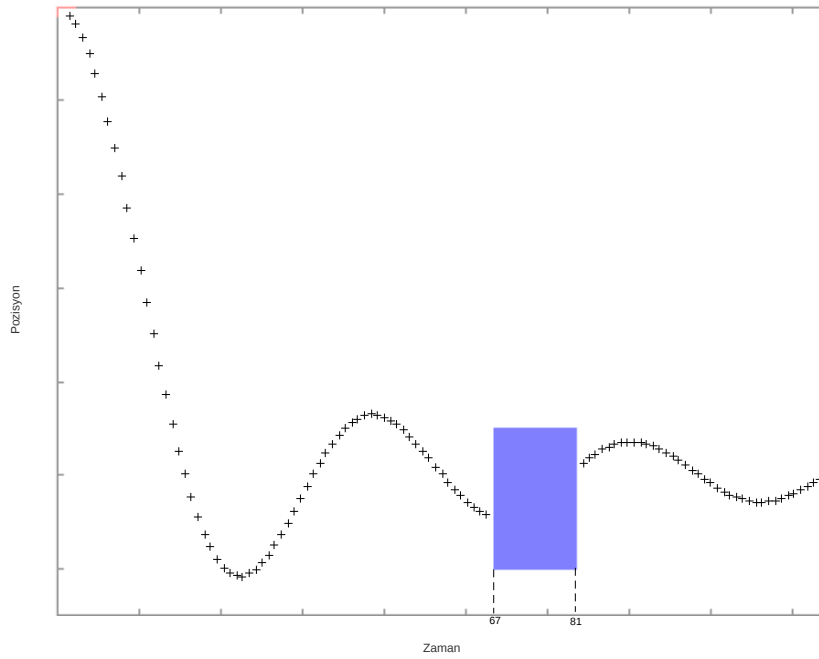
Rastgele salınımla hareket eden nesne, Uygulama 3.1 de daha yakında bulunan bir tünelin altından geçecektir. Bu uygulamada, nesnenin 43. ve 57. çerçeveler arasında tünelde kaybolduğu kabul edilmektedir.



Şekil 3.29. Uygulama 3.1, tünel konumu 43-57. çerçeveler

3.3.1.2. Uygulama 3.2 İçin Problem Tanımı

Uygulama 3.2’de Uygulama 3.1’e göre daha uzağa yerleştirilen tünel kullanılmıştır. Bu versiyonda, nesnenin 67. ve 81. çerçeveler arasında tünelde kaybolduğu kabul edilmektedir.



Şekil 3.30. Uygulama 3.2, tünel konumu 67-81. çerçeveler

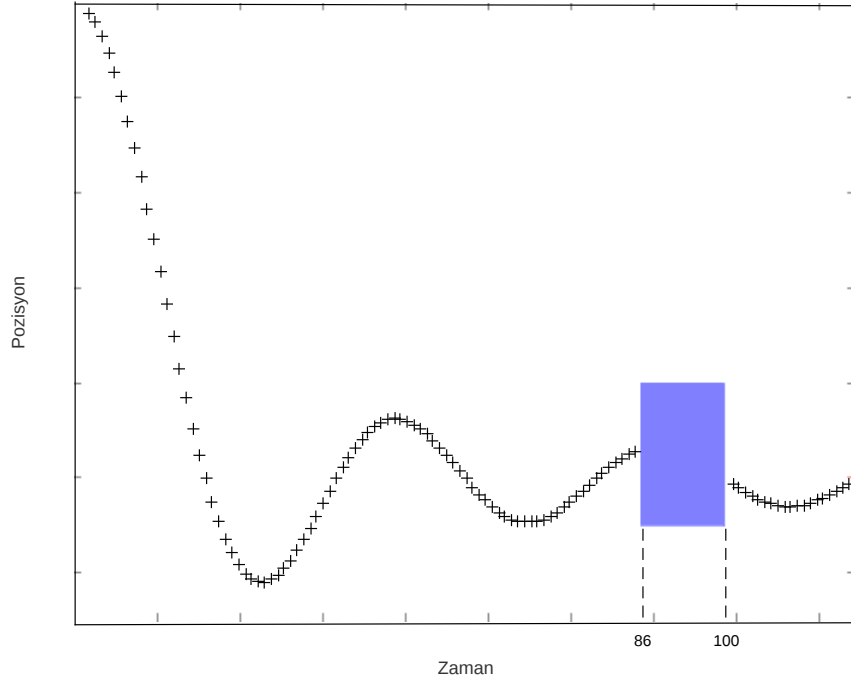
3.3.1.3. Uygulama 3.3 İçin Problem Tanımı

Uygulama 3.3’te tünel daha uzak bir konuma yerleştirilmiştir. Bu uygulamada, Şekil 3.31’de görüldüğü gibi nesne 86. ve 100. çerçeveler arasında tünelde kaybolmaktadır.

3.3.2. Uygulama 3 İçin Uygunluk Fonksiyonu Seçimi

Uygulama 3 için belirlenen uygunluk fonksiyonu Eşitlik 3.28 ile ifade edilmiştir.

$$MSE = \frac{1}{n} \sum_{i=1}^n (a_x^2 + a_y^2 + b_x^2 + b_y^2) \quad (3.28)$$



Şekil 3.31. Uygulama 3.3, tünel konumu 86-100. çerçeveler

Bu eşitlikte n değeri, nesne takibi uygulamasındaki toplam çerçeve sayısını ifade etmektedir. Top tünelde kaybolmadan önceki son iki x koordinat değeri farkıyla, top kaybolduktan sonraki ilk iki x koordinat değerleri farkının toplamının yarısına eşittir. Burada a_x değeri, topun görünen durumlarında tespit edilen x koordinat değeriyle kalman tahmini değerinin farkına, a_y değeri, topun görünen durumlarında tespit edilen y koordinat değeriyle kalman tahmini değerinin farkına, b_x değeri, topun tünelde kaybolduktan sonraki x konumu değerleriyle bir önceki konumlarının farkına, b_y değeri, topun tünelde kaybolduktan sonraki y konumu değerleriyle bir önceki konumlarının farkına eşittir.

3.3.3. Algoritma Temel Parametrelerinin Seçimi

Bu bölümde uygulama 3 için, KS parametrelerinin eniyilenmesinde kullanılacak olan güncel zeki optimizasyon algoritmalarının temel parametrelerinin seçimi yapılmıştır. Farklı kontrol parametreleri seçimi yapılarak 10'ar adet koşma yapıp ortalamaları alınarak en iyi sonucu veren kontrol parametreleri aşağıdaki gibi belirlenmiştir.

3.3.3.1. GA Temel Parametrelerinin Seçimi

Tablo 3.19. GA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Çaprazlama Sabiti	Mutasyon Oranı
200	100	0.8	0.2

GA temel parametrelerinin seçimi Tablo 3.19'teki gibi gerçekleştirilmiştir. Diğer algoritmalarla birlikte yapılan 10'ar koşmanın ardından algoritmalar için de ortak olarak kullanılacak olan popülasyon büyüklüğünün 200, iterasyon sayısının 100 seçilmesi gerektiği saptanmıştır. Yardımlaşma işlemi parametresi olan çaprazlama sabiti 0.8, değişim işlemi parametresi olan mutasyon oranı 0.2 olarak kabul edilmiştir.

3.3.3.2. YAKA Temel Parametrelerinin Seçimi

Tablo 3.20. YAKA Temel Parametreleri ve Değerleri

Koloni Büyüklüğü	<i>limit</i>	İterasyon Sayısı	Yiyecek Kaynağı
200	100	100	100

YAKA temel parametrelerinin seçimi Tablo 3.20'deki gibi gerçekleştirilmiştir. Koloni büyüklüğü 200 olarak alınmıştır. Başlangıç hata kovaryansı P , işlem hata kovaryansı Q ve ölçüm hata kovaryansı parametreleri R_1 , R_2 , R_3 optimize edileceği için optimize edilecek parametre sayısı 5'tir. 100 seçilen 'limit' sayısı kadar denemeden sonra gelişim sağlanamaması halinde işçi arılar yiyecek kaynağını değiştirir. Döngü sayısı bu algoritma için de 100 olarak belirlenmiştir. Son YAKA kontrol parametresi olan yiyecek kaynağı sayısı, koloni sayısının yarısı kadardır ve 100 olarak belirlenmiştir.

3.3.3.3. DGA Temel Parametrelerinin Seçimi

Tablo 3.21. DGA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Çaprazlama Sabiti	Adım Boyutu
200	100	0.8	1.75

DGA temel parametrelerinin seçimi Tablo 3.21'deki gibi gerçekleştirilmiştir. Bu algoritma için diğer algoritmalarda olduğu gibi popülasyon büyüklüğü 200 ve iterasyon sayısı 100 olarak belirlenmiştir. Diğer kontrol parametrelerinden, çaprazlama sabiti

0.8 ölçekleme faktörü diğer adıyla adım büyüklüğü parametre değeri de 1.75 olarak belirlenmiştir. Diğer uygulamalarda olduğu gibi bu uygulamada da *DGA/best/1* algoritma stratejisi kullanılmıştır.

3.3.3.4. PSOA Temel Parametrelerinin Seçimi

Tablo 3.22. PSOA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Parçacık Boyutu	V_{max}	C_1, C_2	w
200	100	5	[0.99 0.99 0.99 0.99 0.99]	1.49	1.1

PSOA temel parametrelerinin seçimi Tablo 3.22'deki gibi gerçekleştirilmiştir. Popülasyon büyüklüğü ve iterasyon sayısı diğer algoritmalarda olduğu gibi bu algoritma için de 200 ve 100 olarak belirlenmiştir. Bir iterasyondaki bir parçacıkta meydana gelebilecek maksimum değişikliği belirten V_{max} [0.99 0.99 0.99 0.99 0.99] değerini, öğrenme faktörleri C_1, C_2 1.49 değerini ve eylemsizlik ağırlığı w 1.1 değerini almıştır.

3.3.3.5. ABA Temel Parametrelerinin Seçimi

Tablo 3.23. ABA Temel Parametreleri ve Değerleri

Popülasyon Büyüklüğü	İterasyon	Alpha	Betamin	Gamma
200	100	0.5	0.2	1

ABA temel parametrelerinin seçimi Tablo 3.23'deki gibi gerçekleştirilmiştir. Popülasyon sayısı 200, iterasyon sayısı 100, adım büyüklüğü Alpha 0.5, ateş böceklerinin çekicilik oranı Betamin 0.2 ve ateş böceklerinin arasındaki çekicilik değişimini belirleyen parametre Gamma 1 olarak seçilmiştir.

3.3.4. Optimize Edilmiş KS'nin Uygulama Sonuçları

Bu uygulamada, popüler bir konu olan nesne takibi uygulamasında gürültülü ortamlarda nesnenin görüldüğü durumlarda takibi, gözden kaybolduğu durumlarda konum kestirimi yapılmıştır. Bu işlemlerin gerçekleştirilmesi KS ile sağlanmıştır. KS parametrelerinden başlangıç hata kovaryansı P , ölçüm hata kovaryansı parametreleri R_1, R_2, R_3 ve işlem hata kovaryansı Q değerleri öncelikle kullanıcı deneyimine bağlı olarak daha

sonra da güncel zeki optimizasyon algoritmalarıyla belirlenmiştir. Nesne takibinde aynı görüntü kullanılıp, görüntüdeki tünelin 3 farklı konuma konulmasıyla 3 ayrı uygulama uyarlanmıştır.

3.3.4.1. Uygulama 3.1 Sonucu

Uygulama 3.1’de kullanıcı deneyimine bağlı olarak Tablo 3.24 ile gösterilen 3 farklı parametre seçimi gerçekleştirilmiştir. Bu parametre seçimine bağlı olarak oluşan sonuç grafikleri Şekil 3.33, 3.34 ve 3.35 ile gösterilmektedir. Bu uygulamada da KS parametreleri için kullanılan güncel zeki optimizasyon algoritmalarının 30 koşmanın en iyi ve en kötü sonucuna göre aldığı parametreler Tablo 3.24 ile gösterilmiştir. Her algoritma için koşulan 30 iterasyon sonucu ortaya çıkan hata dağılımları Tablo 3.25 ile verilmiştir.

Tablo 3.24. Uygulama 3.1 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafında seçilen KS parametreleri

Algoritma	P	Q	R1	R2	R3
Standart-KS (V1)	10	10	10	10	10
Standart-KS (V2)	5	0.1	1	50	100
Standart-KS (V3)	1	0.5	1	3	0.1
GA-KS (En iyi)	0.9998	0.4443	0.00036713	0.00019997	0.1699
GA-KS (En kötü)	1	0.7860	0.0010	0.0013	0.2982
YAKA-KS (En iyi)	1	0.0869	0.0001	0.0001	0.0324
YAKA-KS (En kötü)	1	0.4229	0.0001	0.0001	0.1564
DGA-KS (En iyi)	1	0.0760	0.0001	0.0001	0.0288
DGA-KS (En kötü)	1	0.0766	0.0001	0.0001	0.0290
PSOA-KS (En iyi)	1	0.0759	0.0001	0.0001	0.0287
PSOA-KS (En kötü)	1	0.0775	0.0001	0.0001	0.0293
ABA-KS (En iyi)	0.7063	0.3047	0.00001	0.00001	0.1155
ABA-KS (En kötü)	0.2946	0.5291	0.0390	0.00001	0.2160

Tablo 3.25 incelendiğinde, standart sapması en küçük olan DGA, kararlılığı en yüksek olan algoritmadır. 1.842224E-01 değeri ile PSOA en düşük MSE değerine sahiptir. Diğer performans kriteri olan işlem süresine göre YAKA, algoritmalar içerisinde en hızlı çalışan algoritmadır.

Uygulama 3.1 için yapılan istatistiksel analizde, Tablo 3.26 ile Kruskal Wallis test sonucunda anlamlılık değerinin (p), 2.08E-25 değerine eşit olması bu algoritmalar

Tablo 3.25. Kullanılan algoritmaların uygulama 3.1 için hata dağılımları

Uygulama 3.1 İçin Algoritmaların Hata Dağılımı						
Algoritmalar	En İyi	En Kötü	Medyan	Ortalama	Standart Sapma	İşlem Süresi (sn)
GA	2.039614E-01	2.130149E-01	2.077478E-01	2.080965E-01	1.855015E-03	901
YAKA	1.973760E-01	2.034180E-01	2.002398E-01	2.002731E-01	1.687525E-03	893
DGA	1.972646E-01	1.972652E-01	1.972647E-01	1.972648E-01	1.454168E-07	899
PSOA	1.842223E-01	1.842231E-01	1.842223E-01	1.842224E-01	2.388085E-07	938
ABA	1.898145E-01	2.326505E-01	2.010627E-01	2.022137E-01	7.621251E-03	907

arasında anlamlı farkın olduğunu göstermektedir. Tablo 3.27 incelendiğinde Mann-Whitney U testinin tüm algoritmaların ikili kombinasyonları için uygulandığı ve bu algoritmaların birbirlerine göre anlamlı farka sahip olup olmadıkları gözlemlenmiştir.

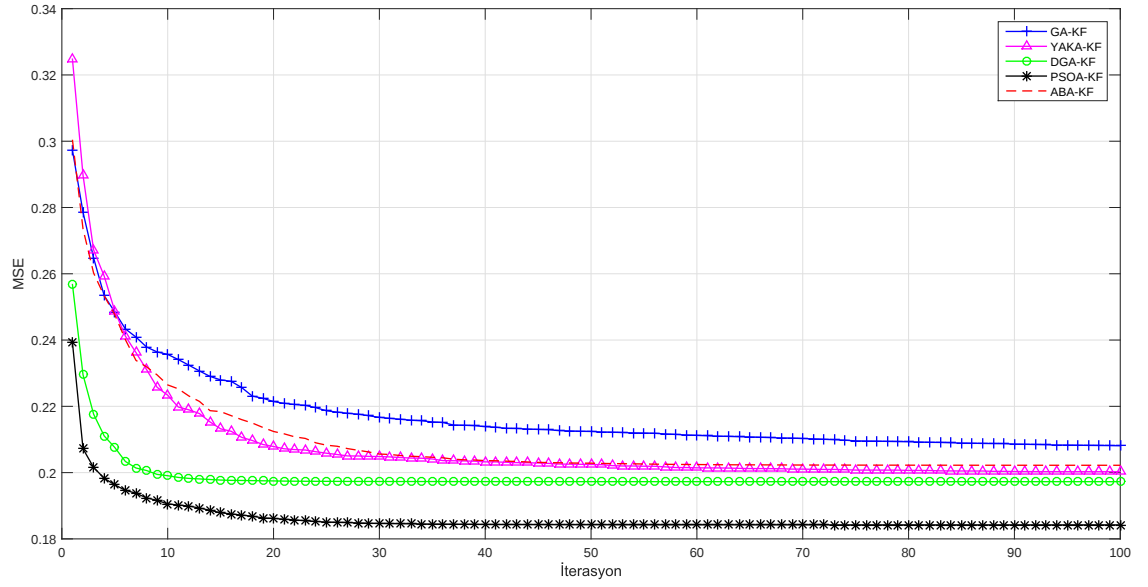
Tablo 3.26. Uygulama 3.1 için tüm algoritmalarda Kruskal Wallis test sonucu

Algoritma	N	df	Chi_Square	p
GA	30	4	121.919	2.08E-25
YAKA	30			
DGA	30			
PSOA	30			
ABA	30			

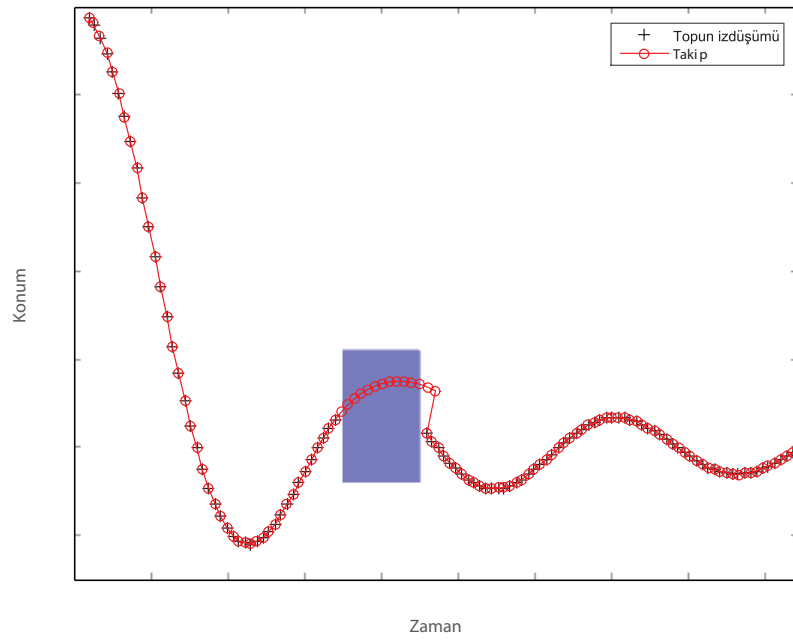
Tablo 3.27. Uygulama 3.1 için ikili algoritmalarda Mann-Whitney U test sonucu

Algoritmalar	Z	Ortalamaların Farkı	p	Anlamlı Fark
GA-YAKA	-6.653	7.82342167E-03	2.87E-11	YAKA Lehine
GA-DGA	-7.112	1.08317334E-02	1.14E-12	DGA Lehine
GA-PSOA	-6.909	2.38740699E-02	4.87E-12	PSOA Lehine
GA-ABA	-4.953	5.88278258E-03	7.32E-07	ABA Lehine
YAKA-DGA	-7.112	3.00831171E-03	1.14E-12	DGA Lehine
YAKA-PSOA	-6.909	1.60506483E-02	4.87E-12	PSOA Lehine
YAKA-ABA	-0.562	-1.94063909E-03	0.574247	Yok
DGA-PSOA	-7.428	1.30423366E-02	1.11E-13	PSOA Lehine
DGA-ABA	-3.319	-4.94895080E-03	0.000904	DGA Lehine
PSOA-ABA	-6.909	-1.79912873E-02	4.87E-12	PSOA Lehine

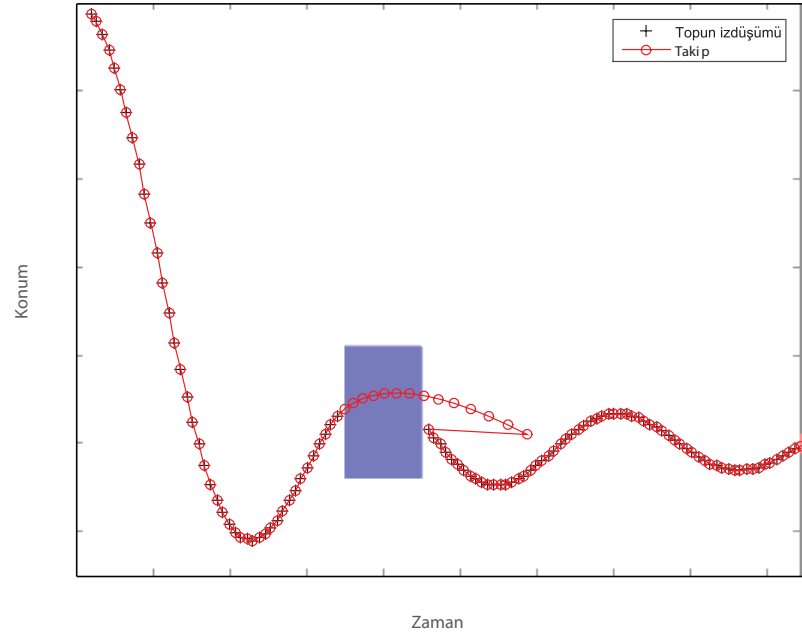
Uygulama 3.1 için yapılan kullanıcı deneyimi tabanlı KS çalışmasında, ortaya çıkan sonuçlar Şekil 3.33, 3.34 ve 3.35'te verilmiştir.



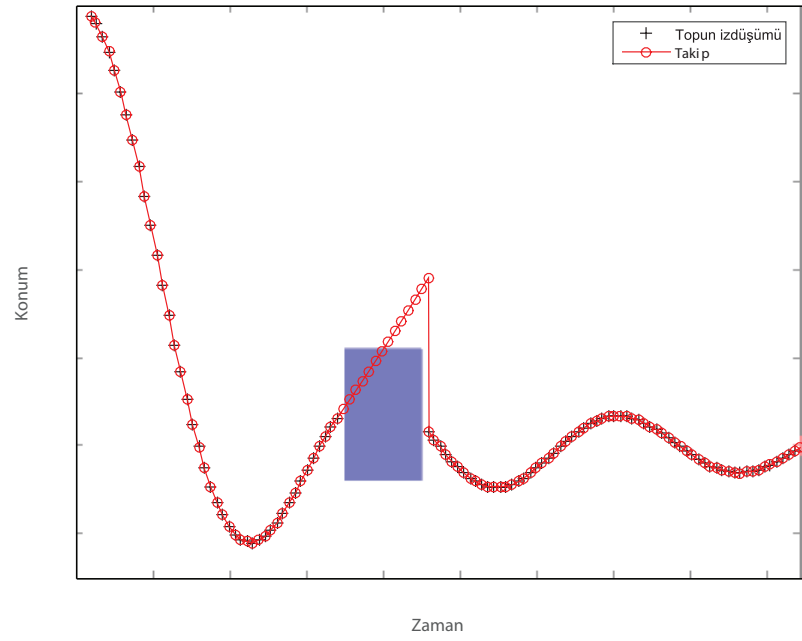
Şekil 3.32. Uygulama 3.1 için yöntemlerin iterasyon-MSE değişimi



Şekil 3.33. Uygulama 3.1, kullanıcı deneyimi tabanlı KS (D1)

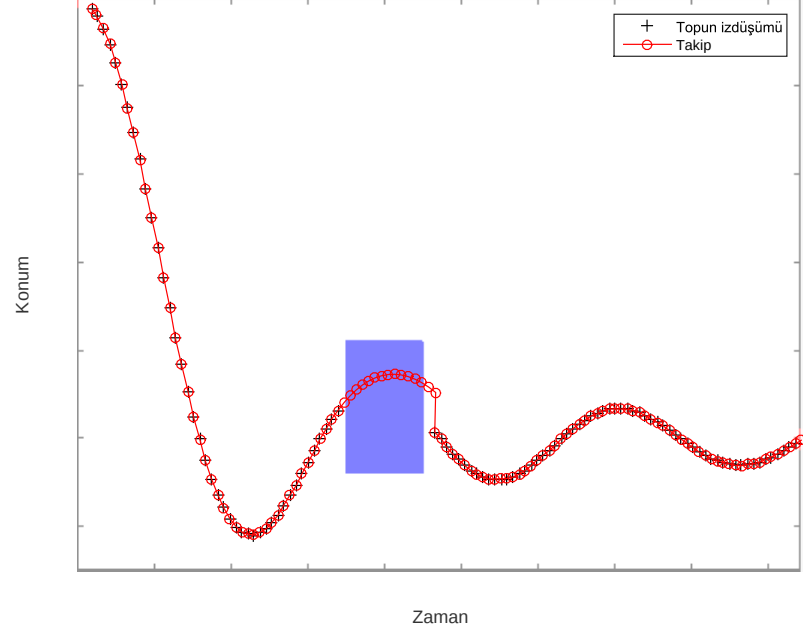


Şekil 3.34. Uygulama 3.1, kullanıcı deneyimi tabanlı KS (D2)

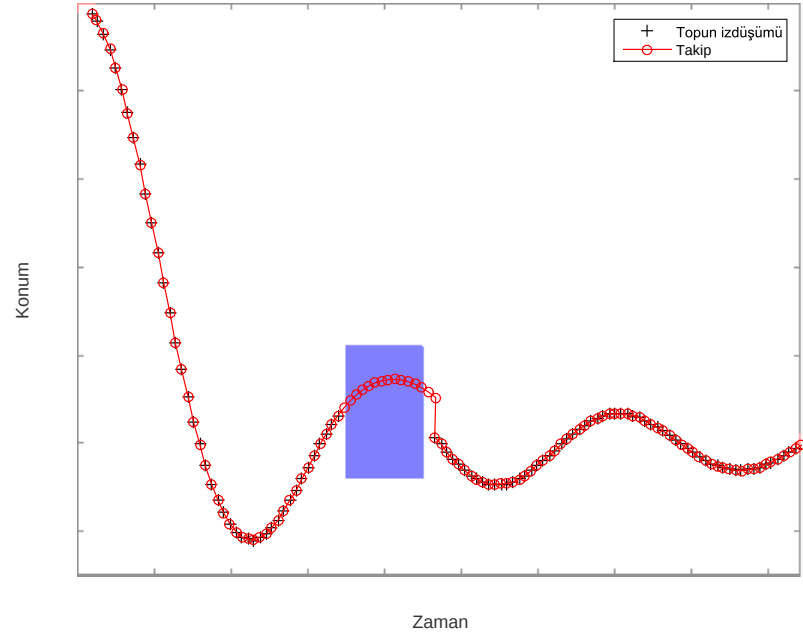


Şekil 3.35. Uygulama 3.1, kullanıcı deneyimi tabanlı KS (D3)

Uygulama 3.1 için yapılan 30 kořmalık GA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.36 ile, alınan en büyük hatalı sonuç Şekil 3.37 ile gösterilmiştir.

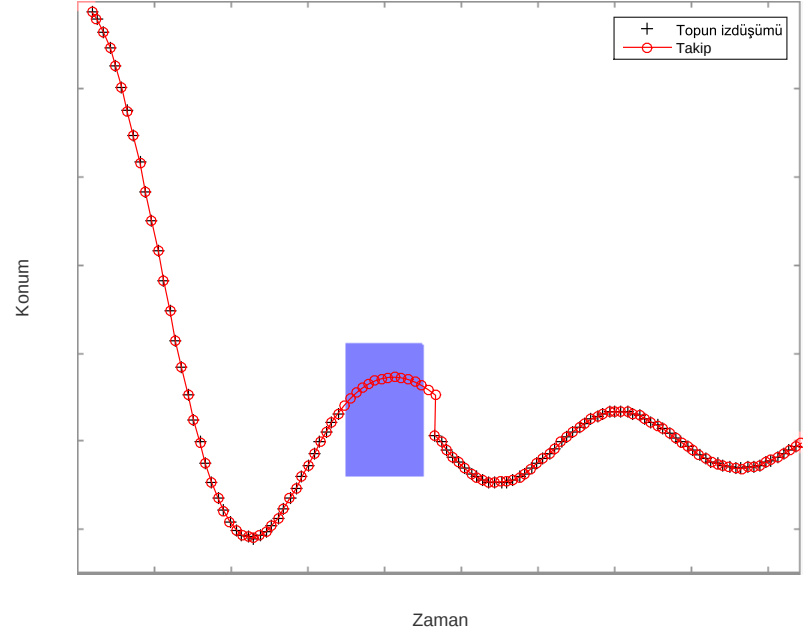


Şekil 3.36. Uygulama 3.1, GA tabanlı KS için 30 kořmanın en iyisi

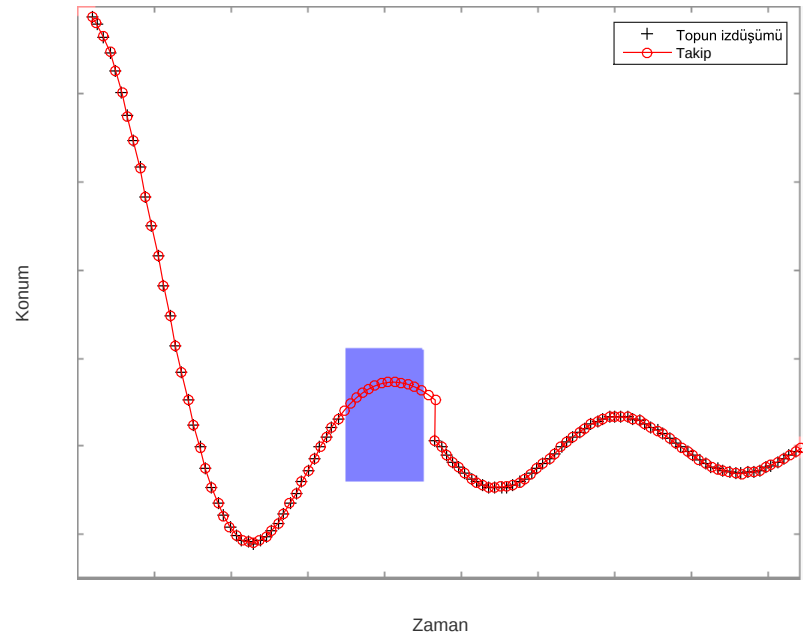


Şekil 3.37. Uygulama 3.1, GA tabanlı KS için 30 kořmanın en kötüsi

Uygulama 3.1 için yapılan 30 koşmalık YAKA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.38 ile, alınan en büyük hatalı sonuç Şekil 3.39 ile gösterilmiştir.

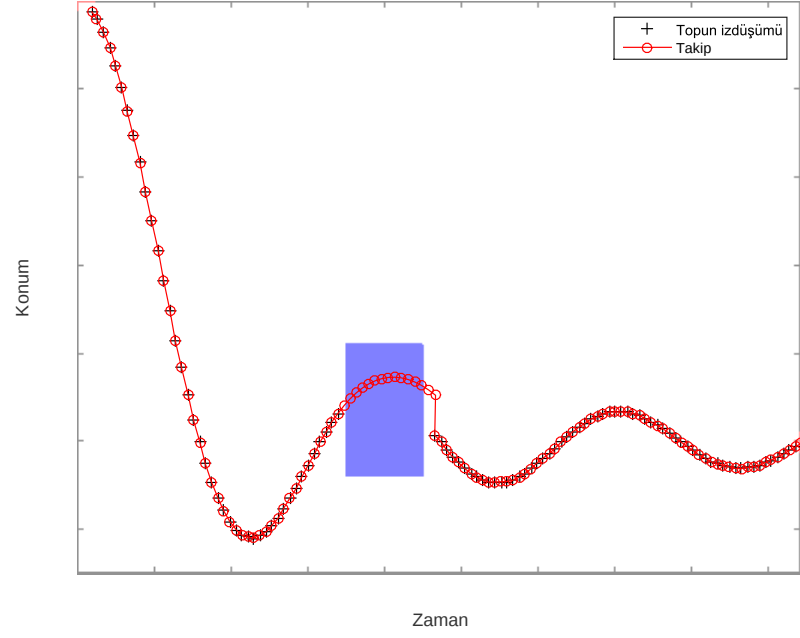


Şekil 3.38. Uygulama 3.1, YAKA tabanlı KS için 30 koşmanın en iyisi

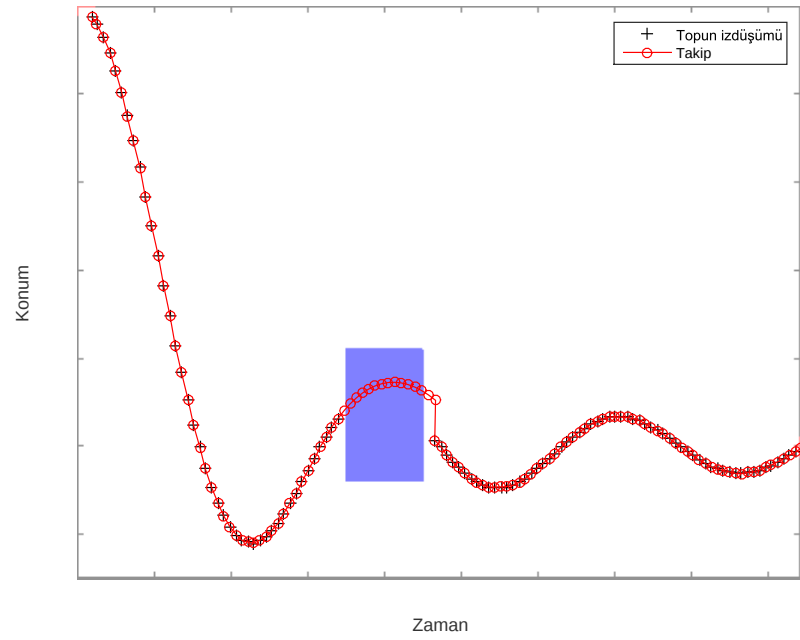


Şekil 3.39. Uygulama 3.1, YAKA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 3.1 için yapılan 30 koşmalık DGA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.40 ile, alınan en büyük hatalı sonuç Şekil 3.41 ile gösterilmiştir.

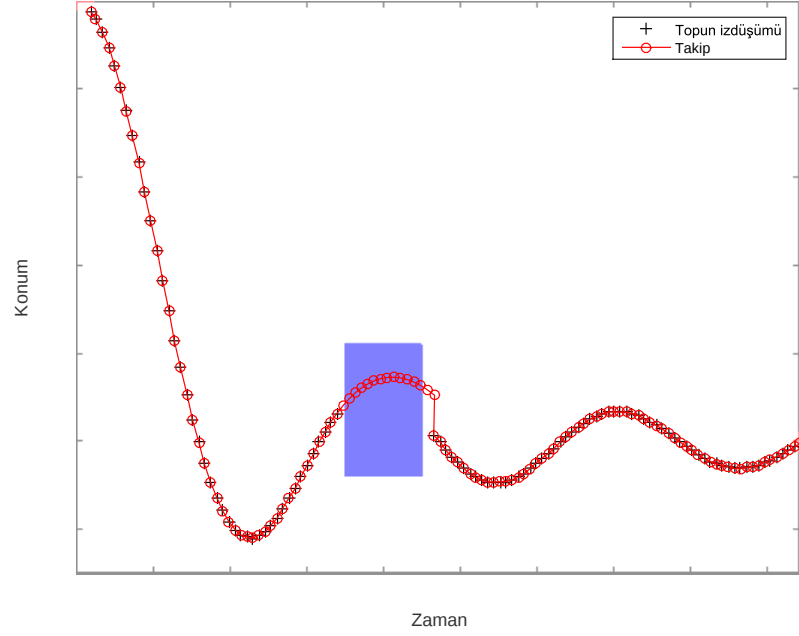


Şekil 3.40. Uygulama 3.1, DGA tabanlı KS için 30 koşmanın en iyisi

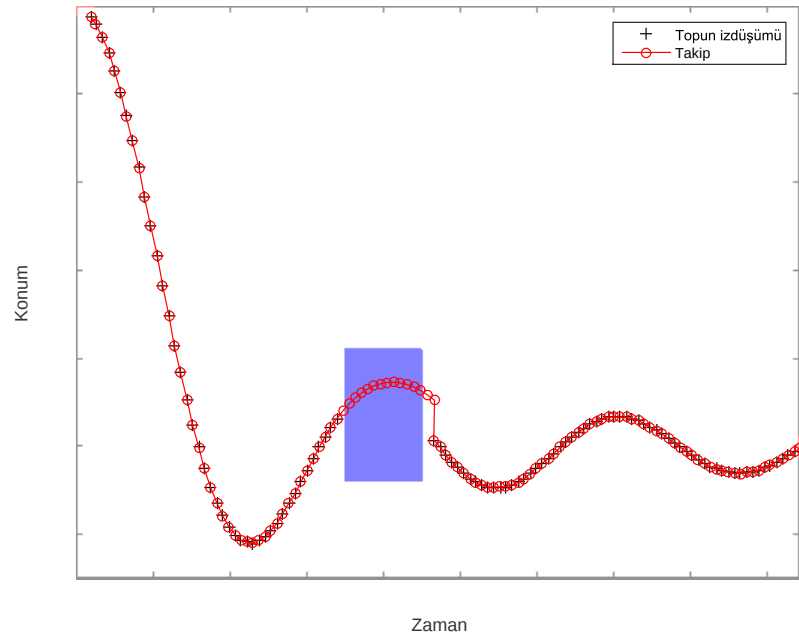


Şekil 3.41. Uygulama 3.1, DGA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 3.1 için yapılan 30 koşmalık PSOA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.42 ile, alınan en büyük hatalı sonuç Şekil 3.43 ile gösterilmiştir.

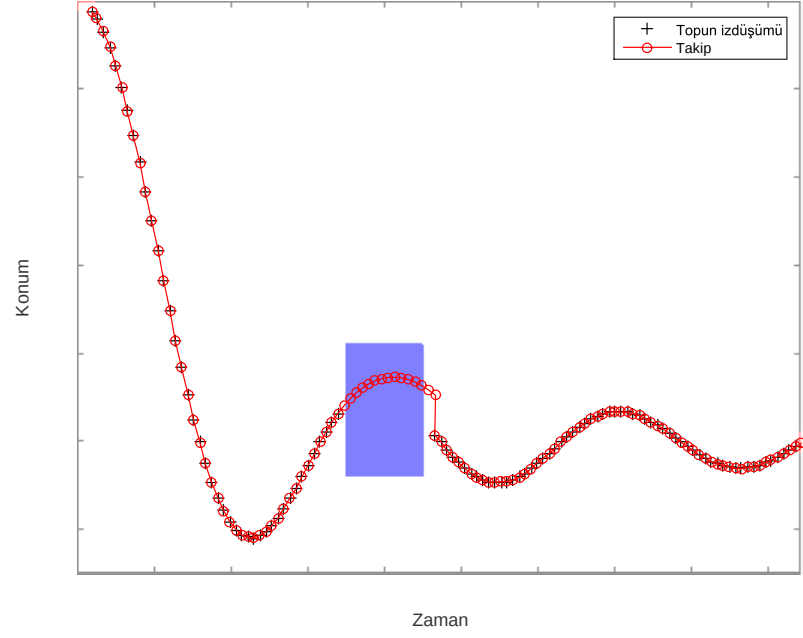


Şekil 3.42. Uygulama 3.1, PSOA tabanlı KS için 30 koşmanın en iyisi

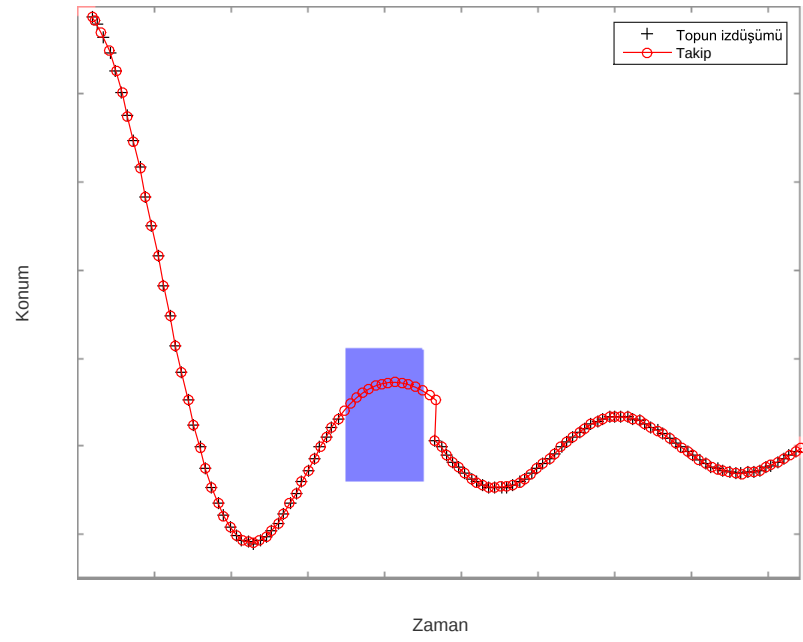


Şekil 3.43. Uygulama 3.1, PSOA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 3.1 için yapılan 30 koşmalık ABA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.44 ile, alınan en büyük hatalı sonuç Şekil 3.45 ile gösterilmiştir.



Şekil 3.44. Uygulama 3.1, ABA tabanlı KS için 30 koşmanın en iyisi



Şekil 3.45. Uygulama 3.1, ABA tabanlı KS için 30 koşmanın en kötüsü

3.3.4.2. Uygulama 3.2 Sonucu

Uygulama 3.2’de kullanıcı deneyimine bağlı olarak Tablo 3.28 ile gösterilen 3 farklı parametre seçimi gerçekleştirilmiştir. Bu parametre seçimine bağlı olarak oluşan sonuç grafikleri Şekil 3.47, 3.48 ve 3.49 ile verilmektedir. Bu uygulamada KS parametreleri için kullanılan güncel zeki optimizasyon algoritmalarının 30 koşmanın en iyi ve en kötü sonucuna göre aldığı parametreler Tablo 3.28 ile gösterilmiştir. Her algoritma için koşulan 30 iterasyon sonucu ortaya çıkan hata dağılımları Tablo 3.29 ile verilmiştir.

Tablo 3.28. Uygulama 3.2 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafında seçilen KS parametreleri

Algoritma	P	Q	R1	R2	R3
Standart-KS (V1)	0.5	0.5	0.5	0.5	0.5
Standart-KS (V2)	5	1	0.1	0.3	0.5
Standart-KS (V3)	0.9	0.1	1	5	1
GA-KS (En iyi)	0.9842	0.0010	0.0241	0.0018	0.9997
GA-KS (En kötü)	0.0703	0.9998	0.5450	0.0014	0.9985
YAKA-KS (En iyi)	0.9873	0.0001	0.0241	0.0001	1
YAKA-KS (En kötü)	0.0408	0.7610	0.3346	0.0001	1
DGA-KS (En iyi)	1	0.0001	0.0244	0.0001001	0.9967
DGA-KS (En kötü)	0.0566	0.9999	0.4479	0.00010008	0.9852
PSOA-KS (En iyi)	1	0.0001	0.0244	0.0001	1
PSOA-KS (En kötü)	0.0563	1	0.4473	0.0001	1
ABA-KS (En iyi)	0.0547	0.9686	0.4319	0.00001	0.5994
ABA-KS (En kötü)	0.2100	0.4286	0.6924	0.00001	0.9717

Tablo 3.29 incelendiğinde standart sapması en küçük olan PSOA, kararlılığı en yüksek olan algoritmadır. 7.633934E-03 değeri ile DGA en düşük MSE değerine sahiptir. İşlem süresi kriterine göre ABA, bu uygulama için çalışan en hızlı algoritmadır.

Tablo 3.29. Kullanılan algoritmaların uygulama 3.2 için hata dağılımı

Uygulama 3.2 İçin Algoritmaların Hata Dağılımı						
Algoritmalar	En İyi	En Kötü	Medyan	Ortalama	Standart Sapma	İşlem Süresi (sn)
GA	7.023915E-03	9.098754E-03	8.615219E-03	8.255288E-03	7.109544E-04	895
YAKA	6.911021E-03	8.482940E-03	7.982581E-03	7.781557E-03	7.093041E-04	904
DGA	6.910793E-03	8.460875E-03	6.910799E-03	7.633934E-03	7.862849E-04	893
PSOA	6.910793E-03	8.460399E-03	8.460237E-03	8.305298E-03	4.727814E-04	903
ABA	7.338314E-03	4.809875E-02	1.079112E-02	1.498578E-02	1.011302E-02	877

Uygulama 3.2 için yapılan istatistiksel analizde, Tablo 3.30 ile Kruskal Wallis test sonucunda anlamlılık değerinin (p), 4.81E-12 değerine eşit olması bu algoritmalar arasında anlamlı farkın olduğunu göstermektedir. Tablo 3.31 incelendiğinde Mann-Whitney U testinin tüm algoritmaların ikili kombinasyonları için uygulandığı ve bu algoritmaların birbirlerine göre anlamlı farka sahip olup olmadıkları gözlemlenmiştir.

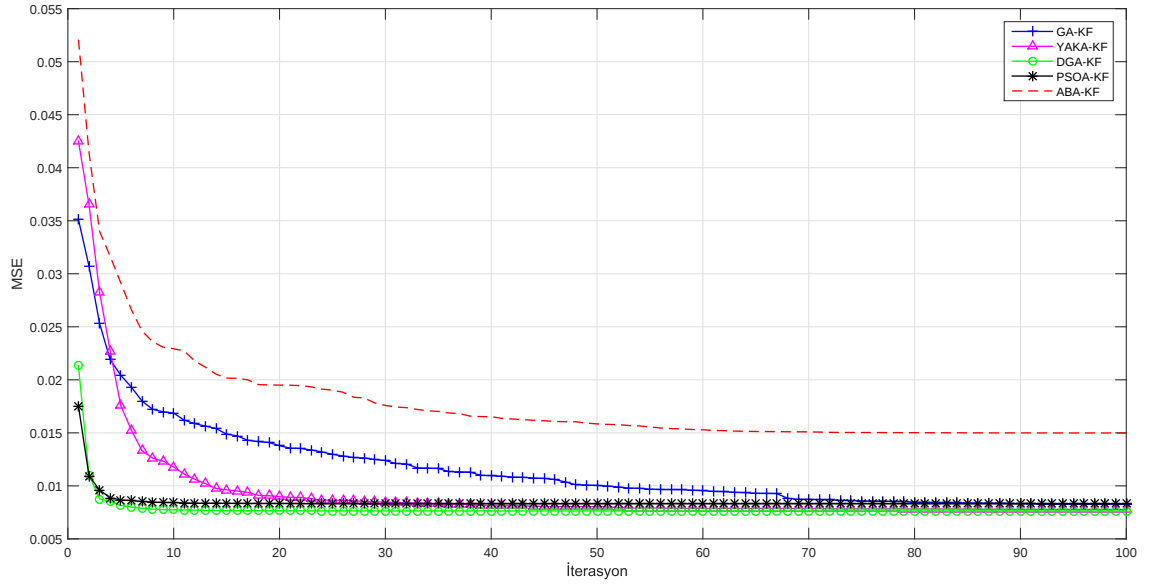
Tablo 3.30. Uygulama 3.2 için tüm algoritmalarda Kruskal Wallis test sonucu

Algoritma	N	df	Chi_Square	p
GA	30	4	58.955	4.81E-12
YAKA	30			
DGA	30			
PSOA	30			
ABA	30			

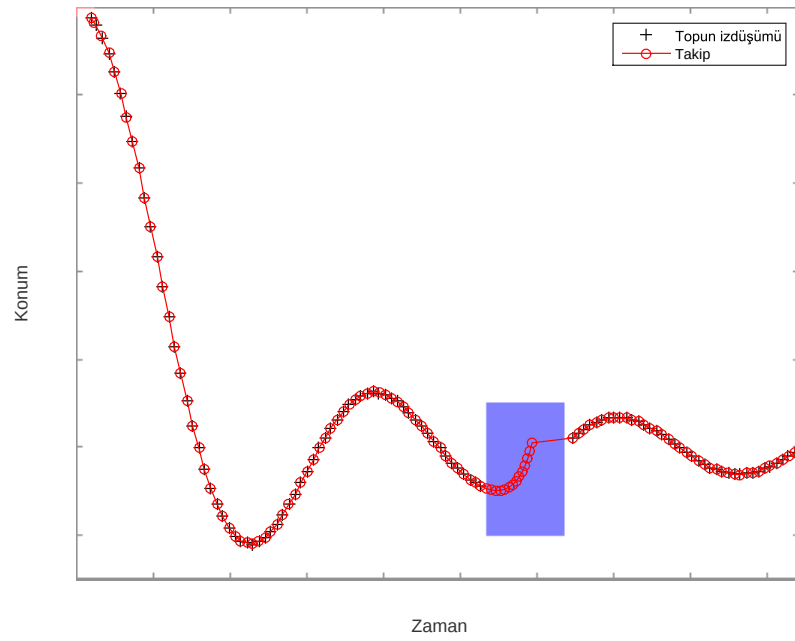
Tablo 3.31. Uygulama 3.2 için ikili algoritmalarda Mann-Whitney U test sonucu

Algoritmalar	Z	Ortalamaların Farkı	p	Anlamlı Fark
GA-YAKA	-4.317	4.73730522E-04	0.000016	YAKA Lehine
GA-DGA	-5.037	6.21353469E-04	4.74E-07	DGA Lehine
GA-PSOA	-3.61	-5.00105352E-05	0.000307	GA Lehine
GA-ABA	-3.681	-6.73049129E-03	0.000232	GA Lehine
YAKA-DGA	-3.368	1.47622948E-04	0.000758	DGA Lehine
YAKA-PSOA	-0.278	-5.23741057E-04	0.781	Yok
YAKA-ABA	-5.293	-7.20422182E-03	1.20E-07	YAKA Lehine
DGA-PSOA	-0.441	-6.71364005E-04	0.659	Yok
DGA-ABA	-5.454	-7.35184476E-03	4.93E-08	DGA Lehine
PSOA-ABA	-4.443	-6.68048076E-03	0.000009	PSOA Lehine

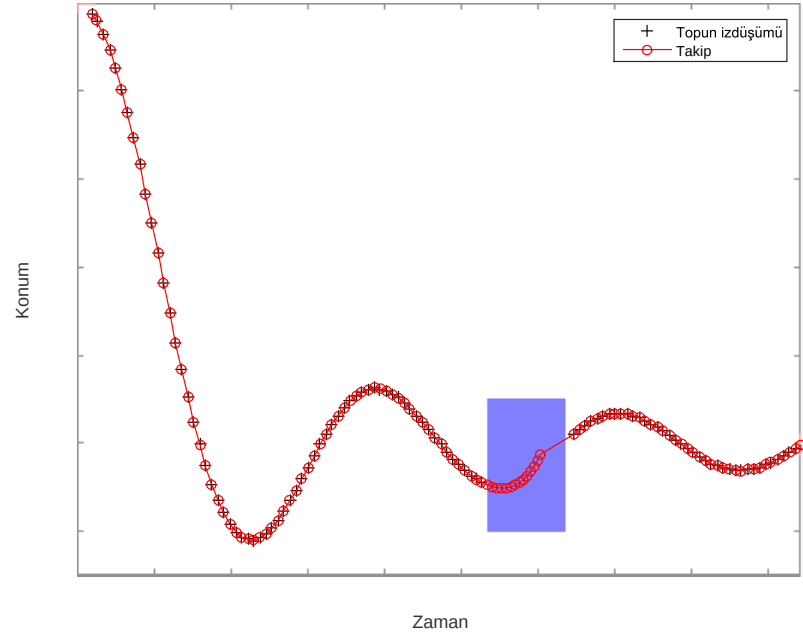
Uygulama 3.2 için yapılan kullanıcı deneyimi tabanlı KS çalışmasında, ortaya çıkan sonuçlar Şekil 3.47, 3.48 ve 3.49’da verilmiştir.



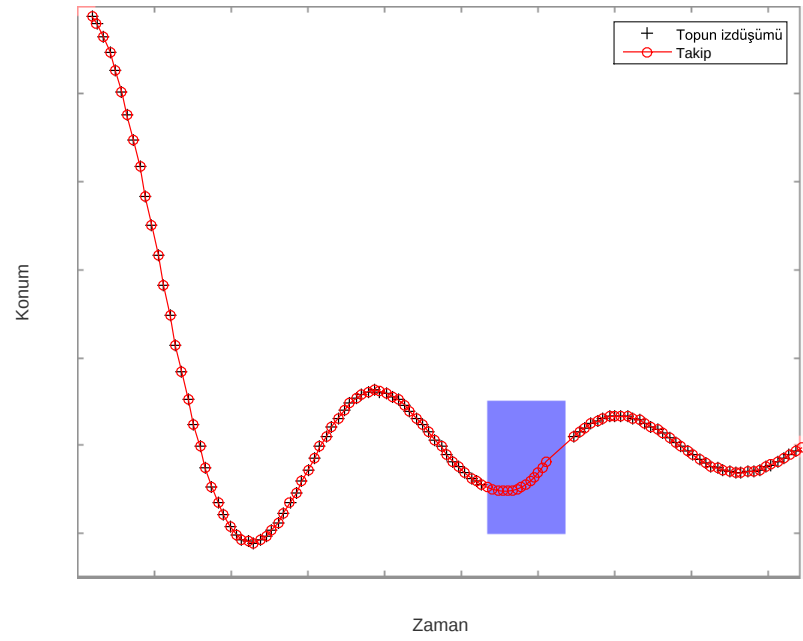
Şekil 3.46. Uygulama 3.2 için yöntemlerin iterasyon-MSE değişimi



Şekil 3.47. Uygulama 3.2, kullanıcı deneyimi tabanlı KS (D1)

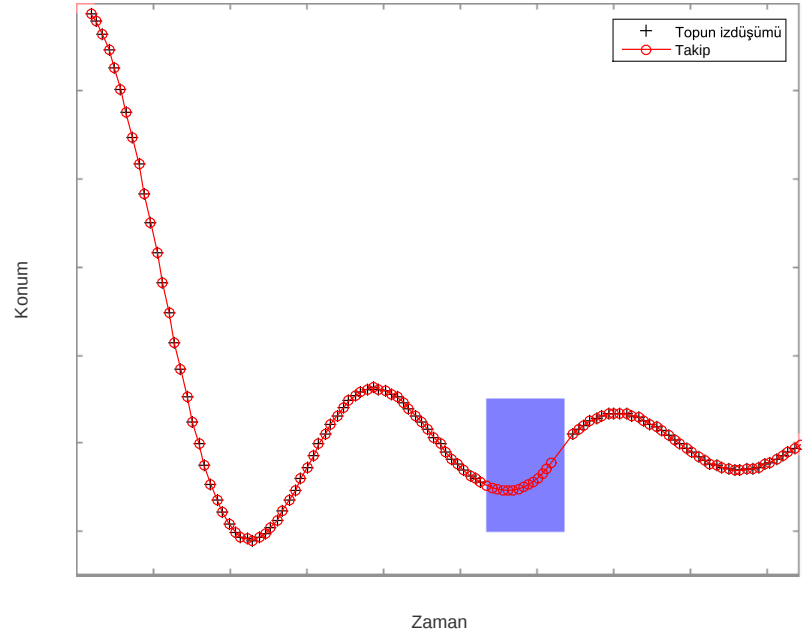


Şekil 3.48. Uygulama 3.2, kullanıcı deneyimi tabanlı KS (D2)

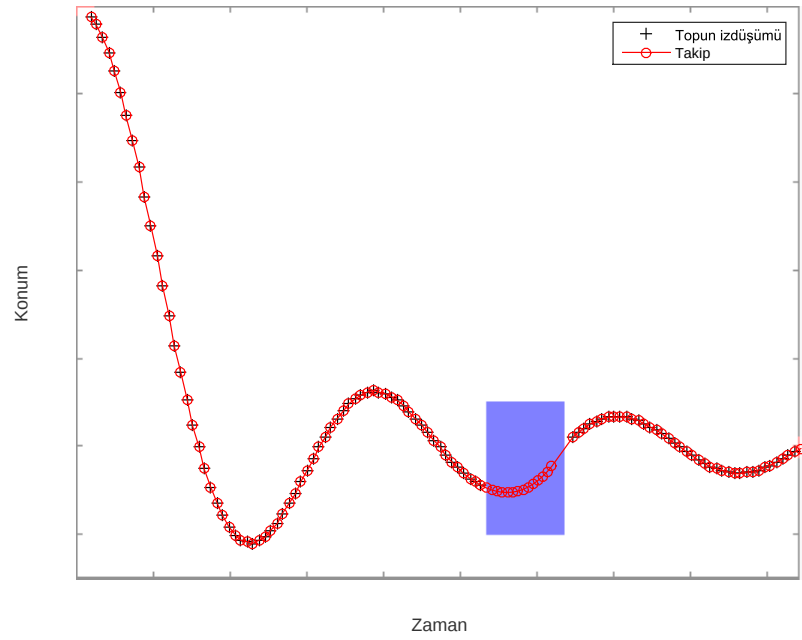


Şekil 3.49. Uygulama 3.2, kullanıcı deneyimi tabanlı KS (D3)

Uygulama 3.2 için yapılan 30 kořmalık GA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.50 ile, alınan en büyük hatalı sonuç Şekil 3.51 ile gösterilmiştir.

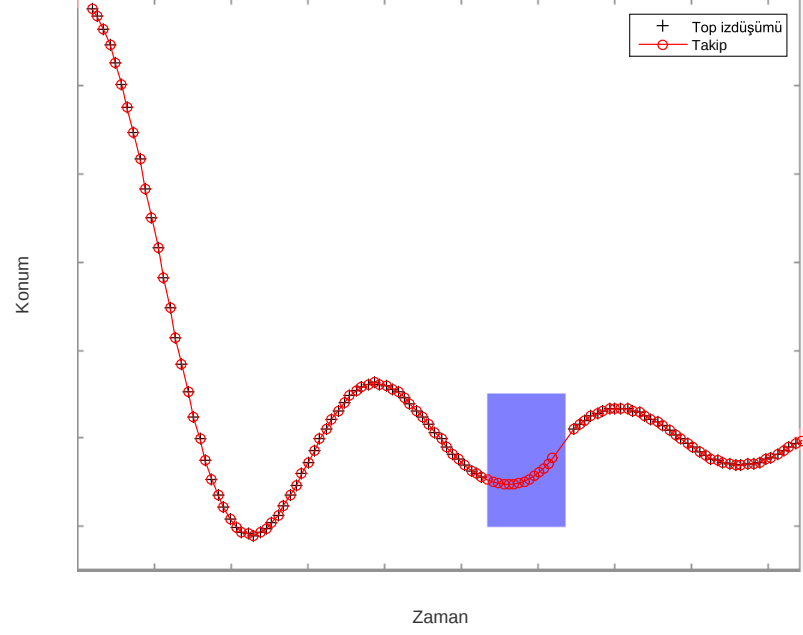


Şekil 3.50. Uygulama 3.2, GA tabanlı KS için 30 kořmanın en iyisi

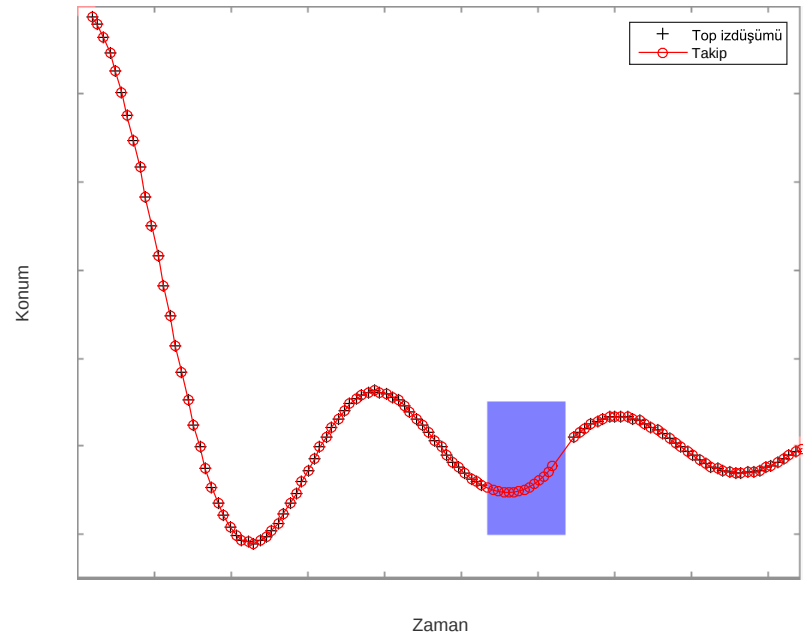


Şekil 3.51. Uygulama 3.2, GA tabanlı KS için 30 kořmanın en kötüsi

Uygulama 3.2 için yapılan 30 koşmalık YAKA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.52 ile, alınan en büyük hatalı sonuç Şekil 3.53 ile gösterilmiştir.

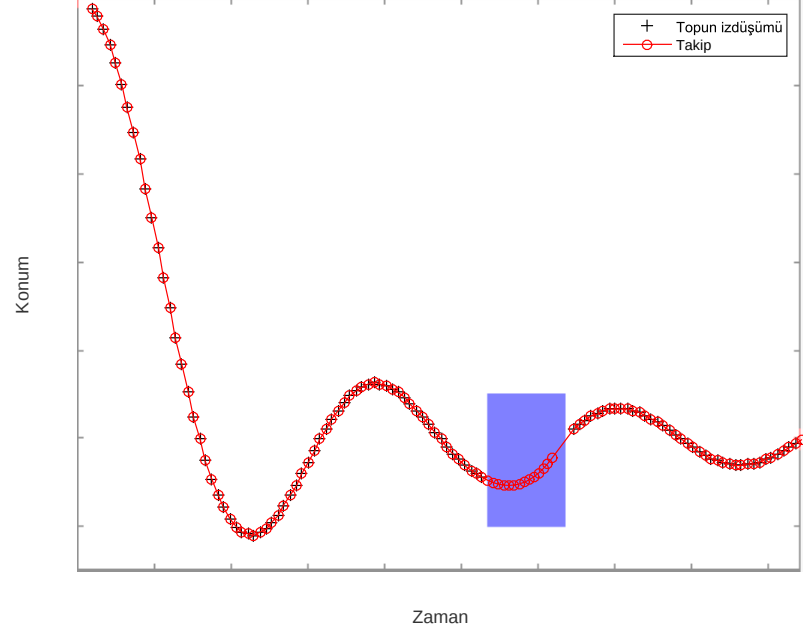


Şekil 3.52. Uygulama 3.2, YAKA tabanlı KS için 30 koşmanın en iyisi

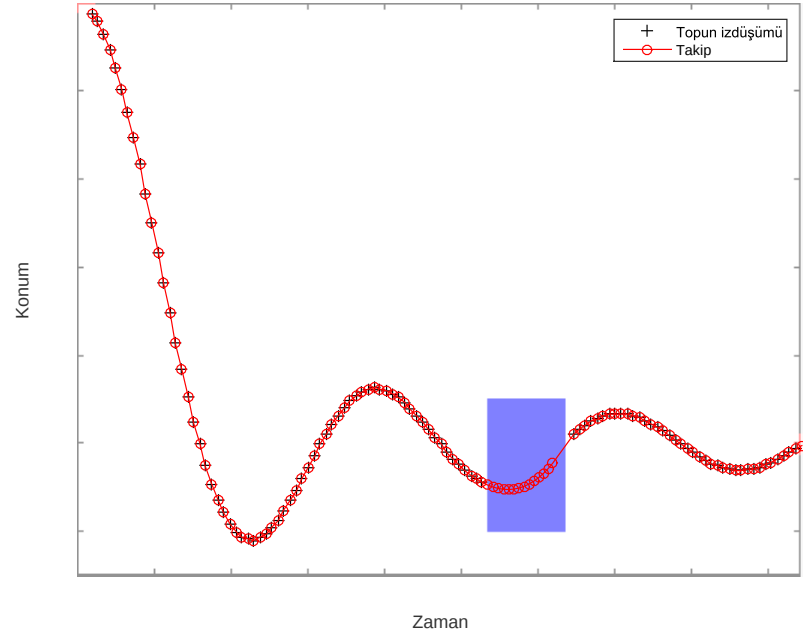


Şekil 3.53. Uygulama 3.2, YAKA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 3.2 için yapılan 30 kořmalık DGA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.54 ile, alınan en büyük hatalı sonuç Şekil 3.55 ile gösterilmiştir.

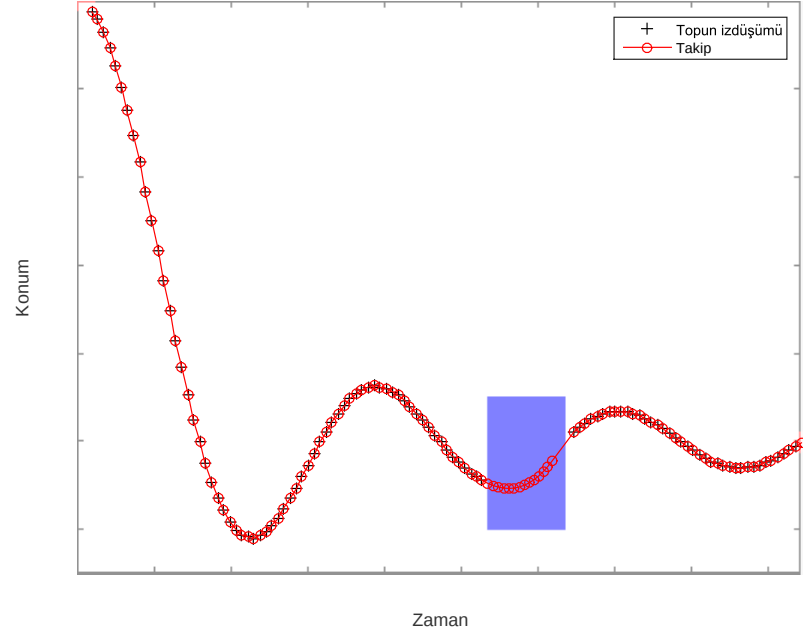


Şekil 3.54. Uygulama 3.2, DGA tabanlı KS için 30 kořmanın en iyisi

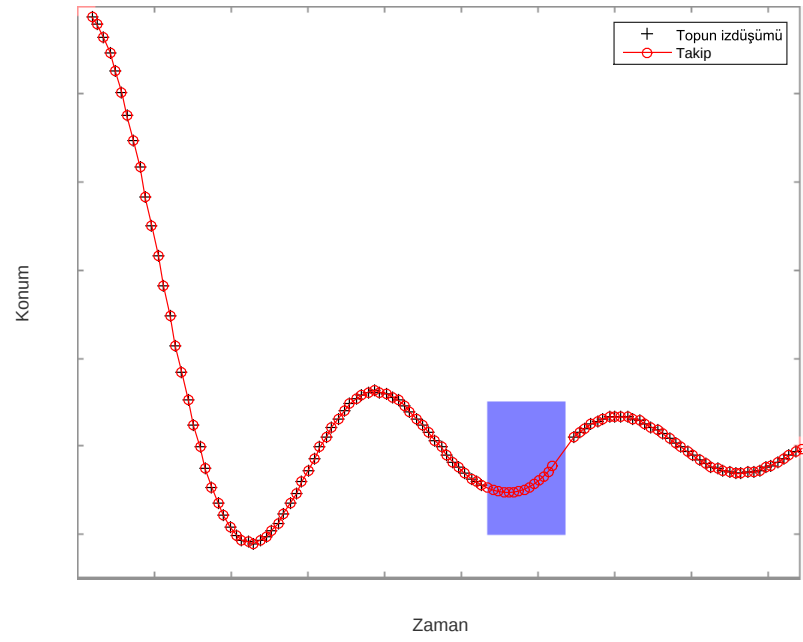


Şekil 3.55. Uygulama 3.2, DGA tabanlı KS için 30 kořmanın en kötüsü

Uygulama 3.2 için yapılan 30 kořmalık PSOA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.56 ile, alınan en büyük hatalı sonuç Şekil 3.57 ile gösterilmiştir.

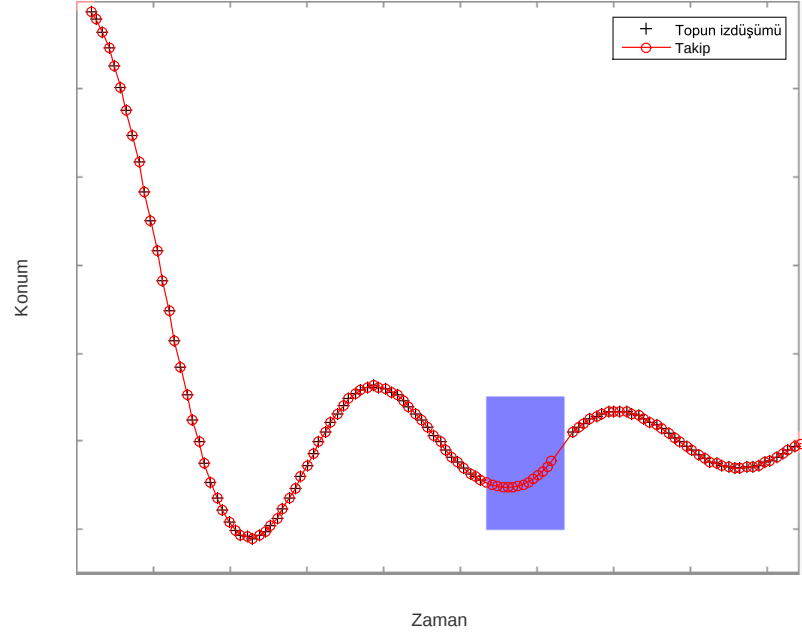


Şekil 3.56. Uygulama 3.2, PSOA tabanlı KS için 30 kořmanın en iyisi

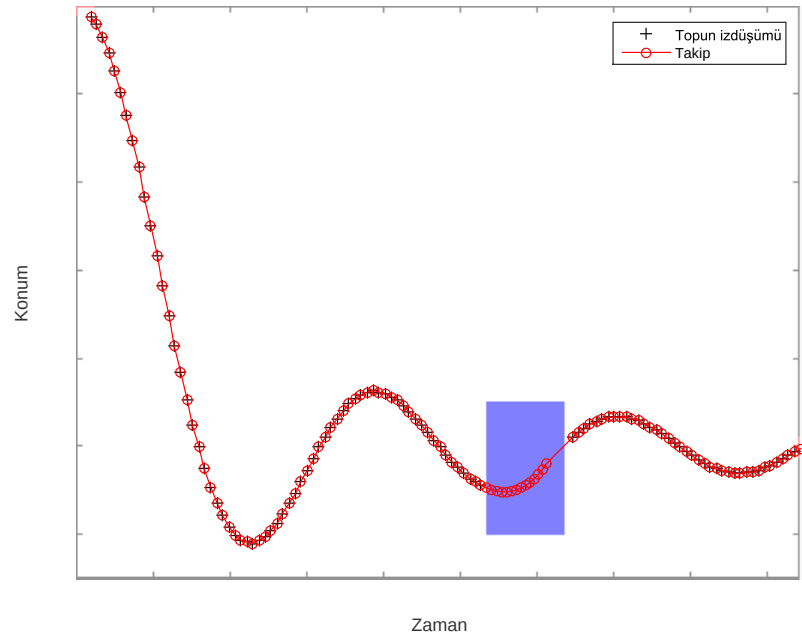


Şekil 3.57. Uygulama 3.2, PSOA tabanlı KS için 30 kořmanın en kötüsi

Uygulama 3.2 için yapılan 30 kořmalık ABA tabanlı KS alışmasında, alınan en düşük hatalı sonuç Şekil 3.58 ile, alınan en büyük hatalı sonuç Şekil 3.59 ile gösterilmiştir.



Şekil 3.58. Uygulama 3.2, ABA tabanlı KS için 30 kořmanın en iyisi



Şekil 3.59. Uygulama 3.2, ABA tabanlı KS için 30 kořmanın en kötüsi

3.3.4.3. Uygulama 3.3 Sonucu

Uygulama 3.3'te kullanıcı deneyimine bağlı olarak Tablo 3.32 ile gösterilen 3 farklı parametre seçimi gerçekleştirilmiştir. Bu parametre seçimine bağlı olarak oluşan sonuç grafikleri Şekil 3.61, 3.62 ve 3.63 ile verilmektedir. Bu uygulamada KS parametreleri için kullanılan güncel zeki optimizasyon algoritmalarının 30 koşmanın en iyi ve en kötü sonucuna göre aldığı parametreler Tablo 3.32 ile gösterilmiştir. Her algoritma için koşulan 30 iterasyon sonucu ortaya çıkan hata dağılımları Tablo 3.33 ile verilmiştir.

Tablo 3.32. Uygulama 3.3 için kullanıcı deneyimi ve optimizasyon algoritmaları tarafından seçilen KS parametreleri

Algoritma	P	Q	R1	R2	R3
Standart-KS (V1)	0.5	0.5	0.5	0.5	0.5
Standart-KS (V2)	1	.99	1	0.3	0.7
Standart-KS (V3)	0.1	1	0.1	3	0.85
GA-KS (En iyi)	0.6611	0.9999	0.0632	0.0040	0.9997
GA-KS (En kötü)	0.9898	0.9996	0.0976	0.0224	1
YAKA-KS (En iyi)	0.4520	0.7029	0.0436	0.0001	1
YAKA-KS (En kötü)	0.2768	0.7633	0.0418	0.0304	1
DGA-KS (En iyi)	0.5735	0.8551	0.0540	0.00058502	0.9996
DGA-KS (En kötü)	0.9988	0.9722	0.0929	0.0022	0.9602
PSOA-KS (En iyi)	0.6610	0.9993	0.0628	0.0015	1
PSOA-KS (En kötü)	0.9783	0.9527	0.0910	0.0021	0.9834
ABA-KS (En iyi)	0.9097	0.7768	0.0795	0.00001	0.2986
ABA-KS (En kötü)	0.9672	0.5243	0.0702	0.00001	0.4440

Tablo 3.33 incelendiğinde standart sapması en küçük olan DGA, kararlılığı en yüksek olan algoritmadır. $7.230761E-05$ değeri ile DGA en düşük MSE değerine sahiptir. Bir diğer performans kriteri olan işlem süresine göre PSOA'nın diğer 4 algoritmaya göre daha hızlı çalıştığı gözlemlenmiştir.

Uygulama 3.3 için yapılan istatistiksel analizde, Tablo 3.34 ile Kruskal Wallis test sonucunda anlamlılık değerinin (p), $1.21E-19$ değerine eşit olması bu algoritmalar arasında anlamlı farkın olduğunu göstermektedir. Tablo 3.35 incelendiğinde Mann-Whitney U testinin tüm algoritmaların ikili kombinasyonları için uygulandığı ve bu algoritmaların birbirlerine göre anlamlı farka sahip olup olmadıkları gözlemlenmiştir.

Tablo 3.33. Kullanılan algoritmaların uygulama 3.3 için hata dağılımı

Uygulama 3.3 İçin Algoritmaların Hata Dağılımı						
Algoritmalar	En İyi	En Kötü	Medyan	Ortalama	Standart Sapma	İşlem Süresi (sn)
GA	7.081927E-05	1.832921E-04	9.273430E-05	1.020650E-04	2.553917E-05	920
YAKA	7.859055E-05	5.496625E-04	9.462992E-05	1.199800E-04	8.804016E-05	922
DGA	6.934227E-05	8.993598E-05	7.000986E-05	7.230761E-05	6.057500E-06	910
PSOA	6.914152E-05	8.950590E-05	8.949613E-05	8.271763E-05	9.752088E-06	890
ABA	6.943283E-05	1.199189E-03	1.476600E-04	2.841548E-04	2.786848E-04	903

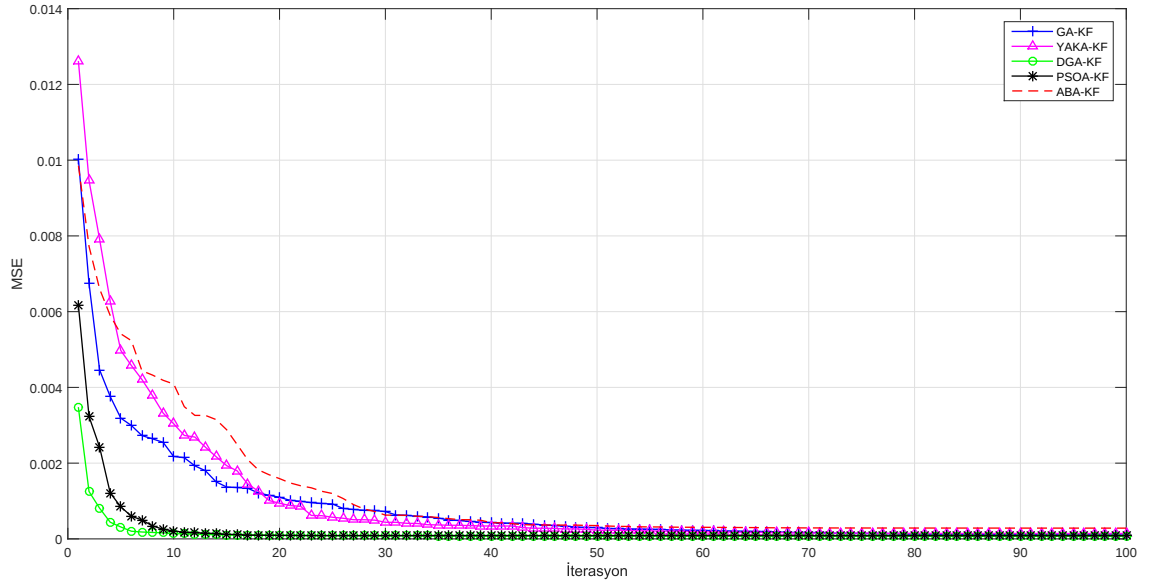
Tablo 3.34. Uygulama 3.3 için tüm algoritmalarda Kruskal Wallis test sonucu

Algoritma	N	df	Chi_Square	p
GA	30	4	94.877	1.21E-19
YAKA	30			
DGA	30			
PSOA	30			
ABA	30			

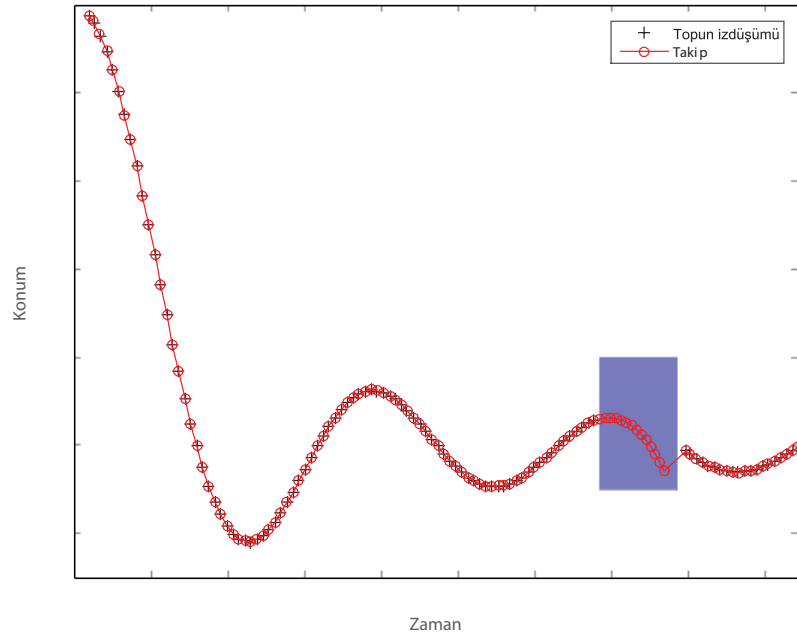
Tablo 3.35. Uygulama 3.3 için ikili algoritmalarda Mann-Whitney U test sonucu

Algoritmalar	Z	Ortalamaların Farkı	p	Anlamlı Fark
GA-YAKA	-2.04	-1.79150333E-05	0.041325	GA Lehine
GA-DGA	-6.298	2.97573497E-05	3.01E-10	DGA Lehine
GA-PSOA	-5.604	1.93473324E-05	2.10E-08	PSOA Lehine
GA-ABA	-3.889	-1.82089794E-04	0.000101	GA Lehine
YAKA-DGA	-6.476	4.76723830E-05	9.44E-11	DGA Lehine
YAKA-PSOA	-5.471	3.72623657E-05	4.48E-08	PSOA Lehine
YAKA-ABA	-2.203	-1.64174761E-04	0.027586	YAKA Lehine
DGA-PSOA	-1.346	-1.04100173E-05	0.178453	Yok
DGA-ABA	-6.21	-2.11847144E-04	5.29E-10	DGA Lehine
PSOA-ABA	-6.063	-2.01437127E-04	1.34E-09	PSOA Lehine

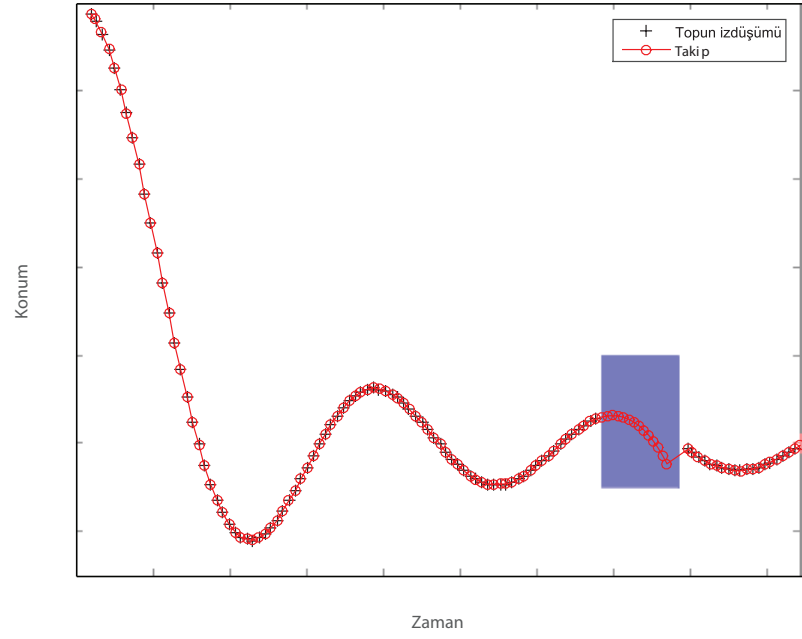
Uygulama 3.3 için yapılan kullanıcı deneyimi tabanlı KS çalışmasında, ortaya çıkan sonuçlar Şekil 3.61, 3.62 ve 3.63'te verilmiştir.



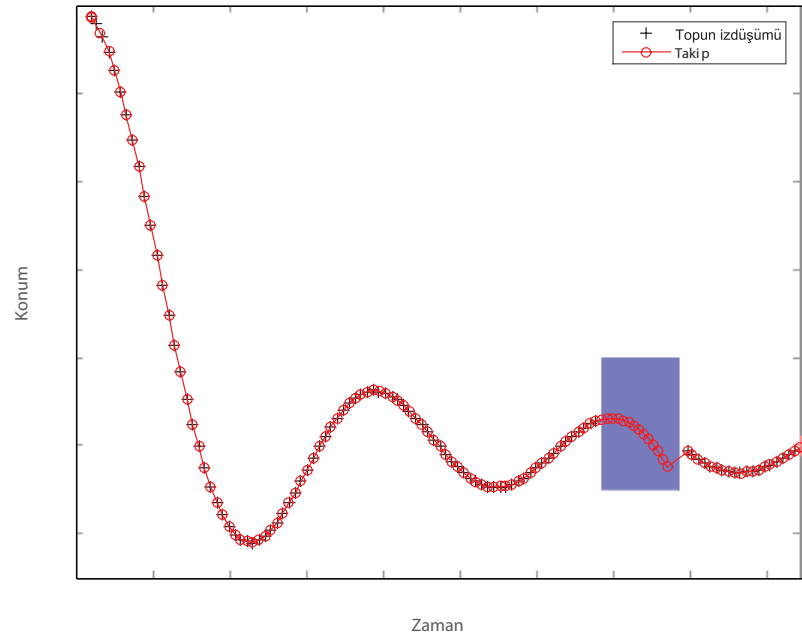
Şekil 3.60. Uygulama 3.3 için yöntemlerin iterasyon-MSE değişimi



Şekil 3.61. Uygulama 3.3, kullanıcı deneyimi tabanlı KS (D1)

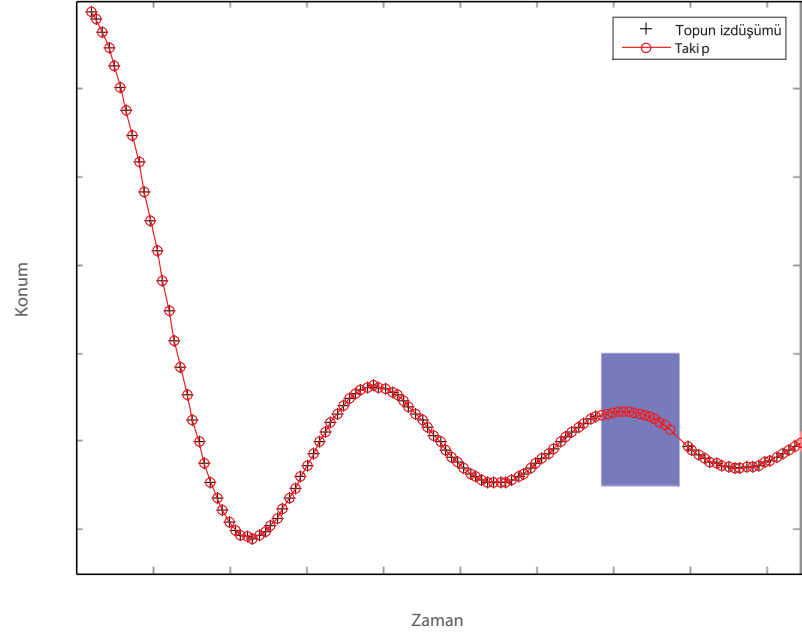


Şekil 3.62. Uygulama 3.3, kullanıcı deneyimi tabanlı KS (D2)

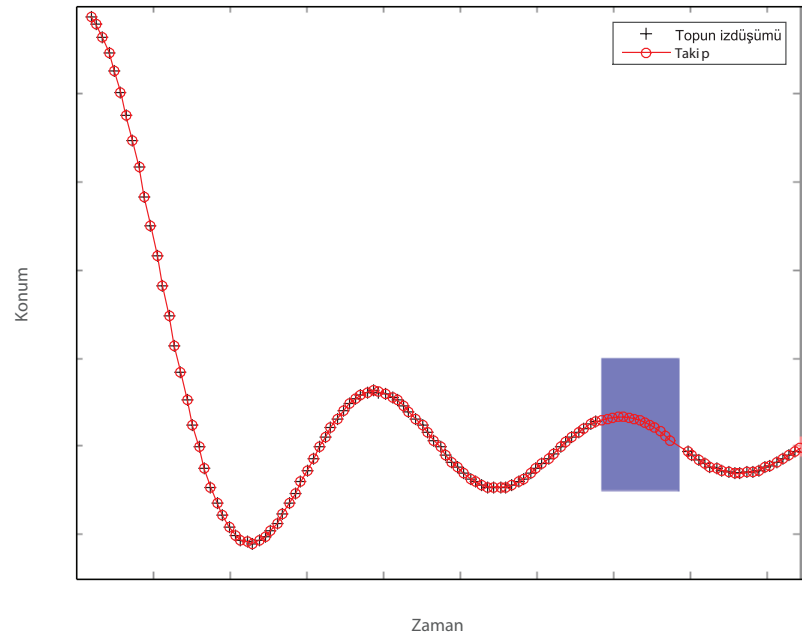


Şekil 3.63. Uygulama 3.3, kullanıcı deneyimi tabanlı KS (D3)

Uygulama 3.3 için yapılan 30 kořmalık GA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.64 ile, alınan en büyük hatalı sonuç Şekil 3.65 ile gösterilmiştir.

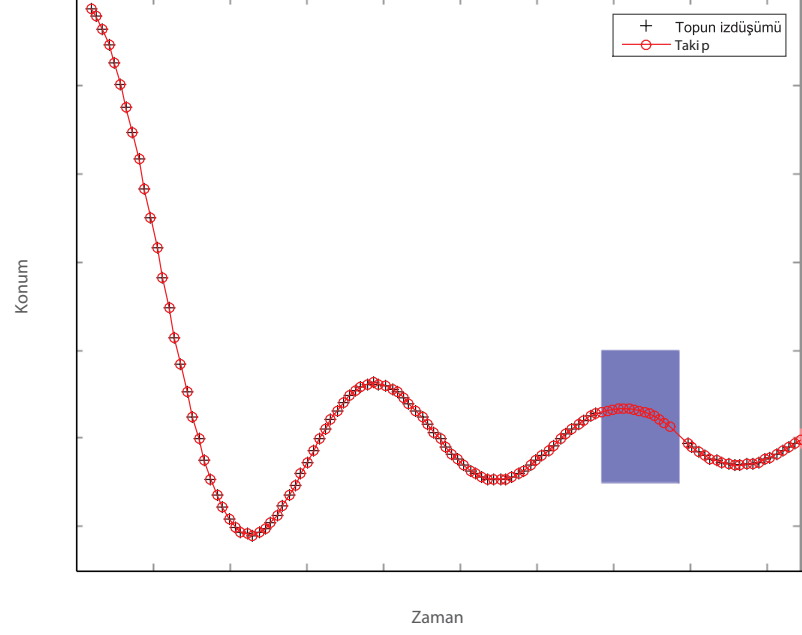


Şekil 3.64. Uygulama 3.3, GA tabanlı KS için 30 kořmanın en iyisi

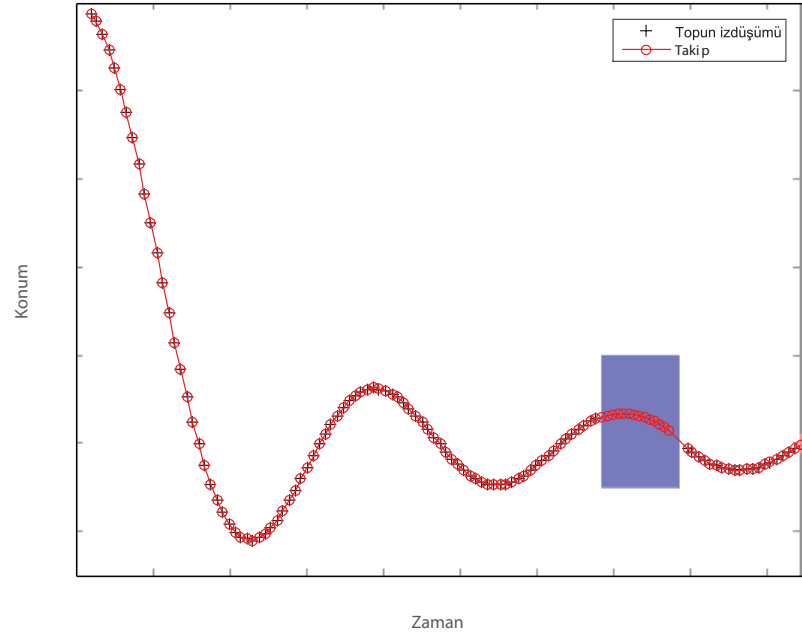


Şekil 3.65. Uygulama 3.3, GA tabanlı KS için 30 kořmanın en kötüsi

Uygulama 3.3 için yapılan 30 koşmalık YAKA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.66 ile, alınan en büyük hatalı sonuç Şekil 3.67 ile gösterilmiştir.

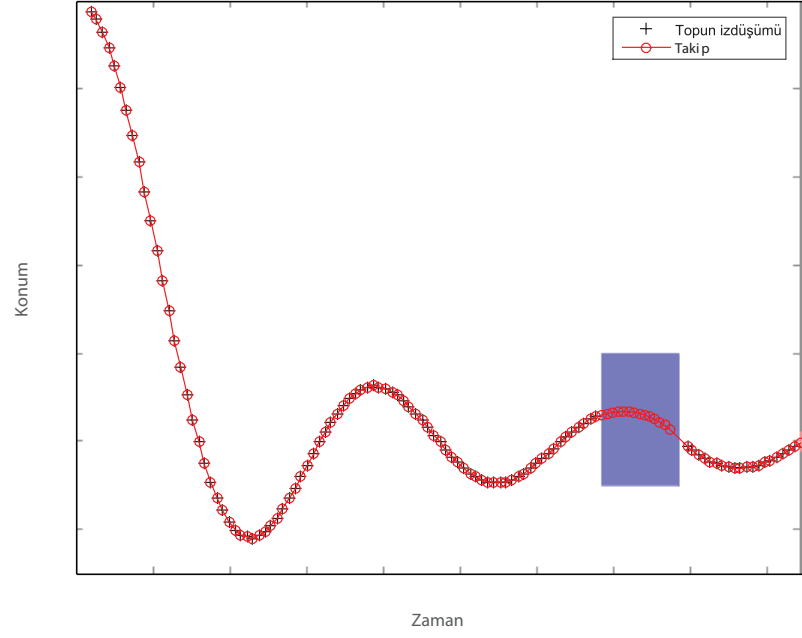


Şekil 3.66. Uygulama 3.3, YAKA tabanlı KS için 30 koşmanın en iyisi

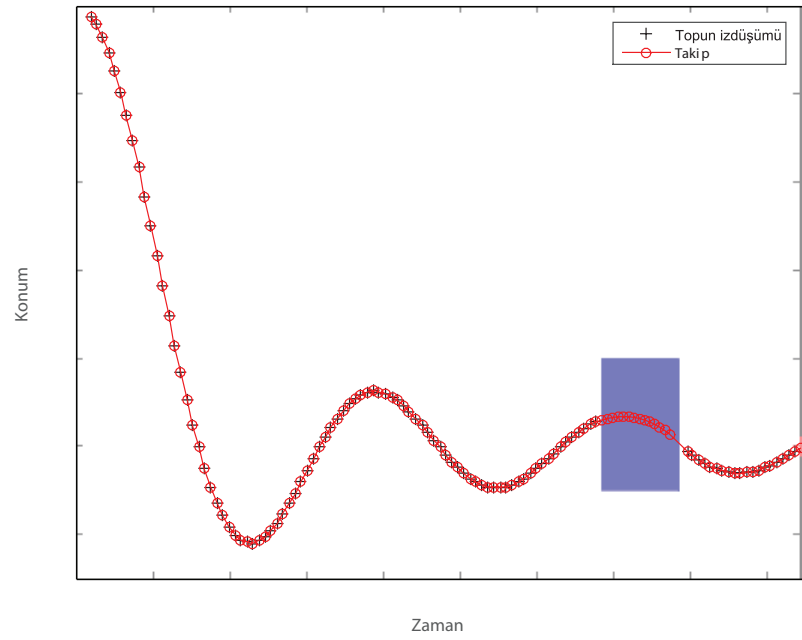


Şekil 3.67. Uygulama 3.3, YAKA tabanlı KS için 30 koşmanın en kötüsü

Uygulama 3.3 için yapılan 30 kořmalık DGA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.68 ile, alınan en büyük hatalı sonuç Şekil 3.69 ile gösterilmiştir.

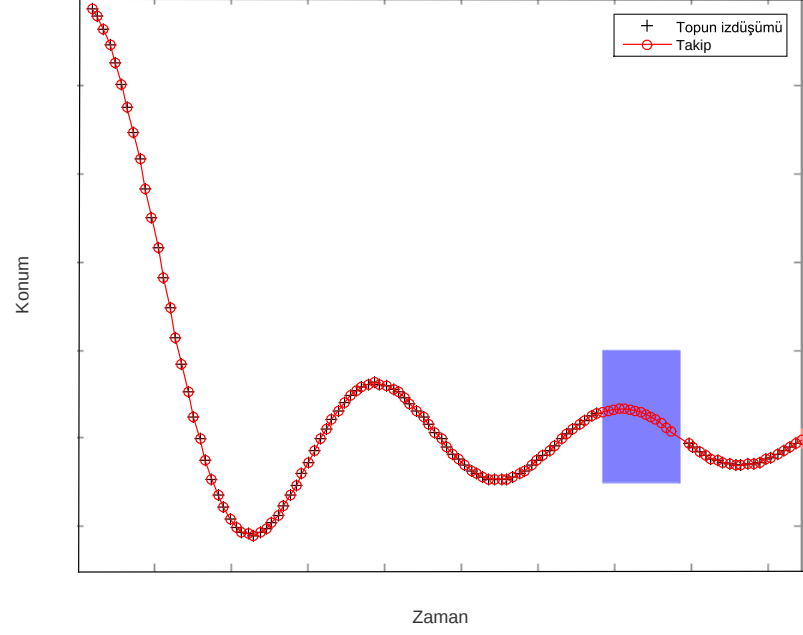


Şekil 3.68. Uygulama 3.3, DGA tabanlı KS için 30 kořmanın en iyisi

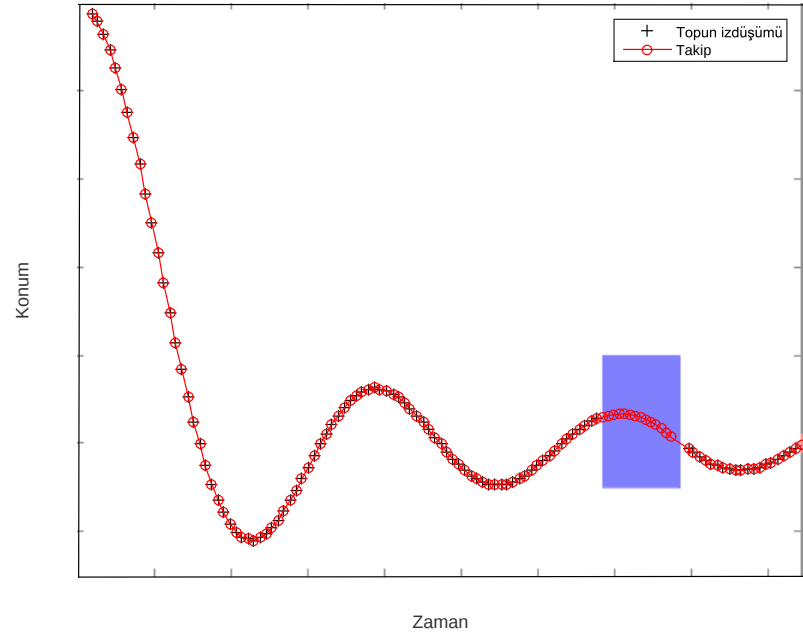


Şekil 3.69. Uygulama 3.3, DGA tabanlı KS için 30 kořmanın en kötüsü

Uygulama 3.3 için yapılan 30 kořmalık PSOA tabanlı KS çalışmasında, alınan en düşük hatalı sonuç Şekil 3.70 ile, alınan en büyük hatalı sonuç Şekil 3.71 ile gösterilmiştir.

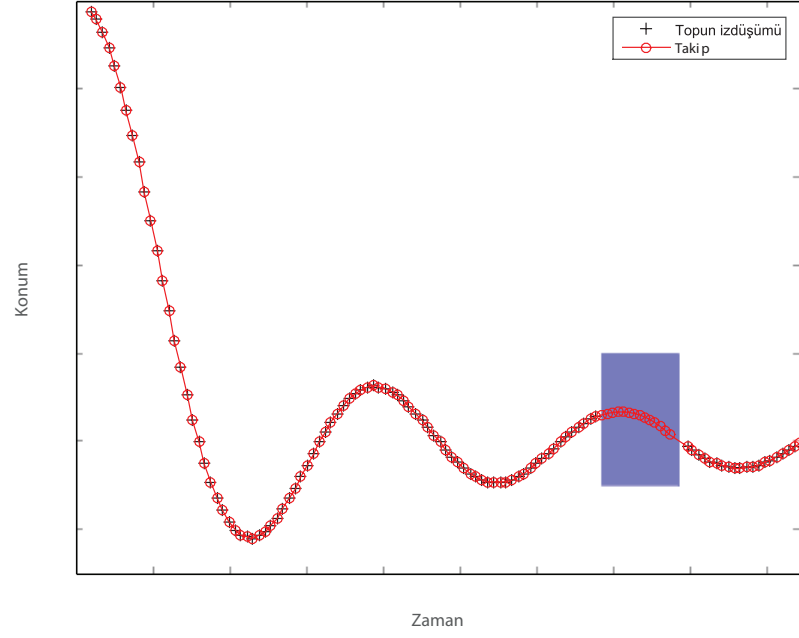


Şekil 3.70. Uygulama 3.3, PSOA tabanlı KS için 30 kořmanın en iyisi

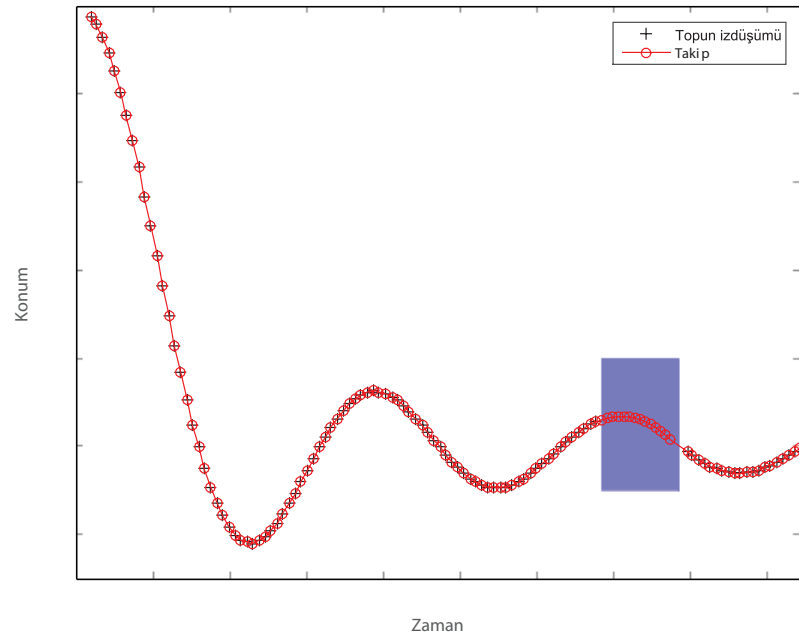


Şekil 3.71. Uygulama 3.3, PSOA tabanlı KS için 30 kořmanın en kötüsi

Uygulama 3.3 için yapılan 30 kořmalık ABA tabanlı KS alışmasında, alınan en düşük hatalı sonuç Şekil 3.72 ile, alınan en büyük hatalı sonuç Şekil 3.73 ile gösterilmiştir.



Şekil 3.72. Uygulama 3.3, ABA tabanlı KS için 30 kořmanın en iyisi



Şekil 3.73. Uygulama 3.3, ABA tabanlı KS için 30 kořmanın en kötüřü

4. BÖLÜM

TARTIŞMA, SONUÇ ve ÖNERİLER

4.1. Tartışma, Sonuç ve Öneriler

KS algoritması 3 farklı uygulamada gürültüden arındırma ve konum tespiti yapmak için kullanılmıştır. KS parametrelerinin doğru seçimi, optimal sonuçların elde edilmesi için büyük önem arz etmektedir. 3 ayrı uygulamada da kullanıcı deneyimine bırakılan KS parametre seçimi dolayısıyla oluşan uygulama sonuçlarında optimal değerlere ulaşmak çok güçtür. KS parametrelerinden ölçüm hata kovaryansı R değeri ilk uygulamada, yine ölçüm hata kovaryansı R_1 ve R_2 değerleri ikinci uygulamada ve son olarak başlangıç hata kovaryansı P , işlem hata kovaryansı Q , ölçüm hata kovaryansı R_1, R_2, R_3 değerleri son uygulamada eniyilenmek üzere güncel zeki optimizasyon algoritmalarıyla belirlenmiştir. Ortaya çıkan sonuca göre, en yüksek hata ile çalışan optimizasyon algoritması bile kullanıcı deneyimine bağlı seçilen parametrelere göre daha doğru sonuçlar üretmiştir. Optimizasyon algoritmaları kendi içinde, doğruluk ve işlem süresi kriterlerine göre karşılaştırılmış ve sonuç olarak her uygulama için farklı algoritmaların hem işlem süresi hem de doğruluk kriterine göre daha başarılı olduğu tespit edilmiştir. Bu tez çalışmasında ayrıca uygulama sonuçlarının istatistiksel analizi yapılmış ve algoritmalar arasındaki farklar anlamlılık değerleriyle karşılaştırılmıştır. Nesne takibi uygulamasında, sistemin ölçülemeyen durumlarında da, geçmiş verilerden faydalanarak dolaylı olarak kestirim yapılabilmektedir. Böylelikle hem tespit edilemeyen durumlarda kestirim yapılmış hem de sistemin gelecek verileri için tahmin mekanizması oluşturulmuştur. Bu uygulamada ayrıca aynı sahnede tünelin farklı konumlara yerleştirilerek tünelden önce ne kadar çok bilgi girişi olursa, uygulama sonucunda hata oranının çok daha fazla düştüğü tespit edilmiştir.

Gelecekte yapılabilecek çalışmalar için öneriler:

- Aynı uygulamalar bu tez çalışmasında denenmemiş olan farklı güncel zeki optimizasyon algoritmalarıyla test edilebilir.
- Bu tez çalışmasında kullanılan optimizasyon algoritmaları farklı uygulamalar için denenebilir.
- Farklı donanım özelliklerindeki bilgisayarla koşma yapıp karşılaştırma yapılabilir.
- Farklı uygunluk fonksiyonları uygulanarak karşılaştırma yapılabilir.

KAYNAKLAR

1. Kalman, R.E., 1960. A New Approach to Linear Filtering and Prediction Problems, **Journal of Basic Engineering**, **82**(1):35.
2. Yan, J., Yuan, D., Xing, X., and Jia, Q., 2008. Kalman filtering parameter optimization techniques based on genetic algorithm, *2008 IEEE International Conference on Automation and Logistics*, Institute of Electrical & Electronics Engineers (IEEE).
3. Ramakoti, N., Vinay, A., and Jatoth, R.K., 2009. Particle Swarm Optimization Aided Kalman Filter for Object Tracking, *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, Institute of Electrical & Electronics Engineers (IEEE).
4. Jatoth, R.K. and Kumar, T.K., 2009. Particle Swarm Optimization Based Tuning of Unscented Kalman Filter for Bearings Only Tracking, *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, Institute of Electrical & Electronics Engineers (IEEE).
5. Jatoth, R.K. and Reddy, G.A., 2010. A Hybrid GA-Adaptive Particle Swarm Optimization Based Tuning of Unscented Kalman Filter for Harmonic Estimation, *Swarm, Evolutionary, and Memetic Computing*, 380–388, Springer Science Business Media.
6. Jatoth, R.K., Rao, D.N., and Kumar, K.S., 2010. Particle Swarm Optimization aided unscented kalman filter for ballistic target tracking, *2010 International Conference On Communication Control And Computing Technologies*, Institute of Electrical & Electronics Engineers (IEEE).
7. Jin, Y., 2012. Application of Differential Evolution to the Parameter Optimization of the Unscented Kalman Filter, *Communications in Computer and Information Science*, 341–346, Springer Science Business Media.

8. Ting, T.O., Man, K.L., Lim, E.G., and Leach, M., 2014. Tuning of Kalman Filter Parameters via Genetic Algorithm for State-of-Charge Estimation in Battery Management System, **The Scientific World Journal**, **2014**:1–11.
9. Lin, G., Jing, Z., and Liu, Z., 2014. Tuning of Extended Kalman Filter using Improved Particle Swarm Optimization for Sensorless Control of Induction Motor, **Journal of Computational Information Systems**, **2014**:2455–2462.
10. Laamari, Y., Chafaa, K., and Athamena, B., 2014. Particle swarm optimization of an extended Kalman filter for speed and rotor flux estimation of an induction motor drive, **Electr Eng**, **97**(2):129–138.
11. ve Alper Basturk, T.O., 2016. Kalman Süzgeci Parametrelerinin Nesne Takibi Amacıyla Zeki Optimizasyon Algoritmalarıyla Belirlenmesi, *1st International Conference on Engineering Technology and Applied Sciences*, 1384–1388, Afyon Kocatepe University.
12. Ozcan, T., Badem, H., and Basturk, A., 2016. Artificial Bee Colony Algorithm Based Parameter Tuning of Kalman Filter for Object Tracking, *International Conference on Information Complexity and Statistical Modeling in High Dimensions with Applications*, nev.
13. Fridholm, B., Wik, T., and Nilsson, M., 2016. Kalman filter for adaptive learning of look-up tables with application to automotive battery resistance estimation, **Control Engineering Practice**, **48**:78–86.
14. Cayiroglu, I., 2012. Kalman Filtresi ve Programlama, *Fen ve Teknoloji Bilgi Paylaşımı, 2012-1*, Karabuk University.
15. Grewal, M.S. and Andrews, A.P., 2001. Kalman Filtering : Theory and Practice Using MATLAB, Wiley-Interscience.
16. Zhou, Z., Wu, D., and Zhu, Z., 2016. Object tracking based on Kalman particle filter with LSSVR, **Optik - International Journal for Light and Electron Optics**, **127**(2):613–619.

17. Li, Y., Wang, C., and Gong, J., 2016. A combination Kalman filter approach for State of Charge estimation of lithium-ion battery considering model uncertainty, **Energy**, **109**:933–946.
18. Boudet, J., L  v  que, E., Borgnat, P., Cahuzac, A., and Jacob, M., 2016. A Kalman filter adapted to the estimation of mean gradients in the large-eddy simulation of unsteady turbulent flows, **Computers & Fluids**, **127**:65–77.
19. Holland, J.H., 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, A Bradford Book.
20. Karaboga, D., 2011. *Yapay Zeka Optimizasyon Algoritmaları*, Nobel.
21. Storn, R. and Price, K., 1997. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, **Journal of Global Optimization**, **11**(4):341–359.
22. Das, S., Konar, A., and Chakraborty, U.K., 2005. Improving particle swarm optimization with differentially perturbed velocity, *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO 05*, Association for Computing Machinery (ACM).
23. Das, S., Konar, A., and Chakraborty, U.K., 2007. Annealed Differential Evolution, *2007 IEEE Congress on Evolutionary Computation*, Institute of Electrical & Electronics Engineers (IEEE).
24. Moore, P.W. and Venayagamoorthy, G.K., 2006. Evolving digital circuits using hybrid Particle Swarm Optimization and Differential Evolution, **Int. J. Neur. Syst.**, **16**(03):163–177.
25. Zhang, W.J. and Xie, X.F., 2003. DEPSO: hybrid particle swarm with differential evolution operator, *SMC03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, Institute of Electrical & Electronics Engineers (IEEE).

26. Kaelo, P. and Ali, M.M., 2007. Differential evolution algorithms using hybrid mutation, **Computational Optimization and Applications**, **37**(2):231–246.
27. Kennedy, J. and Eberhart, R., 1995. Particle swarm optimization, *Proceedings of ICNN95 - International Conference on Neural Networks*, Institute of Electrical & Electronics Engineers (IEEE).
28. Nieto, P.G., Garcia-Gonzalo, E., Arbat, G., Duran-Ros, M., de Cartagena, F.R., and Puig-Bargues, J., 2016. A new predictive model for the filtered volume and outlet parameters in micro-irrigation sand filters fed with effluents using the hybrid PSO–SVM-based approach, **Computers and Electronics in Agriculture**, **125**:74–80.
29. Moghaddam, M.A., Golmezergi, R., and Kolahan, F., 2016. Multi-variable measurements and optimization of GMAW parameters for API-X42 steel alloy using a hybrid BPNN–PSO approach, **Measurement**, **92**:279–287.
30. Garg, H., 2016. A hybrid PSO-GA algorithm for constrained optimization problems, **Applied Mathematics and Computation**, **274**:292–305.
31. Yang, X.S., 2010. Nature-Inspired Metaheuristic Algorithms: Second Edition, Luniver Press.
32. Othman, M., El-Khattam, W., Hegazy, Y., and Abdelaziz, A.Y., 2016. Optimal placement and sizing of voltage controlled distributed generators in unbalanced distribution networks using supervised firefly algorithm, **International Journal of Electrical Power & Energy Systems**, **82**:105–113.
33. Gope, S., Goswami, A.K., Tiwari, P.K., and Deb, S., 2016. Rescheduling of real power for congestion management with integration of pumped storage hydro unit using firefly algorithm, **International Journal of Electrical Power & Energy Systems**, **83**:434–442.
34. Oguzlar, A., 2007. Istatistiksel Veri Analizi 1, Ezgi Kitabevi Yayinlari.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı : Tayyip ÖZCAN
Uyruğu : Türkiye (T.C.)
Doğum Tarihi ve Yeri : 01.06.1986 Yahyalı/KAYSERİ
Telefon : 0 352 207 66 66
Belgegeçer : 0 352 437 57 84
E-posta : tozcan@erciyes.edu.tr
Adres : Erciyes Üniversitesi Mühendislik Fakültesi
 Bilgisayar Mühendisliği Bölümü
 38039, Melikgazi KAYSERİ TÜRKİYE

EĞİTİM

Derece	Kurum	Mez.Yılı
Lisans	Kültür Üni. Müh. ve Mim. Fakültesi Bilgisayar Müh., İSTANBUL	2010

İŞ DENEYİMİ

Yıl	Kurum	Görev
2011-2011	Şahin Yazılım, KAYSERİ	Yazılım Destek
2011-2013	Merkez Çelik A.Ş., KAYSERİ	Yazılımcı
2013-Halen	ERÜ Mühendislik Fakültesi Bilgisayar Müh. Bölümü, KAYSERİ	Araştırma Görevlisi