

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ÇOKLU MODEL PARÇACIK FİLTRELERİNDE
AĞIRLIKLANDIRILMIŞ İSTATİSTİKSEL MODEL SEÇİMİ**

MURAT BARKAN UÇAR

YÜKSEK LİSANS TEZİ

2015

**ÇOKLU MODEL PARÇACIK FİLTRELERİNDE
AĞIRLIKLANDIRILMIŞ İSTATİSTİKSEL MODEL SEÇİMİ**

**WEIGHTED STATISTICAL MODEL SELECTION FOR
MULTI MODEL PARTICLE FILTERS**

MURAT BARKAN UÇAR

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
ELEKTRİK-ELEKTRONİK Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2015

“Çoklu Model Parçacık Filtrelerinde Ağırlıklandırılmış İstatistiksel Model Seçimi” başlıklı bu çalışma, jürimiz tarafından, 12/08/2015 tarihinde, **ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan :
Doç. Dr. Hasan Şakir Bilge

Üye (Danışman) :
Yrd. Doç. Dr. Derya Yılmaz

Üye :
Doç. Dr. Mustafa Doğan

ONAY

.../.../.....

Prof. Dr. Emin AKATA
Fen Bilimleri Enstitüsü Müdürü

TEŐEKKÜR

Tez danışmanım Sayın Yrd. Doç. Dr. Derya YILMAZ'a çalışmanın sonuca ulaştırılmasında ve karşılaşılan güçlüklerin aşılmasında her zaman yardımcı ve yol gösterici olduđu için...

Aileme, tez boyunca verdikleri manevi destek için...

ÖZ

ÇOKLU MODEL PARÇACIK FİLTRELERİNDE AĞIRLIKLANDIRILMIŞ İSTATİSTİKSEL MODEL SEÇİMİ

Murat Barkan UÇAR

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Radar hedeflerinin takibindeki önemli unsurlardan birisi kullanılan algoritmanın hızıdır. Çünkü hedefler hareket halinde olduğu için gerçek zamanlı takip gerektirir. Hedef takibinde sıklıkla tercih edilen çoklu model parçacık filtresi (ÇMPF) algoritmasında, kullanılan parçacık sayısı ve hareket modellerinin tahmini için gerçekleştirilen işlem sayısı filtrenin hızını belirleyen en önemli parametrelerdir. Parçacık sayısının ve/veya model hesaplamalarının azaltılması, algoritmanın hızlanmasını sağlayarak hedeflerin gerçek zamanlı takibini kolaylaştıracaktır. Bu çalışmada, model hesaplamalarının azaltılması amacıyla ağırlıklandırılmış istatistiksel model seçimi (AİMS) adı verilen yeni bir yaklaşım önerilmekte ve ÇMPF üzerinde gerçekleştirilen uygulamalarına ilişkin sonuçlar sunulmaktadır. Önerilen algoritmanın başarısını değerlendirmek için, farklı senaryolar üzerinde gerçekleştirilen benzetimlerde, üç farklı ÇMPF kullanılmaktadır. Bu filtrelerden biri, bu tez çalışması için tarafımızdan önerilen yeni bir ÇMPF olup, kullanılan filtrelerden ikisine önerdiğimiz AİMS algoritması entegre edilmektedir. Elde edilen sonuçlar işlem süresi ve tahmin hatası kriterleri esas alınarak karşılaştırılmaktadır. Sonuçlar incelendiğinde, önerilen yaklaşımın uygulandığı ÇMPF'lerde; işlem sürelerinin azaldığı ve algoritma hızının arttığı, tahmin hatasında ise önemli bir artış olmadığı görülmektedir. Sonuç olarak, önerilen bu yeni yaklaşım, radar hedeflerinin gerçek zamanlı takibinde etkin bir biçimde kullanılabilir.

ANAHTAR SÖZCÜKLER: çoklu model parçacık filtresi, hedef takip modelleri, optimizasyon, ağırlıklandırılmış istatistiksel model seçimi

Danışman: Yrd.Doç.Dr. Derya YILMAZ, Başkent Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü.

ABSTRACT

WEIGHTED STATISTICAL MODEL SELECTION FOR MULTI MODEL PARTICLE FILTERS

Murat Barkan UÇAR

Başkent University, Institute of Science Engineering

Department of Electrical and Electronics Engineering

The algorithm speed is the most important fact for tracking of radar targets. Because it requires a real time follow-up for targets motion. In the most preferred algorithm of multi model particle filter (MMPF) for target tracking, the number of calculations for the number of particles and the maneuvering model selection is the most important parameter for determining the process speed of filter. The particle number and/or model calculations should be reduced as much as possible so the reduction of these described two facts expedites the algorithm and eases a real time follow-up. In this study, a new approach which called weighted statistical model selection (WSMS) algorithm is proposed for reduction of model calculations and the results are presented about the applications preformed on MMPF. For evaluate the success of proposed algorithm, in simulations preformed on different scenarios, three different MMPF are used. One of them is a new MMPF which is proposed by us for this thesis. The WSMS is integrated into two of these filters in simulations and the obtained results are compared based on processing time and prediction error criteria. When the results are analyzed, MMPF with the proposed model selection approach; process time decreases so algorithm speed increases, there is no significant increase for prediction error. As a result, WSMS algorithm can be used effectively for maneuvering radar targets in real time follow-up.

KEYWORDS: multi model particle filter, target tracking models, optimization, weighted statistical model selection

Supervisor: Assoc.Prof.Dr. Derya YILMAZ, Başkent University, Department of Electrical and Electronics Engineering

İÇİNDEKİLER LİSTESİ

	<u>Sayfa</u>
ÖZ	i
ŞEKİLLER LİSTESİ	v
ÇİZELGELER LİSTESİ	vii
SİMGELER VE KISALTMALAR LİSTESİ	viii
1. GİRİŞ	1
2. TAKİP ALGORİTMALARI	4
2.1. Bayesian Filtresi	4
2.2. Monte Carlo Yöntemi	7
2.3. Önem Örnekleme	9
2.4. Kalman Filtresi	10
2.5. Parçacık Filtresi	12
2.5.1. Parçacıkların oluşturulması	15
2.5.2. Ağırlıklandırma	16
2.5.3. Yeniden örnekleme	17
2.5.3.1. Sistemik yeniden örnekleme (SYÖ)	18
2.5.3.2. Sıralı önem yeniden örnekleme (SÖYÖ)	21
2.6. Parçacık Filtresi ile Hedef Takibi	23
3. TAKİP MODELLERİ	26
3.1. Hedef Takibinde Matematiksel Modeller	26
3.2. Sabit Hız Modeli	27
3.3. Sabit İvme Modeli	29
3.4. Singer Hızlanma Modeli	30
3.5. İki Boyutlu Hareket Modelleri	31
3.5.1. Dönüş oranı bilinen koordineli dönüş modeli	31
3.5.2. Dönüş oranı bilinmeyen koordineli dönüş modeli	33
4. ÇOKLU MODEL FİLTRELERİ	34
4.1. Çoklu Model Parçacık Filtresi (ÇMPF)	34
4.1.1. Birinci algoritma	34
4.1.2. İkinci algoritma	36
4.1.3. Üçüncü algoritma	37
4.2. Etkileşimli Çoklu Model (EÇM)	39
4.2.1. Etkileşim adımı	40
4.2.2. Model eşleme filtresi	41

4.2.3. Model olasılığının güncelleştirilmesi	41
4.2.4. Kestirim ve kovaryans kombinasyonu	41
5. AĞIRLIKLANDIRILMIŞ İSTATİSTİKSEL MODEL SEÇİMİ (AIMS)	43
6. UYGULAMA.....	49
6.1. Kullanılan Materyal	49
6.2. Senaryo Parametreleri.....	50
6.3. Karşılaştırma Kriterleri	51
6.4. Benzetimlerde Kullanılan Senaryolar.....	51
6.4.1. Senaryo 1	51
6.4.2. Senaryo 2.....	53
6.4.3. Senaryo 3.....	54
6.5. Benzetim Sonuçları	55
6.5.1. Yarı dinamik yaklaşıma ait sonuçlar	56
6.5.2. Dinamik yaklaşıma ait sonuçlar	75
6.5.3. EÇM filtresi.....	96
7. SONUÇ VE ÖNERİLER.....	97
KAYNAKLAR LİSTESİ	100

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1	Bayesian Tahmin ve Güncelleme Adımları [10]	6
Şekil 2.2	Kalman Filtresi Döngüsü [12]	12
Şekil 2.3	Parçacık Filtresi Akış Diyagramı [10]	15
Şekil 2.4	Parçacık Ağırlık Grafiği [4]	19
Şekil 2.5	Sistemik Yeniden Örnekleme [4]	20
Şekil 2.6	Sistemik Yeniden Örnekleme Sonrası Parçacık Ağırlık Grafiği [4]	20
Şekil 2.7	Sıralı Önem Yeniden Örnekleme [22; 23]	22
Şekil 2.8	Hedef Yörünge Bilgisi [15]	23
Şekil 2.9	Parçacık Filtresi ile Hedef Takibi	25
Şekil 4.1	Algoritma 1 Akış Diyagramı	35
Şekil 4.2	Algoritma 2 Akış Diyagramı	37
Şekil 4.3	Algoritma 3 Akış Diyagramı	39
Şekil 4.4	EÇM Blok Diyagramı [29]	42
Şekil 5.1	AİMS Algoritması Akış Şeması	45
Şekil 6.1	Senaryo 1 Hedef Rota Bilgisi	52
Şekil 6.2	Senaryo 2 Hedef Rota Bilgisi	53
Şekil 6.3	Senaryo 3 Hedef Rota Bilgisi	55
Şekil 6.4	Senaryo 1 Tüm Parçacıklar İçin Hata Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)	65
Şekil 6.5	Senaryo 1 Tüm Parçacıklar İçin Zaman Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)	65
Şekil 6.6	Senaryo 1 Tüm Gürültüler İçin Hata Grafiği (Yarı Dinamik ve $N=5000$)	66
Şekil 6.7	Senaryo 1 Tüm Gürültüler İçin Zaman Grafiği (Yarı Dinamik ve $N=5000$)	66
Şekil 6.8	Senaryo 2 Hedef Takibi (Yarı Dinamik, $\sigma^2=0,1$ ve $N=5000$)	68
Şekil 6.9	Senaryo 2 RMSE Grafiği (Yarı Dinamik, $\sigma^2=0,1$ ve $N=5000$)	69
Şekil 6.10	Senaryo 2 Tüm Parçacıklar İçin Hata Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)	69
Şekil 6.11	Senaryo 2 Tüm Parçacıklar İçin Zaman Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)	70
Şekil 6.12	Senaryo 2 Tüm Gürültüler İçin Hata Grafiği (Yarı Dinamik ve $N=5000$)	71
Şekil 6.13	Senaryo 2 Tüm Gürültüler İçin Zaman Grafiği (Yarı Dinamik ve $N=5000$)	71
Şekil 6.14	Senaryo 3 Tüm Parçacıklar İçin Hata Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)	73
Şekil 6.15	Senaryo 3 Tüm Parçacıklar İçin Zaman Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)	73
Şekil 6.16	Senaryo 3 Tüm Gürültüler İçin Hata Grafiği (Yarı Dinamik ve $N=5000$)	74
Şekil 6.17	Senaryo 3 Tüm Gürültüler İçin Zaman Grafiği (Yarı Dinamik ve $N=5000$)	74
Şekil 6.18	Senaryo 1 Hedef Takibi (Dinamik, $\sigma^2=0,1$ ve $N=5000$)	84
Şekil 6.19	Senaryo 2 RMSE Grafiği (Dinamik, $\sigma^2=0,1$ ve $N=5000$)	85

Şekil 6.20	Senaryo 1 Tüm Parçacıklar İçin Hata Grafiği (Dinamik ve $\sigma^2=0,1$)	85
Şekil 6.21	Senaryo 1 Tüm Parçacıklar İçin Zaman Grafiği (Dinamik ve $\sigma^2=0,1$)..	86
Şekil 6.22	Senaryo 1 Tüm Gürültüler İçin Hata Grafiği (Dinamik ve $N=5000$)	87
Şekil 6.23	Senaryo 1 Tüm Gürültüler İçin Zaman Grafiği (Dinamik ve $N=5000$)..	87
Şekil 6.24	Senaryo 2 Tüm Parçacıklar İçin Hata Grafiği (Dinamik ve $\sigma^2=0,1$)	89
Şekil 6.25	Senaryo 2 Tüm Parçacıklar İçin Zaman Grafiği (Dinamik ve $\sigma^2=0,1$)..	89
Şekil 6.26	Senaryo 2 Tüm Gürültüler İçin Hata Grafiği (Dinamik ve $N=5000$)	90
Şekil 6.27	Senaryo 2 Tüm Gürültüler İçin Zaman Grafiği (Dinamik ve $N=5000$)..	90
Şekil 6.28	Senaryo 3 Hedef Takibi (Dinamik, $\sigma^2=0,15$ ve $N=5000$)	92
Şekil 6.29	Senaryo 3 RMSE Grafiği (Dinamik, $\sigma^2=0,15$ ve $N=5000$)	93
Şekil 6.30	Senaryo 3 Tüm Parçacıklar İçin Hata Grafiği (Dinamik ve $\sigma^2=0,1$)	93
Şekil 6.31	Senaryo 3 Tüm Parçacıklar İçin Zaman Grafiği (Dinamik ve $\sigma^2=0,1$)..	94
Şekil 6.32	Senaryo 3 Tüm Gürültüler İçin Hata Grafiği (Dinamik ve $N=5000$)	95
Şekil 6.33	Senaryo 3 Tüm Gürültüler İçin Zaman Grafiği (Dinamik ve $N=5000$)..	95

ÇİZELGELER LİSTESİ

	<u>Sayfa</u>
Çizelge 2.1 Kalman Filtresi ve Parçacık Filtresi Karşılaştırması [13; 14]	13
Çizelge 5.1 Model Ağırlık Çizelgesi	47
Çizelge 5.2 Model Ağırlık Çizelgesi	48
Çizelge 5.3 Toplam Olasılık Değeri	48
Çizelge 6.1 Senaryo 1 – Yarı Dinamik Yaklaşım Sonuçları	57
Çizelge 6.2 Senaryo 2 – Yarı Dinamik Yaklaşım Sonuçları	58
Çizelge 6.3 Senaryo 3 – Yarı Dinamik Yaklaşım Sonuçları	59
Çizelge 6.4 Senaryo 1 – Yarı Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi	61
Çizelge 6.5 Senaryo 2 – Yarı Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi	62
Çizelge 6.6 Senaryo 3 – Yarı Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi	63
Çizelge 6.7 Senaryo 1 –Dinamik Yaklaşım Sonuçları	76
Çizelge 6.8 Senaryo 2 – Dinamik Yaklaşım Sonuçları	77
Çizelge 6.9 Senaryo 3 – Dinamik Yaklaşım Sonuçları	78
Çizelge 6.10 Senaryo 1 –Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi	80
Çizelge 6.11 Senaryo 2 –Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi	81
Çizelge 6.12 Senaryo 3 –Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi	82
Çizelge 6.13 EÇM Filtresi Sonuçları	96

SİMGELER VE KISALTMALAR LİSTESİ

a	ivme
A	durum matrisi
B	gürültü matrisi
c	normalizasyon faktörü
$C(m)$	kümülatif ağırlık toplamı
d	konum bilgisi
F_k	durum geçiş fonksiyonu
f_z	gözlem fonksiyonu
G_k	işlem gürültü matrisi
G_k^u	kontrol giriş matrisi
H	ölçme matrisi
k	ayrık zaman sabiti
K	toplam adım sayısı
K_k	Kalman kazancı
L	benzetim sayısı
M	yeniden örnekleme sonrası oluşturulan parçacık sayısı
M_P	model türü
N	sisteme giriş yapan parçacık sayısı
N_{eff}	etkin örnek sayısı
P	pencere eleman sayısı
P_k	hata kovaryansı
Q	işlem gürültüsü kovaryansı
$r^{(m)}$	yeniden kopyalama faktörü
R	ölçüm gürültüsü kovaryansı
s	hız
T	periyot
u_k	durum gürültü vektörü
$U(m)$	tekdüze değer
v_k	gözlem gürültü vektörü
w	önem ağırlığı

w_n	beyaz gürültü
W_p	model ağırlığı
x	durum bilgisi (tahmin vektörü)
X	Çoklu model parçacık filtresi model sayısı
z	gözlem bilgisi (gözlem vektörü)
δ	Dirac fonksiyonu
Δt	zamandaki değişim
σ	varyans
σ^2	hızlanma varyansı
ω	dönüş açısal oranı
τ	korelasyon zaman sabiti
θ	hedef açı bilgisi
$\lambda(k)$	model benzerliği
$\mu(k)$	model olasılığı
AİMS	Ağırlıklandırılmış İstatistiksel Model Seçimi
ÇMPF	Çoklu Model Parçacık Filtresi
EÇM	Etkileşimli Çoklu Model
RADAR	RAdio Detecting And Ranging
RMSE	Ortalama Karesel Hata Miktarı
SÖYÖ	Sıralı Önem Yeniden Örneklemesi
SYÖ	Sistemik Yeniden Örnekleme

1. GİRİŞ

Radarlar (RAdioDetectingAndRanging), hedef tespiti ve hedef konumlamasını elektromanyetik dalgalar aracılığıyla yapan sistemlerdir. Hedefin tespit edilmesi gönderilen elektromanyetik enerji darbelerinin hedeften yansması ve bu yansıyan enerjinin tespit edilerek işlenmesi mantığına dayanır.

Her radarın çalışması üç temel fizik kuralına dayanır [1]. Bunlar; elektromanyetik dalgaların yansması, elektromanyetik dalgaların sabit yayılma hızı ve elektromanyetik dalgaların doğrusal yayılmasıdır. Yayın yapılan elektromanyetik dalgalar elektriksel iletken bir cisme çarpmaları halinde yansır. Yansıma sinyali yankı olarak algılanır. Yansıyan bu yankı algılama yapılan yönde bir hedefin var olduğu konusunda bilgi verir. Elektromanyetik dalgalar ışık hızına yakın bir hızla yayılırlar. Bu hız 3×10^8 m/s olarak kabul edilir. Hedeften yansıyan elektromanyetik dalgaların geri dönüş süresi ve elektromanyetik dalga hızından hedefin menzili hesaplanır. Elektromanyetik dalgalar radarın frekans bandı boyunca doğrusal olarak yayılırlar. Bu dalgalar anten ve faz kaydırıcılar tarafından dikey ve yatay ekseninde yoğunlaştırılabilirler. Bu üç özellik sayesinde hedefin irtifa, menzil ve açısı hesaplanabilir.

Hedeflere ait koordinatlar çizitler ve izler vasıtası hesaplanmaktadır. Çizitler olası hedefleri belirtirken, izler gerçek hedefleri belirtmektedir. Sistemde bulunan “yerel izleyici” hedefin önceden hesaplanan koordinatları ve radarın gelen son koordinatlarının ağırlıklı ortalamasını alarak izleri günceller [1]. Algılanan her bir çizitin mevcut bir ize uygun olup olmadığı kontrol edilir. Eğer uygunsa çizit ilgili izi güncellemek için kullanılır, uygun değil ise ilişkilendirilemeyen çizitler yeni çizitler oluşturmak için kullanılırlar.

İzlenen hedef öngörülemez bir manevra yaparsa veya radarın çok yakın bölgesinde bulunan, aktif bir gürültü kaynağı etkisi ile vs., bir gürültünün içerisine girerse hedefin izlenmesi güçleşecektir. “Yerel izleyici” bu koşullar altında elinde

bulunan algoritma veya algoritmaları verimli bir biçimde kullanmalı ve izleri gerçek zamanlı bir şekilde tespit edebilmelidir.

Radar, çizitin hesaplanan koordinatlarını önceden hesaplanan “beklenti penceresinde” bulunan mevcut izlerden hangisiyle eşleşebildiğini araştırır. Bu pencerenin boyutu hedefin davranışına (manevralarına), hızına ve ivmelenmesine bağlı olarak değişmektedir. Bu nedenle hedefin davranışları, hızı ve ivmelenmesinin modellenmesi ve kullanılacak olan algoritmalara bu modellerin entegre edilmesi gerekmektedir.

Çizit ve izin eşleştirilmesi için hedef takip algoritmaları kullanılır. Bunlardan biri parçacık filtresidir. Parçacık filtresi Monte Carlo tabanlı bir filtredir. Bu nedenle parçacık filtrelerinin gelişimini Monte Carlo filtrelerinden itibaren incelemek yerinde olur. Monte Carlo filtrelerinin geçmişi 1950’li yıllara dayanmaktadır [2]. Fakat bu yıllarda Monte Carlo filtresi hesaplama gücünü etkin kullanılamaması ve hesaplanan verilerde meydana gelen bozulmalar nedeniyle göz ardı edilmiştir. Özçıkırım (bootstrap) filtresinin ve yeniden örnekleme yöntemlerinin ortaya çıkmasından sonra bu alandaki çalışmalarda büyük gelişmeler yaşanmıştır. Gerçek anlamda parçacık filtresinin ortaya çıkması 1993 yılında olmuştur [3]. 1993 yılında Gordon, Salmond ve Smith tarafından ortaya atılan ve sıralı Monte Carlo filtresi olarak bilinen bu filtre, aynı zamanda parçacık filtresi olarak isimlendirilmektedir. Günümüzde hedef takibi, ekonomi, sinyal işleme, biyomedikal uygulamalar gibi çeşitli alanlarda aktif olarak kullanılan farklı parçacık filtreleri vardır ve hepsinin temeli Gordon ve arkadaşlarının çalışmasına dayanmaktadır [3].

Takip edilmesi planlanan hedeflere ait her bir hareketin modellenmesi ve kullanılacak olan parçacık filtresine bu modellerin tanıtılması hedefin tam olarak izlenebilmesi açısından önemlidir. Parçacık filtresine tanıtılan model sayısı arttıkça hedefin tam olarak izlenebilmesi sağlanır. Bu amaç ile çoklu model parçacık filtreleri (ÇMPF) sıklıkla kullanılmaktadır. Literatürde bulunan çoklu model parçacık filtrelerinin çalışma prensibi parçacık filtresinde yer alan tüm parçacıkların her zaman adımı için her bir modelde ilerletilmesidir [4; 5; 6; 7; 8]. Fakat bu işlem algoritma hızını düşürmekte ve işlem birimini gereksiz yere meşgul etmektedir. Bu

problemi çözmek amacıyla yani işlem hızını arttırabilmek amacıyla bu çalışmada ağırlıklandırılmış istatistiksel model seçimi (AIMS) algoritması olarak isimlendirilen yeni bir yaklaşım önerilmiştir. Bu algoritma ile her bir zaman adımında tüm modellerin hesaplanması yerine, geçmiş model seçimlerinden elde edilen istatistiksel bilgileri kullanılarak bir sonraki adımda hangi modelin kullanılması gerektiği tahmin edilmektedir. Bu sayede her bir zaman adımında tek bir modelin hesaplanması ile işlem süresi etkin bir biçimde kullanılmaktadır.

Tez çalışmasında önerilen AİMS algoritması, başarısının değerlendirilmesi amacıyla üç farklı ÇMPF üzerinde kullanılmış olup, birbirinden farklı gürültü ve senaryolar ile test edilmiştir. ÇMPF'lerden biri, bu tez çalışması için tarafımızdan önerilen yeni bir çoklu model parçacık filtresidir. Filtrelerden ikisine önerdiğimiz AİMS algoritması entegre edilerek, elde edilen sonuçlar işlem süresi ve tahmin hatası kriterleri esas alınarak karşılaştırılmıştır. Kullanılan ÇMPF'ler işlem süresi ve tahmin hatası bakımından etkileşimli çoklu model (EÇM) ile de karşılaştırılmıştır.

2. TAKİP ALGORİTMALARI

Birçok takip algoritması durum uzay modellerini kullanır. Bu modeller genellikle istatistiksel tabanlı modellerdir. İstatistiksel modeller; teorik analizleri, ölçüm değerlerini ve model belirsizliklerini göz önünde bulundurarak, takip edilen nesnenin durum tahmini ile nesne takip problemini çözmeye çalışırlar. Bu kısımda takip algoritmalarından Bayesian Filtresi, Monte Carlo Yöntemi, Kalman filtresi ve Parçacık Filtresi açıklanmıştır.

2.1. Bayesian Filtresi

Bayesian teoremi, bir durumun ortaya çıkma olasılığının edinilen ek bilgiler ile nasıl değişeceğini gösteren bir teoremdir.

Dinamik sistemlerde iki farklı fonksiyon mevcuttur. Bunlar sırası ile durum fonksiyonu ve gözlem (ölçüm) fonksiyonudur. Durum fonksiyonu hedefin gerçekte nerede olduğunu belirten fonksiyondur. Gözlem (ölçüm) fonksiyonu ise gözlemci tarafından elde edilen ölçümlerin belirtildiği fonksiyondur.

Durum ve ölçüm vektörleri sırasıyla Eşitlik 2.1 ve Eşitlik 2.2'deki gibi tanımlanır. T periyodu, x durum değişkenlerini, z ise gözlem değişkenlerini ifade eder.

$$\{x_{1T}, x_{2T}, \dots, x_{kT}\} = \{x_1, x_2, \dots, x_k\} = x_{1:k} \quad (2.1)$$

$$\{z_{1T}, z_{2T}, \dots, z_{kT}\} = \{z_1, z_2, \dots, z_k\} = z_{1:k} \quad (2.2)$$

Durum fonksiyonunu tanımlamak amacıyla x durum vektörü tanımlanmış olsun. Buna göre k anı için durum vektörü Eşitlik 2.3'teki gibi tanımlanır. Bu ifadede x_k k anındaki durum vektörünü, x_{k-1} bir önceki andaki durum vektörünü, f_x durum geçiş fonksiyonunu, u_k ise durum gürültü vektörünü gösterir.

$$x_k = f_x(x_{k-1}, u_k) \quad (2.3)$$

Gözlem fonksiyonunu tanımlamak amacıyla z gözlem vektörü tanımlanmış olsun. k süresi için gözlem vektörü Eşitlik 2.4'teki gibi tanımlanır. Eşitlik 2.4'te z_k k anındaki

gözlem değerlerini ifade ederken, f_z gözlem fonksiyonunu, v_k gözlem gürültüsünü ifade eder.

$$z_k = f_z(x_k, v_k) \quad (2.4)$$

Takip algoritmalarının amacı, k anına kadar alınan tüm gözlem değerini hesaba katarak x_k durum değişkenini tahmin etmektir [9]. Yani $z_{1:T} = \{z_1, \dots, z_T\}$ gözlem değerlerinden $x_{0:T} = \{x_0, \dots, x_T\}$ durum değerleri tahmin edilmektedir. Bu tahmin işlemi sonsal olasılık yoğunluk fonksiyonun bütün gözlemler için hesaplanması ile bulunur ve Bayesian filtresi için iki adımda yapılır. Bunlar tahmin ve güncelleme (düzeltme) adımlarıdır.

Genel Bayesian denklemi Eşitlik 2.5'te verilmiştir. Eşitlikte yer alan x durum vektörünü, z gözlem değerlerini ifade etmektedir.

$$p(x | z) = \frac{p(z | x)}{p(z)} p(x) \quad (2.5)$$

$k=0$ anı için tahmin adımı Eşitlik 2.6'da, güncelleme adımı ise Eşitlik 2.7'de verilmiştir [10].

$$p(x_0 | z_0) = \frac{p(z_0 | x_0)}{p(z_0)} p(x_0) \quad (2.6)$$

$$p(x_1 | z_0) = \int p(x_1 | x_0) p(x_0 | z_0) dx_0 \quad (2.7)$$

$k>0$ anı için tahmin adımı Eşitlik 2.8, güncelleme adımı ise Eşitlik 2.9'da verilmiştir [10].

$$p(x_k | z_{1:k}) = \frac{p(z_k | x_k)}{p(z_k | z_{1:k-1})} p(x_k | z_{1:k-1}) \quad (2.8)$$

$$p(x_{k+1} | z_{1:k}) = \int p(x_{k+1} | x_k) p(x_k | z_{1:k}) dx_k \quad (2.9)$$

Eşitlik 2.8 ve Eşitlik 2.9'da yer alan olasılık dağılım fonksiyonlarında;

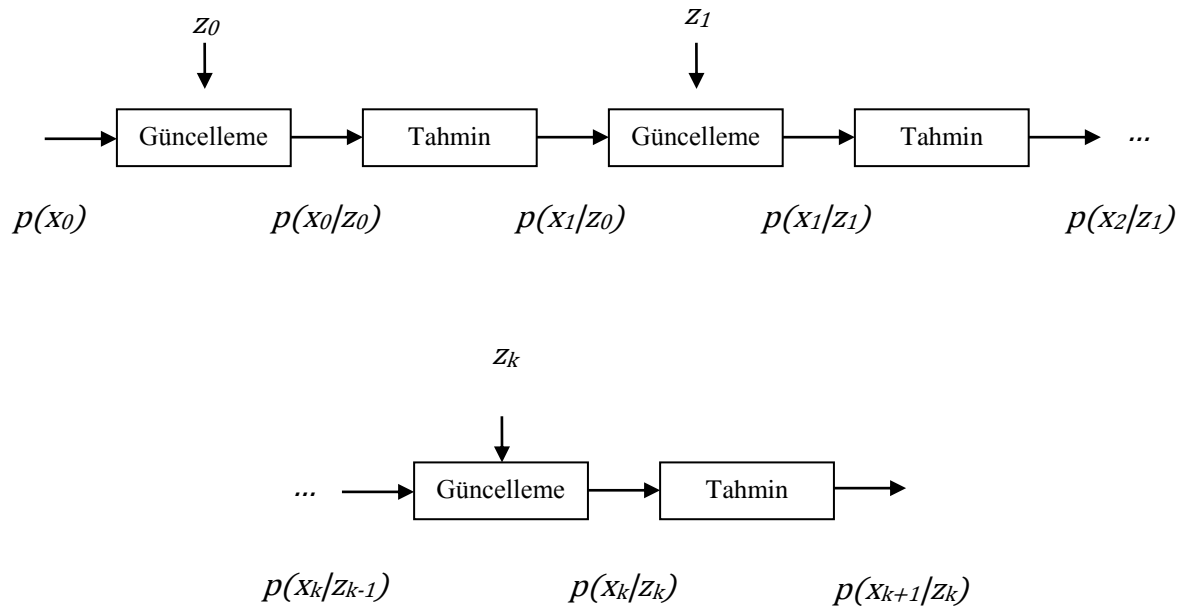
$p(x_k|z_{1:k})$ sonsal olasılık yoğunluk fonksiyonu: $z_{1:k}$ süresince elde edilen gözlem değerleri göz önünde bulundurularak, hedefin bulunabileceği x_k değerlerinin bütün x_k değerlerine göre olasılığını,

$p(x_{k+1}|x_k)$ önsel olasılık yoğunluk fonksiyonu: Hedefin hareket modelini temsil eder. k anındaki x değerine göre hedefin $k + 1$ anında nerede bulunabileceğinin olasılığını,

$p(z_k|x_k)$ olabilirlik fonksiyonu: x_k konum bilgisi bilinen bir hedefin z_k gözlem değerinin olabilirliğini,

$p(z_k|z_{1:k-1})$ standartlaştırma katsayısını ifade eder.

Tahmin ve güncelleme adımlarının zamana bağlı olarak hesaplanma şekli Şekil 2.1'de verilmiştir [10]. Şekil 2.1'de verilen Bayesian modeli özyinelemeli Bayesian modeline aittir.



Şekil 2.1 Bayesian Tahmin ve Güncelleme Adımları [10]

Bayesian filtresinde iki temel problem vardır. Birincisi, çözümün kavramsal olması yani matematiksel ifadelerin uygulanabilir olmayışıdır. İkincisi ise en yakın sonucun sadece az sayıdaki durum için geçerli olmasıdır.

2.2. Monte Carlo Yöntemi

Monte Carlo yöntemi 1930'lu yıllarda Los Alamos Laboratuvarlarında nükleer silah geliştirilmesi amacıyla projede çalışan bilim adamları tarafından ortaya atılmıştır. Monte Carlo yöntemi, gerçek hayattaki gelişmiş güzel davranış sergileyen problemlerde, matematiksel çözümlerinin ve uygulanmasının çok zor olduğu problemlerin çözümüne rasgele alınan örnekler ile yaklaşılmaya çalışılmasıdır. Bu yöntem; sayısal analiz, doğal olayların simülasyonu, nükleer fizik modellerinin simülasyonları, deneysel cihazların simülasyonları, borsa modelleri, istatistiksel dağılım fonksiyonlarının hesabında olmak üzere birçok alanda kullanılmaktadır.

Monte Carlo yöntemini diğer yöntemlerden ayıran en önemli özellikler, rastgele elemanlar içermesi, sistemin davranışını anlamaya yönelik olması ve matematiksel çözümlerin çok zor olduğu durumlarda kullanılmasıdır [11].

Monte Carlo yönteminde sistemde bulunan düzgün dağılımlı sayılar arasından rastgele seçimler yapılır. Seçilen bu sayılar problemin yapısına ve modele göre uyarlanır. Bu işlem için bütün değerler eşit olasılıkla seçilmeli ve seçilen her bir değer önceki değerden bağımsız olmalıdır. Monte Carlo yöntemine ait sistemi temsil edecek modelinin oluşturulabilmesi için izlenmesi gereken aşamalar aşağıda verilmiştir [11].

- Sistemin her bir rastgele elemanı için olasılık dağılım fonksiyonu hesaplanır.
- Hesaplanan olasılık dağılımları kümülatif olasılık dağılımına dönüştürülür.
- Rastgele sayılar kullanılarak, değişkenlerin değerleri hesaplanır.
- Sonuçlar sanal olay üzerinde denenir ve değerlendirilir.

Monte Carlo yönteminde kullanılan girdi değişkenlerinin geçmiş davranışları, gerçekleşen değerlere bakılarak olasılık dağılım fonksiyonlarına dönüştürülürler.

Bu işlemden sonra rastgele sayılar kullanılarak sistemin modeli işletilir ve sonuç elde edilir [4].

Rastgele x değişkenlerine ait olasılık dağılım fonksiyonu $p(x)$ Eşitlik 2.10'daki gibi ifade edilir.

$$p(x) \approx \sum_{i=1}^N w^i \cdot \delta(x - x^i) \quad (2.10)$$

Eşitlik 2.10'da yer alan N değeri toplam parçacık sayısını (rastgele oluşturulmuş ve ağırlıklandırılmış değişkenler), w her bir parçacığın ağırlığını, x^i parçacık konumunu, δ ise Dirac fonksiyonunu belirtir.

Monte Carlo yönteminde amaç Eşitlik 2.11'deki tahmin denklemini hesaplamaktır. Eşitlikte yer alan $p(x)$ olasılık dağılım fonksiyonunu ifade ederken, $f(x)$ sıklık fonksiyonunu ifade etmektedir.

$$\int f(x)p(x)dx \approx \sum_i f(x^{(i)}), \quad x^{(i)} \sim p(x) \quad (2.11)$$

Eşitlik 2.12'de herhangi bir bölgeye ait f fonksiyonun integrali verilmiştir.

$$\int f(x)dx \approx \sum_i f(x^{(i)}) \quad (2.12)$$

Eşitlik 2.12'de verilen ifade ağırlıklandırılmış olarak Eşitlik 2.13'teki gibi ifade edilebilir [4].

$$\hat{f} \approx \sum_{i=1}^N w^i \cdot f(x^i) \quad (2.13)$$

Tahmin edilecek değerlerin hesaplanmasında iki temel işlem vardır. Birincisi olasılık dağılım fonksiyonundan (Eşitlik 2.12) örnekleri çekmek, ikincisi ise Eşitlik 2.13'teki yaklaşık integral ifadesinin hesaplanmasıdır.

2.3.Önem Örneklemesi

Önem örneklemesinde hedef her bir parçacığa önemine uygun ağırlıkların verilmesidir. Bazı dağılımlarında örnekleme işlemi doğrudan kolay bir şekilde yapılabildiği gibi, bazılarında ise doğrudan örnekleme yapılamamaktadır. Doğrudan örnekleme yapılamayan dağılımlarda önem örneklemesi kullanılmaktadır. Eşitlik 2.10'da verilen ifadeye $q(x)$ önem dağılım fonksiyonun eklenmiş hali Eşitlik 2.14'te verilmiştir.

$$\int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx \quad (2.14)$$

Bütün örnekler önem dağılımından alınmaktadır. $q(x)$ önem dağılımdan N adet örnek alındığı kabul edilirse, önem dağılım fonksiyonu Eşitlik 2.15'teki gibi ifade edilebilir.

$$q(x) \approx \sum_{i=1}^N \delta(x - x^i) \quad (2.15)$$

Eşitlik 2.14 ve Eşitlik 2.15 kullanılarak tahmin denklemi Eşitlik 2.16'daki gibi yazılabilir [4].

$$\hat{f} \approx \sum_{i=1}^N \tilde{w}(x^i).f(x^i) \quad (2.16)$$

Eşitlik 2.16'da yer alan \tilde{w}^i önem ağırlığını ifade etmektedir. w^i ise önem ağırlığının standartlaştırılmış halidir. Önem ağırlığı Eşitlik 2.17'deki denklemdeki gibi ifade edilebilir [4].

$$\tilde{w}(x^i) \stackrel{\Delta}{=} \tilde{w}^i = \frac{p(x^i)}{q(x^i)} \quad (2.17)$$

\tilde{w}^i önem ağırlığının standartlaştırılması ile Eşitlik 2.18'deki tahmin denklemi yazılabilir [4].

$$\hat{f} = \frac{\sum_{i=1}^N \tilde{w}^i f(x^i)}{\sum_{i=1}^N \tilde{w}^i} = \sum_{i=1}^N w^i f(x^i) \quad (2.18)$$

2.4. Kalman Filtresi

Kalman filtresi, Gauss dağılımına sahip gürültü içeren doğrusal sistemlerde tahmin gerçekleştirmek için kullanılır. 1960 yılında Macar asıllı Amerikan matematiksel sistem teoristi Rudolf Kalman tarafından ortaya atılmıştır.

Kalman filtresinin en önemli özelliği modellenen sisteme ait geçmiş, şimdiki ve gelecekteki durumların öngörülebilmesidir. Hatta bu özellik tam olarak modellenememiş sistemler için de geçerlidir [12]. Kalman filtresinde sistemin doğrusal bir sistem olduğu, ölçümlerin ve gürültülerin Gauss dağılımına sahip olduğu varsayılır. Bu nedenle Kalman filtresinin en büyük dezavantajı Gauss dağılımına sahip ölçümler ile kullanılabilmesidir. Bundan dolayı kullanım alanı sınırlıdır.

Kalman filtresi iki farklı denklem grubunu içermektedir. Bunlar zaman güncellemesi ve ölçüm güncellemesi denklemleridir. Zaman güncellemesinde, şimdiki durumdan bir sonraki durum tahmin edilir ve oluşan hatanın kovaryansı hesaplanır. Ölçüm güncellemesinde ise Kalman kazancı hesaplanarak, son ölçüm değeri ile durum güncellemesi yapılır ve ardından hata kovaryansı güncellenir. Kısacası ölçüm güncellemesi ile tahminde performans iyileştirmesi yapılmış olur.

Kalman filtresinde zaman güncellemesine ait denklemler Eşitlik 2.19 ve Eşitlik 2.20'de; ölçüm güncellemesine ait denklemler Eşitlik 2.21, Eşitlik 2.22 ve Eşitlik 2.23'de verilmiştir.

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_{k-1} \quad (2.19)$$

$$P_k = AP_{k-1}A^T + Q \quad (2.20)$$

\hat{x}_k durum tahmini Eşitlik 2.19 ile bulunur. Durum tahmini sırasında önceki durum değeri olan \hat{x}_{k-1} ve u_{k-1} önceki durum değerine ait olan kontrol sinyal değeri kullanılır.

P_k hata kovaryansını (hatanın değişme miktarı) ifade etmektedir ve Eşitlik 2.20 ile bulunur. Hata kovaryans hesabı sırasında önceki hata kovaryansı olan P_{k-1} ile Q işlem gürültüsü kovaryansı kullanılır. İşlem gürültüsünün Gauss dağılımlı olduğu varsayılır.

Eşitlik 2.19 ve Eşitlik 2.20 denklemlerinde yer alan A ve B durum matrislerini ifade etmektedir. Çoğu problem çözümünde sadece birer nümerik bir değer olmasının yanında bu değerlerin problemlerde sabit olduğu varsayılabilir.

Eşitlik 2.21 Kalman kazancını ifade eder. Eşitlik 2.22'de son ölçüm değeri ve Kalman kazancı kullanılarak durum güncellemesi yapılırken, Eşitlik 2.23 ile hata kovaryansı güncellenir.

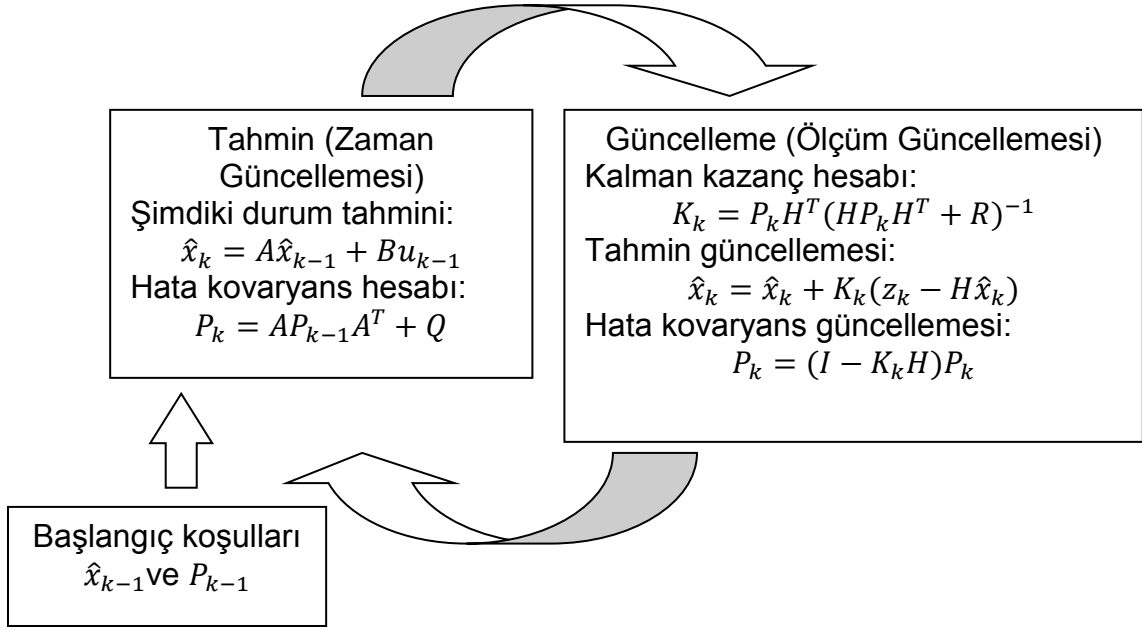
$$K_k = P_k H^T (H P_k H^T + R)^{-1} \quad (2.21)$$

$$\hat{x}_k = \hat{x}_k + K_k (z_k - H \hat{x}_k) \quad (2.22)$$

$$P_k = (I - K_k H) P_k \quad (2.23)$$

Eşitlik 2.21'deki eşitlik Kalman kazancını ifade ederken eşitlikteki K_k Kalman kazancını, R ölçüm gürültüsü kovaryansını, H ise ölçme matrisini ifade eder. Eşitlik 2.22'de tahmin güncellemesi yapılmaktadır. Eşitlik 2.22'de yer alan z_k son ölçüm değeri, H ise ölçme matrisini, K_k Kalman kazancını, \hat{x}_k tahmin değerini ifade eder. Eşitlik 2.23'te ise hata kovaryans güncellemesi yapılmakta olup, ilgili eşitlikte H ise ölçme matrisini, K_k Kalman kazancını, I birim matrisi, P_k ise hata kovaryansını ifade etmektedir.

Kalman filtresine ait döngü Şekil 2.2'de verilmiştir.



Şekil 2.2 Kalman Filtresi Döngüsü [12]

2.5. Parçacık Filtresi

Kalman filtresi doğrusal sistemler ve Gauss dağılımına sahip durum değişkenleri (sistem gürültüsü) ile en iyi sonucu vermektedir. Bu nedenle Kalman filtresinin kullanım alanı doğrusal ve Gauss dağılımına sahip durum değişkenleri ile sınırlıdır. Sistem doğrusallıktan uzaklaştıkça ve durum değişkenlerinin Gauss olmayan bir dağılıma yaklaştıkça parçacık filtresi kullanmak bu problemlerin aşılmasını sağlayacaktır.

Kalman filtresi ile parçacık filtresinin karşılaştırılması Çizelge 2.1'de verilmiştir.

Çizelge 2.1 Kalman Filtresi ve Parçacık Filtresi Karşılaştırması [13; 14]

	Kalman Filtresi	Parçacık Filtresi
Durum-Gözlem Denklemleri	$x_{k+1} = F_k x_k + G_k u_k$ $y_k = H_k x_k + v_k$	$x_{k+1} = f_k(x_k, u_k)$ $y_k = h_k(x_k, v_k)$
Gürültü Tipi	Gauss	Rastgele
Sonuç	Tam sonuca yakın	Yaklaşık
Hesap Hızı	Hızlı	Yavaş
Çözüm Tipi	Analitik	Sıralı Monte Carlo

Parçacık filtreleri sıralı Monte Carlo kestirim yöntemi yardımıyla ağırlıklandırılmış çok sayıda parçacık (durum uzayından alınan örnekler) kullanılarak sonsal dağılım belirlenmeye çalışılır. Sistemde kullanılan parçacıklar izlenen hedefin dinamiklerine (modeline) göre ağırlıklandırılır ve bu parçacıklardan olabilirliği yüksek olan parçacıkların nesneye daha yakın olduğu varsayılır. Bu parçacıklar sistemde daha fazla kalırlar.

Parçacık filtresi gibi doğrusal olmayan sistemlerde çalışabilen filtreler için durum ve gözlem (ölçüm) denklemleri sırasıyla Eşitlik 2.24 ve Eşitlik 2.25'te verilmiştir.

$$x_{k+1} = f_k(x_k, u_k) \quad (2.24)$$

$$y_k = h_k(x_k, v_k) \quad (2.25)$$

Her iki eşitlikte k zaman değişkeni olarak verilmiştir. Eşitlik 2.24'teki x_k durumunu, u_k işlem gürültüsünü, $f_k(\cdot)$ ise sistem geçiş fonksiyonunu ifade etmektedir. Eşitlik 2.25'te ise y_k gözlem vektörünü, v_k gözlem gürültüsünü, $h_k(\cdot)$ gözlem fonksiyonunu ifade etmektedir. $f_k(\cdot)$ ve $h_k(\cdot)$ doğrusal olmayan fonksiyonlardır.

Parçacık filtresinde sonsal dağılıma yaklaşmak için k anındaki durum yoğunluğu ω ağırlığı ile ağırlıklandırılmış M adet parçacık ile ifade edilir. Ağırlıklar parçacığın önemini belirtir. k anında M adet parçacık içeren rastgele bir ölçüm Eşitlik 2.26'daki gibi tanımlanır.

$$\{x_{1:k}^{(m)}, w_k^{(m)}\}_{m=1}^M \quad (2.26)$$

Eşitlik 2.26'da belirtilen $x_k^{(m)}$ sistemde yer alan k anındaki m 'inci parçacığı belirtir. $x_{1:k}^{(m)}$ ise 1'den k anına kadarki yörüngedeki parçacıkları belirtmektedir. $w_k^{(m)}$ ise k anındaki parçacığın ağırlığını belirtmektedir.

Parçacıkların gözlem değerlerine göre olasılık dağılım fonksiyonu Eşitlik 2.27'deki gibidir. Eşitlik 2.27'de yer alan M değeri toplam parçacık sayısını (rastgele oluşturulmuş ve ağırlıklandırılmış değişkenler), w her bir parçacığın ağırlığını, x parçacık konumunu, z gözlem değerini δ ise Dirac fonksiyonunu belirtir.

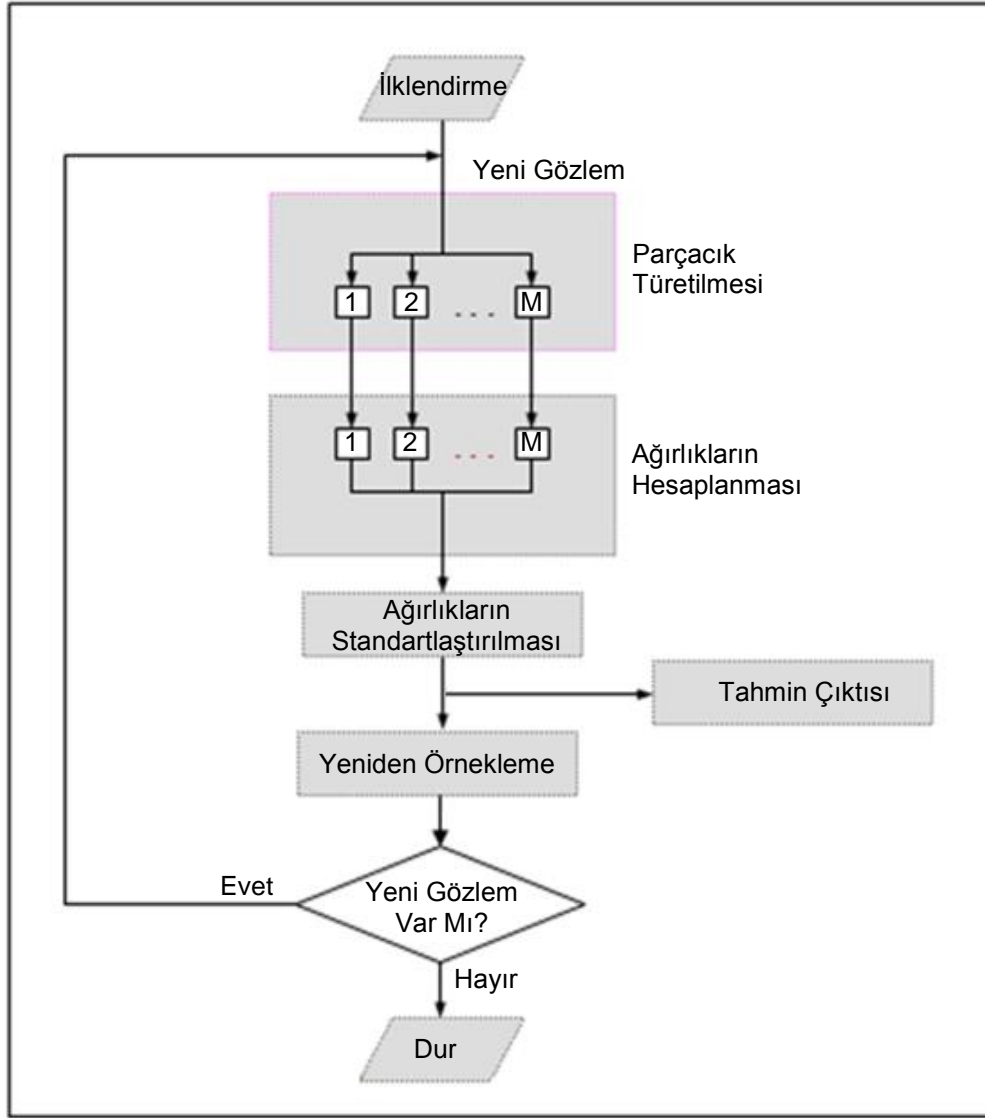
$$p(x_{1:k} | z_{1:k}) \cong \sum_{m=1}^M w_k^{(m)} \delta(x_{1:k} - x_{1:k}^{(m)}) \quad (2.27)$$

Parçacık filtresinin uygulanmasında üç temel adım vardır [15]. Bunlar;

- Parçacıkların oluşturulması (ilklendirme),
- Ağırlıklandırma (parçacıklara ait ağırlıkların hesaplanması (güncellenmesi) ve standartlaştırılması),
- Yeniden örneklemedir.

Parçacık filtresi için belirtilen üç temel adımı içeren akış diyagramı Şekil 2.3'te verilmiştir.

Parçacık filtrelerinde kullanılan önem örnekleme, Monte Carlo benzetim yöntemi ile Bayes filtrelemesinin yapılabilmesini sağlayan bir örnekleme çeşididir. Bu örnekleme ile dağılım parametreleri, sonsal dağılımın ağırlıklandırılmış rastgele seçilen parçacıkları kullanılarak Monte Carlo yöntemi ile kestirilebilmektedir [16].



Şekil 2.3 Parçacık Filtresi Akış Diyagramı [10]

2.5.1. Parçacıkların oluşturulması

Parçacıkların oluşturulma işlemi önem yoğunluk fonksiyonundan yararlanılarak yapılır. Bu nedenle önem yoğunluk fonksiyonun önemli bir yeri vardır. Önem yoğunluk fonksiyonuna ait Eşitlik 2.28'de verilmiştir. Eşitlik 2.28'de yer alan x durum değerini, z ise gözlem değerini ifade etmektedir. Bu eşitlik ile k anındaki duruma ait m adet farklı parçacık oluşturulmaktadır.

$$x_k^{(m)} \cong q(x_k | x_{k-1}^{(m)}, z_{1:k}) \quad (2.28)$$

Eğer oluşturulan bütün parçacıklar sinyal uzayında yer alıyorsa, parçacıkların tahmin değerleri ve ağırlıkları zayıf olur ve parçacık filtresi performansı düşer. Fakat eğer parçacıklar önemli bir olasılık kümesinde yer alıyorsa, parçacık filtresinin performansı artar [10].

2.5.2. Ağırlıklandırma

Ağırlıkların hesaplanması ve ağırlıkların standartlaştırılması önem örnekleme yoluyla yapılmaktadır. Ağırlıklandırma adımı iki basamaktan oluşmaktadır. Bunlar, parçacıklara ait ağırlıkların hesaplanması ve standartlaştırılmasıdır.

Eşitlik 2.27'de yer alan ağırlık katsayısı yerine ağırlık katsayısını ifade eden Eşitlik 2.29 yazılabilir. Eşitlikte yer alan $p(x_{1:k}^{(m)} | z_{1:k})$ olasılık dağılım fonksiyonunu, $q(x_{1:k}^{(m)} | z_{1:k})$ önem yoğunluk fonksiyonları ifade etmektedir. x ve z daha önceki eşitliklerde belirtildiği gibi durum ve gözlem değerlerini ifade etmektedir.

$$w_k^{(m)} \cong \frac{p(x_{1:k}^{(m)} | z_{1:k})}{q(x_{1:k}^{(m)} | z_{1:k})} \quad (2.29)$$

Eşitlik 2.29'da k anındaki önem yoğunluk fonksiyonu çarpanlarına ayrıldığı takdirde Eşitlik 2.30 ortaya çıkmaktadır.

$$q(x_{1:k} | z_{1:k}) = q(x_k | x_{1:k-1}, z_{1:k}) \cdot q(x_{1:k-1} | z_{1:k-1}) \quad (2.30)$$

Yine aynı şekilde Eşitlik 2.29'da k anındaki sonsal olasılık yoğunluk fonksiyonu çarpanlarına ayrıldığı takdirde Eşitlik 2.31 ortaya çıkmaktadır [4].

$$\begin{aligned} p(x_{1:k} | z_{1:k}) &= \frac{p(z_k | x_{1:k}, z_{1:k-1}) \cdot p(x_{1:k} | z_{1:k-1})}{p(z_k | z_{1:k-1})} \\ &= \frac{p(z_k | x_{1:k}, z_{1:k-1}) \cdot p(x_k | x_{1:k-1}, z_{1:k-1})}{p(z_k | z_{1:k-1})} p(x_{1:k-1} | z_{1:k-1}) \end{aligned}$$

$$= \frac{p(z_k | x_k) \cdot p(x_k | x_{k-1})}{p(z_k | z_{1:k-1})} p(x_{1:k-1} | z_{1:k-1}) \quad (2.31)$$

Eşitlik 2.31 yaklaşık olarak Eşitlik 2.32'deki gibi yazılabilir [4].

$$p(x_{1:k} | z_{1:k}) \cong p(z_k | x_k) p(x_k | x_{k-1}) p(x_{1:k-1} | z_{1:k-1}) \quad (2.32)$$

Eşitlik 2.30 ve Eşitlik 2.32, Eşitlik 2.29'da yerlerine konulduğunda Eşitlik 2.33 elde edilir.

$$\begin{aligned} w_k^{(m)} &\cong \frac{p(z_k | x_k) \cdot p(x_k | x_{k-1}) \cdot p(x_{1:k-1} | z_{1:k-1})}{q(x_k | x_{1:k-1}, z_{1:k}) \cdot q(x_{1:k-1} | z_{1:k-1})} \\ &\cong \frac{p(z_k | x_k^{(m)}) \cdot p(x_k^{(m)} | x_{k-1}^{(m)})}{q(x_k^{(m)} | x_{1:k-1}^{(m)}, z_{1:k})} w_{k-1}^{(m)} \end{aligned} \quad (2.33)$$

Eşitlik 2.33'te güncellenen ağırlıkların standartlaştırılması gerekmektedir. Bunun için Eşitlik 2.34 kullanılır.

$$w_k^{*(m)} = \frac{w_k^{(m)}}{\sum_{i=1}^M w_k^{(m)}} \quad (2.34)$$

2.5.3. Yeniden örnekleme

Seçilen bazı parçacıkların ağırlıklarının zaman içerisinde çok büyümesi ve diğer ağırlıkların değerlerinin çok küçük kalması (sıfıra yaklaşması), diğer bir deyişle sonsal dağılımın az sayıda örnek ile temsil edilmesi sonsal dağılım kestiriminin doğruluğunun azalmasına yol açmaktadır. Bu durum bozulma olgusu olarak adlandırılır [16].

Eşitlik 2.35'te verilen eşitlik ile etkin örnek sayısının hesaplanabilmekte, bu sayede sonsal dağılım kestiriminin doğruluğunda meydana gelen bozulmanın miktarı tespit edilmektedir.

$$N_{eff} = \frac{1}{\sum_{i=1}^M (w_k^i)^2} \quad (2.35)$$

Etkin örnek sayısının zamanla düşmesi performansı olumsuz yönde etkileyeceğinden bu durumun engellenmesi gerekmektedir. Bunun için yeniden örnekleme yöntemi kullanılır.

Yeniden örnekleme yönteminde sıfıra yaklaşan ağırlıklar örnek kümesinden atılır ve yüksek ağırlıklı örnekler çoğaltılarak bu örneklerin yerine geçirilir. Fakat yeniden örnekleme yöntemi başka bir problemi de yanında getirmektedir. Bu probleme örnek fakirleşmesi adı verilir.

Örnek fakirleşmesinde yüksek ağırlığa sahip örneklerin birçok kez seçilmesi ve kullanılması, yeni oluşturulan örnek kümesinde çeşitliliğin azalmasına yol açacaktır. Bu durum bozulma olgusundaki gibi sonsal dağılım kestiriminin doğruluğunun azalmasına yol açacaktır. Yeni örneklerin sıfır ortalamalı (küçük varyanslı) Gauss dağılıma sahip gürültü örnekleri ile toplanması bu durum engellenebilir. Böylelikle kendisini tekrar eden örnekler engellenerek çeşitlilik korunmuş olur. Seçilen gürültünün sonsal dağılımı bozmayacak şekilde (küçük varyanslı) seçilmesi uygun olacaktır [16].

Sistemik yeniden örnekleme, sıralı önem yeniden örnekleme, artık yeniden örnekleme, çok terimli yeniden örnekleme, katmanlı yeniden örnekleme gibi birçok yeniden örnekleme yöntemi olmasına rağmen en sık kullanılan yöntemler sistemik yeniden örnekleme ve sıralı önem yeniden örneklemesidir. Bu çalışmada kolay uygulanabildiği için sıralı önem yeniden örnekleme metodu kullanılmıştır.

2.5.3.1. Sistemik yeniden örnekleme (SYÖ)

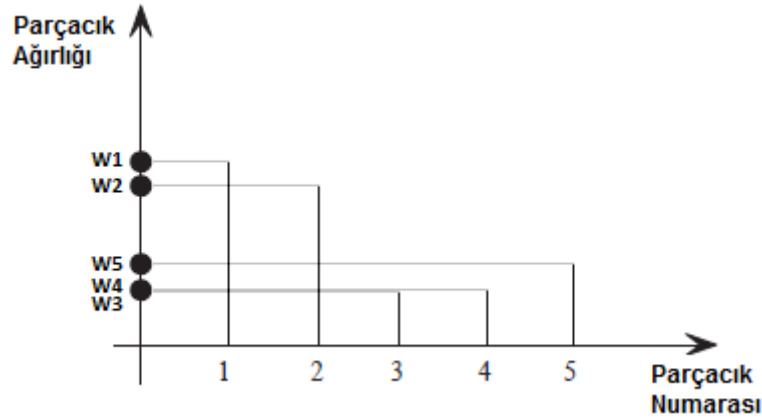
SYÖ'de örnekleme yapılacak aralık M adet katmana bölünmesi ve elde edilen her bir katmandan örnekleme yapılması mantığına dayanır [17].

Şekil 2.4 ve Şekil 2.5'te $M=5$ adet örnek için sistematik yeniden örnekleme örneği verilmiştir. Şekil 2.4'te mevcut parçacıkların ağırlıkları gösterilmektedir. Şekil 2.5'te yer alan $C^{(m)}$ Şekil 2.4'teki ağırlıkların kümülatif toplamını ifade etmektedir. Kümülatif toplam Eşitlik 2.36'da verilmiştir.

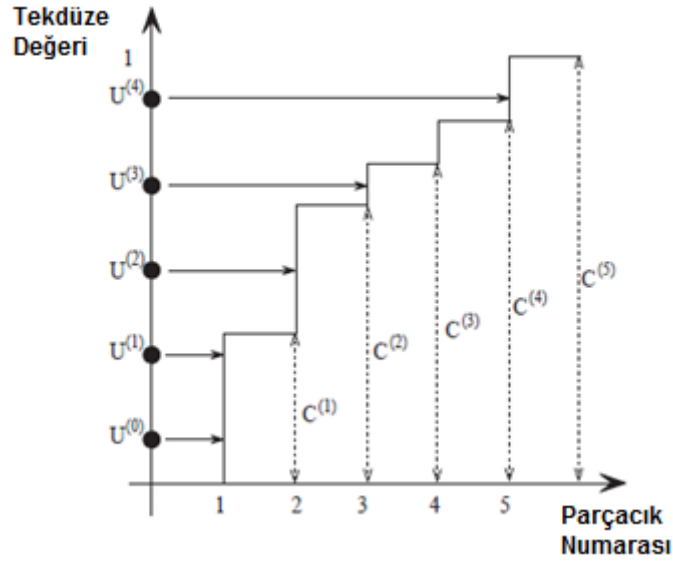
$$C^{(m)} = \sum_{i=1}^m w_n^{(i)} \quad (2.36)$$

SYÖ yönteminde bulunan $C^{(m)}$ değeri $U^{(m)}$ tekdüze değerleri ile karşılaştırılır. İlk tekdüze değeri olan $U^{(0)}$, $U\left[0, \frac{1}{M}\right]$ tekdüze dağılımından seçilir ve her bir m için $U^{(m)} = U^{(m-1)} + 1/M$ eşitliği ile güncellenir. Seçilen her bir m parçacığı $[C^{(m-1)}, C^{(m)})$ aralığında yeniden örneklenir.

Şekil 2.5'te görüldüğü gibi birinci parçacık $[0, C^{(1)})$ aralığında $U^{(0)}$ ve $U^{(1)}$ olmak üzere iki kez yeniden örneklenmiştir. İki ve üçüncü parçacık birer kez yeniden örneklenirken, dördüncü parçacık için herhangi bir işlem yapılmamıştır. Bunun nedeni dördüncü parçacığın yeniden örnekleme faktörünün ($r^{(m)}$) yani kaç kere kopyalanacağına ilişkin değerın sıfır olmasıdır.

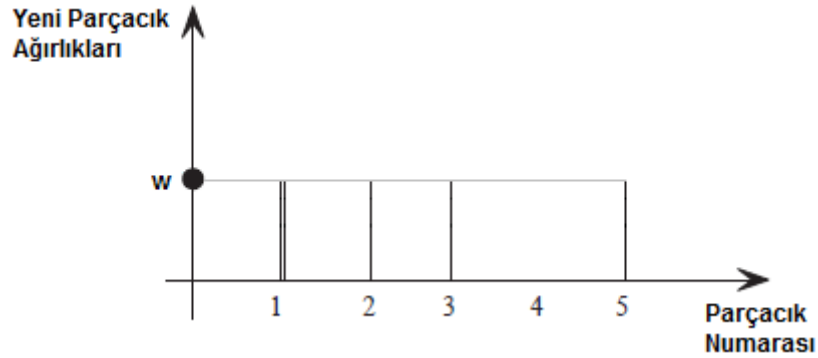


Şekil 2.4 Parçacık Ağırlık Grafiği [4]



Şekil 2.5 Sistematik Yeniden Örnekleme [4]

Yeniden örnekleme sonucunda elde edilen yeni parçacık ağırlık grafiği Şekil 2.6'da verilmiştir. Grafiğe göre 4 numaralı parçacık dışındaki diğer tüm parçacıklar yeniden örneklenmiştir.



Şekil 2.6 Sistematik Yeniden Örnekleme Sonrası Parçacık Ağırlık Grafiği [4]

SYÖ'ye ait sözde kodu aşağıda verilmiştir [15]. Kodda yer alan N sisteme giriş yapan parçacıkların toplam sayısını, M yeniden örnekleme sonrası çoklanarak üretilen parçacık sayısını, w standartlaştırılmış ağırlıkları, r ise yeniden örnekleme faktörünü ifade etmektedir.

$(r) = SR(w, N, M)$

$U \sim U(0, 1/M]$ // İlk tekdüze değerini rastgele seç

$s = 0$ // İlk kümülatif ağırlık değerini sıfırla

for $m = 1 : N$ // Ana döngü

$k = 0$ // Yeniden örnekleme faktörü sayacını sıfırla

$s = s + w^{(m)}$ // Kümülatif ağırlık değerini güncelle

while ($s > U$) // Yeniden örnekleme döngüsü

$k = k + 1$ // Yeniden örnekleme faktörünü bir arttır

$U = U + 1/M$ // Tekdüze değerini güncelle

end

$r^{(m)} = k$ // Örnekleme faktörlerini sakla

end

2.5.3.2. Sıralı önem yeniden örnekleme (SÖYÖ)

SÖYÖ yöntemi önem örnekleme tabanlı yeniden örnekleme metodudur [18]. SÖYÖ'nün ana hedefi, düşük ağırlıklı parçacıkları ortadan kaldırarak bunların yerine daha yüksek ağırlığa sahip olan parçacıkların kopyalanarak parçacık sayısının çoğaltılmasıdır. Bu sayede sistemde oluşabilecek sonsal dağılım kestiriminin doğruluğunun azalması (bozulma olgusu) engellenebilmektedir [19].

SÖYÖ'ye ait akış diyagramı aşağıda verilmiştir [20; 21]:

1. $x_0^{(i)} \sim p(x_0)$ 'dan örnekleme yap ($i = \{1, \dots, N\}$ için).

2. Parçacıklara ağırlık ata $w_0^{(i)} = p(y_0|x_0^{(i)})$ ve ağırlıkları normalize et $w_0^{(i)} = \frac{\tilde{w}_0^{(i)}}{\sum_{j=1}^N \tilde{w}_0^{(j)}}$ ($i = \{1, \dots, N\}$ için).

3. $\{w_{t-1}^{(i)}\}_{i=1}^N$ ağırlıkları yardımıyla $\{\tilde{x}_{t-1}^{(i)}\}_{i=1}^N$ 'den yeniden örnekleme işlemini gerçekleştir $\tilde{x}_{t-1}^{(i)}$ ($i = \{1, \dots, N\}$ ve $t = \{1, \dots, T\}$ için).

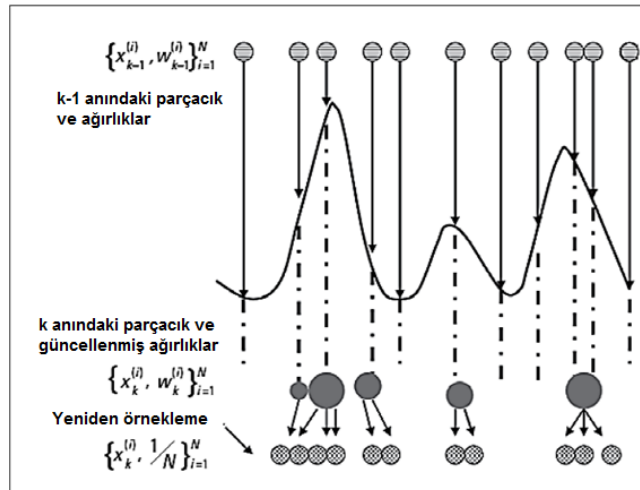
4. $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^N$ parçacık seti yerine yeniden örneklenen $\{\tilde{x}_{t-1}^{(i)}, \frac{1}{N}\}_{i=1}^N$ yeni seti aktar ($i = \{1, \dots, N\}$ ve $t = \{1, \dots, T\}$ için).

5. Yeni ağırlıkların üret $\tilde{w}_t^{(i)} = \frac{p(y_t|x_t^{(i)})p(x_t^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{0:t})}$ ($i = \{1, \dots, N\}$ ve $t = \{1, \dots, T\}$ için).

6. Ağırlıkları normalize et $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^N \tilde{w}_t^{(j)}}$ ($i = \{1, \dots, N\}$ için).

7. Tahmin değerinin elde edilmesi $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ ($t \in \{0, T\}$ için).

SÖYÖ'nün basit gösterimi Şekil 2.7'de verilmektedir. $k-1$ anındaki parçacıklardan k anındaki parçacıklar elde edilmektedir. Ağırlığı baskın olan parçacıklar yeniden örneklenerek ağırlık değerleri sıfıra yakın parçacıkların yerine kopyalanmaktadır.

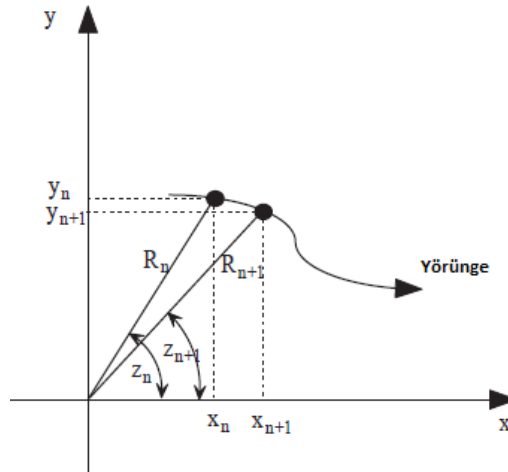


Şekil 2.7 Sıralı Önem Yeniden Örnekleme [22; 23]

2.6. Parçacık Filtresi ile Hedef Takibi

Parçacık filtresi ile hedef takibi dört ana bölümden meydana gelmektedir. Bunlar örnekleme adımı, ağırlıklandırma adımı, yeniden örnekleme adımı ve tahmin adımıdır. Örnekleme adımında hedefe ait konum ve hız bilgileri tahmin edilir. Fakat bu tahmin değerleri her bir parçacık için yapılan tahminler olup, hedefe ait nihai tahmin değeri değildir. Ağırlıklandırma adımında her bir parçacık olabilirliklerine göre ağırlıklandırılarak, hedefin muhtemel konumu tespit edilmeye çalışılır. Ağırlığı yüksek olan parçacıkların hedefe yakın olduğu düşünülür. Yeniden örnekleme adımında düşük ağırlıklı parçacıklar sistemden uzaklaştırılarak yerlerine yüksek ağırlıklı parçacıklar kopyalanır. Böylece hedefin nihai konum bilgisi daha doğru bir şekilde tahmin edilebilecektir. Son adım olan tahmin adımında ise ağırlıklar ise tahmin edilecek değerler çarpılarak her bir parçacık için kümülatif olarak toplanır. Bu sayede hedefe ait nihai konum bilgisi elde edilir.

Şekil 2.8'de bir hedefe ait k ve $k + 1$ zamanlı iki boyutlu yörünge bilgisi verilmiştir. Bu yörünge bilgisi radardan alınan hedefe ait açı ve yön bilgisini içermektedir. Şekilde R radar ile hedef arasındaki mesafeyi, z ise hedefin x eksenine yaptığı açıyı ifade etmektedir.



Şekil 2.8 Hedef Yörünge Bilgisi [15]

Hedefe ait Kartezyen koordinatlardaki durum vektörü Eşitlik 2.37'de verilmiştir.

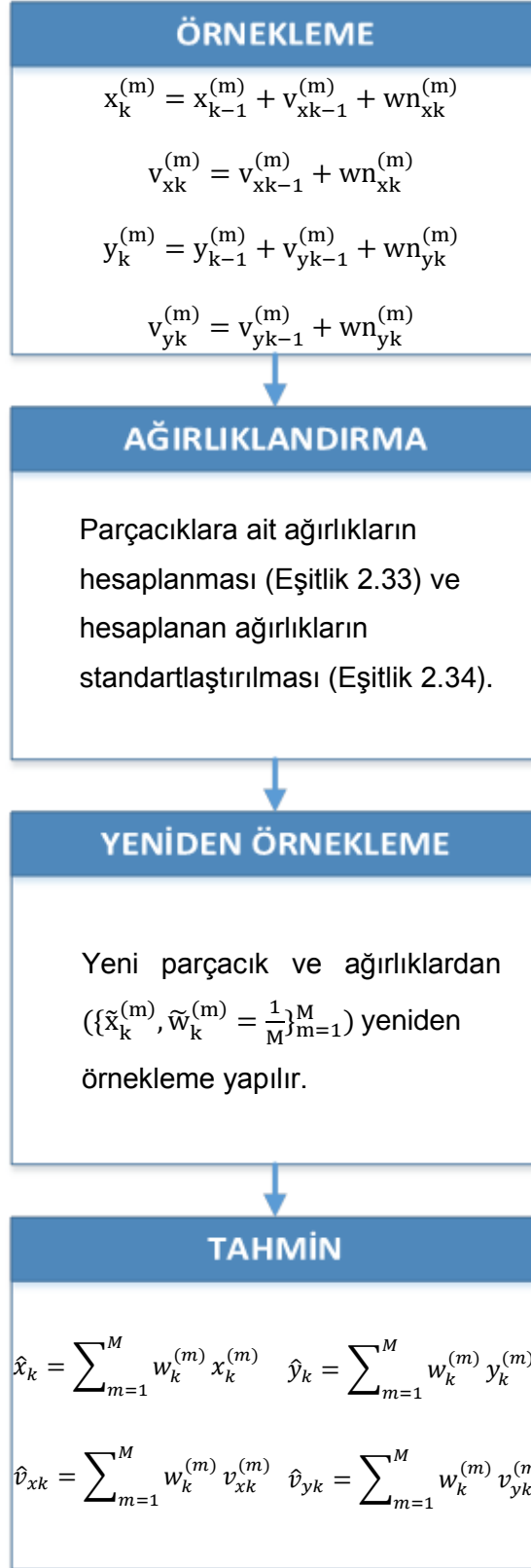
$$x_k = [x_k, v_{xk}, y_k, v_{yk}]' \quad (2.37)$$

İki boyutlu düzlemde hareket eden hedefe ait durum modeli Eşitlik 2.38'de verilmiştir.

$$x_k = Fx_{k-1} + Gwn_k \quad k = 1, \dots, N \quad (2.38)$$

Eşitlik 2.38'de yer alan F ve G durum geçiş matrislerini ifade etmektedir. Eşitlikte yer alan x_k yerine eşitlik 2.37'deki ifade yazılabilir. wn_k beyaz gürültüyü ifade etmektedir.

Yukarıda yazılan bilgiler doğrultusunda hedef takibi için uyarlanmış standart bir parçacık filtresine ait akış diyagramı Şekil 2.9'da verilmiştir. Burada amaç hedefe ait koordinat ve hız bilgilerini tahmin etmek olup, gözlem bilgisi (z_k) ve parçacıklar ($\{x_{k-1}^{(m)}\}_{m=1}^M$) filtrede girdi olarak kullanılmıştır [15; 17]. Alınan örnekler ağırlıklandırılıp, yeniden örneklenerak tahmin değerleri elde edilmektedir.



Şekil 2.9 Parçacık Filtresi ile Hedef Takibi

3. TAKİP MODELLERİ

3.1. Hedef Takibinde Matematiksel Modeller

Hareketli hedeflerde hedefin en iyi şekilde takip edilebilmesi için hedefin her bir hareketinin modellenmesi ve gözlemlerden maksimum düzeyde bilgi edinilmesi gerekmektedir. Farklı hareket modelleri sayesinde mevcut sistemdeki belirsizlikler ve boşluklar giderilebilmektedir.

Hemen hemen bütün hareketli hedefler çeşitli modeller ile ifade edilebilir. Yani hedefe ait hızlanma, manevra yapma gibi her bir hareketi matematiksel olarak ifade edilebilmektedir. Bu kapsamda tanımlanacak durum uzay modeli Eşitlik 3.1 ve Eşitlik 3.2'deki gibidir [24].

$$x_{k+1} = f_k(x_k, u_k, wn_k) \quad (3.1)$$

$$z_k = h_k(x_k) + v_k \quad (3.2)$$

Burada x_k hedefin durum bilgisini, z_k hedefin gözlem bilgisini, u_k ise kontrol giriş vektörünü ifade etmektedir. k ise zamanı (ayrık zaman) bildirmektedir. wn_k ve v_k işlem ve ölçüm gürültülerini ifade etmektedir. Yukarıda belirtilen ayrık zamanlı bir model sürekli zamanlı bir modelden elde edilebilir. Sürekli zamanlı modele ait ifade Eşitlik 3.3 ve Eşitlik 3.4'te verilmiştir.

$$\dot{x}(t) = f(x(t), u(t), t) + wn(t), \quad x(t_0) = x_0 \quad (3.3)$$

$$z(t) = h(x(t), t) + v(t) \quad (3.4)$$

k anındaki durum yazılmak istenir ise; $x_k = x(t_k)$, $z_k = z(t_k)$, $v_k = v(t_k)$, $u_k = u(t_k)$, $h_k = h(x(t_k), t_k)$ ifadelerinin Eşitlik 3.3 ve Eşitlik 3.4'e yerleştirilmesi yeterlidir. Ayrık zamanlı modelin genel ifadesi Eşitlik 3.5 ve Eşitlik 3.7'de verilmiştir [24]. Eşitlik 3.5'te yer alan F_k durum matrisini, x_k tahmin vektörünü, G_k^u kontrol giriş matrisini, u_k kontrol giriş vektörünü, wn_k işlem gürültüsünü, G_k ise işlem gürültü

matrisini ifade etmektedir. Eşitlik 3.6'da yer alan ifadeler ise Eşitlik 3.5 ile benzer olup, sürekli zamana ait ifadelerdir.

Eşitlik 3.7'de ise H_k ise gözlem matrisini, x_k tahmin vektörünü, v_k ise gözlem gürültüsünü ifade etmektedir. Eşitlik 3.8'de yer alan ifadeler ise Eşitlik 3.7 ile benzer olup, sürekli zamana ait ifadelerdir.

$$x_{k+1} = F_k x_k + G_k^u u_k + G_k w n_k \quad (3.5)$$

$$\dot{x}(t) = A(t)x(t) + B^u(t)u(t) + B(t)w n_c(t), \quad x(t_0) = x_0 \quad (3.6)$$

$$z_k = H_k x_k + v_k \quad (3.7)$$

$$z(t) = C(t)x(t) + v(t) \quad (3.8)$$

Eşitlik 3.6 ve Eşitlik 3.8'de sürekli zamana ait hedef modeli ve gözlem modeli verilmiştir.

3.2. Sabit Hız Modeli

Sabit hız modelin hedefin düz bir hat üzerinde sabit hız ile yol aldığını ifade eden temel ve basit bir modeldir. Sabit hız modeli için hedefe ait konum ve hız bilgileri Kartezyen koordinatlar ile tanımlanabilmektedir. Eşitlik 3.9 ve Eşitlik 3.10'da sırası ile iki ve üç boyutlu durum vektörü tanımlanmıştır [25].

$$x = [x, s_x, y, s_y]' \quad (3.9)$$

$$x = [x, s_x, y, s_y, z, s_z]' \quad (3.10)$$

Eşitlik 3.9 ve Eşitlik 3.10'daki ifadelerde yer alan (x, y, z) konum bilgisini içerirken, $[s_x, s_y, s_z]'$ hız vektörünü ifade etmektedir.

Sabit hız modelinde sürekli zaman için hedefe ait durum uzayı eşitlik 3.11'de verilmiştir. Eşitlik 3.11'de yer alan A durum matrisini, B ise işlem gürültü matrisini ifade etmektedir.

$$\dot{x}(t) = Ax(t) + Bwn(t) \quad (3.11)$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Eşitlik 3.11'de verilen eşitliği ayrık zamanda tanımlayabilmek için Eşitlik 3.12 ve Eşitlik 3.13'te Euler özdeşliğinden yararlanılır. Bu eşitliklerden elde edilen ayrık zamanlı model Eşitlik 3.14'te verilmiştir. Eşitlik 3.12'de $F(t_{k+1}, t_k)$ olarak belirtilen ifade durum geçiş matrisidir. Durum geçiş matrisindeki t_{k+1} ve t_k zaman ifadelerinin farkı alınırsa Δt ifadesi bulunur. Durum geçiş matrisi bu durumda $F(\Delta t)$ olarak ifade edilebilir.

F_k durum geçiş matrisine uygulanan çözümün bir benzeri G_k durum geçiş matrisine uygulanır ise ilgili matrisin ayrık zamandaki ifadesi bulunur.

$$F_k^\Delta = F(t_{k+1}, t_k) = F(t_{k+1} - t_k) = F(\Delta t) = e^{A\Delta t} \quad (3.12)$$

$$G_k^\Delta = G(t_{k+1}, t_k) = \int_0^{\Delta t} e^{A(\Delta t - \tau)} D d\tau \quad (3.13)$$

Eşitlik 3.5'ten yararlanarak Eşitlik 3.14 ayrık zaman eşitliği yazılabilir. Eşitlik 3.14'te yer alan F_k ve G_k durum geçiş matrisleri olup, sırasıyla durum ve gürültü matrislerini ifade etmektedirler.

$$x_{k+1} = F_k x_k + G_k wn_k \quad (3.14)$$

$$F_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} I & \Delta t \cdot I \\ 0 & I \end{bmatrix}$$

$$G_k = \begin{bmatrix} \Delta t^2/2 & 0 \\ 0 & \Delta t^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} = \begin{bmatrix} (\Delta t^2/2).I & \\ & \Delta t.I \end{bmatrix}$$

F_k ve G_k durum geçiş matrisleri Eşitlik 3.14'teki yerlerine konulduğunda sabit hız modeline ait Eşitlik 3.15'teki ifade bulunur.

$$x_{k+1} = \begin{bmatrix} I & \Delta t.I \\ 0 & I \end{bmatrix} x_k + \begin{bmatrix} (\Delta t^2/2).I \\ \Delta t.I \end{bmatrix} u_k \quad (3.15)$$

Eğer Eşitlik 3.15 gözlemci için hareket modeli olarak kullanılıyor ise, u_k bilinen ivmeyi ifade eder. Eğer bu eşitlik hedef için hareket modeli olarak kullanılıyor ise u_k bilinmeyen ivmeyi ifade etmektedir [4].

3.3. Sabit İvme Modeli

Bu modelde ivmedeki değişimin sabit olduğu kabul edilir. Sabit ivme modeli için hedefe ait konum, hız ve ivme bilgileri Kartezyen koordinatlar ile Eşitlik 3.16 ve Eşitlik 3.17'de sırası ile iki ve üç boyutlu durum vektörü olarak tanımlanmıştır.

$$x = [x, s_x, a_x, y, s_y, a_y]' \quad (3.16)$$

$$x = [x, s_x, a_x, y, s_y, a_y, z, s_z, a_z]' \quad (3.17)$$

Eşitlik 3.16 ve Eşitlik 3.17 ifadelerinde yer alan (x, y, z) konum bilgisini içerirken, $[s_x, s_y, s_z]'$ hız vektörünü, $[a_x, a_y, a_z]'$ ise ivmeyi ifade etmektedir.

İvme modeline iki farklı yöntem kullanılmaktadır. Birinci model, beyaz gürültü modelidir. Bu modelde ivmenin türevi beyaz gürültüye eşit kabul edilir [24]. Beyaz gürültü modeline ait durum uzayı Eşitlik 3.19'da verilmiştir.

$$x_{k+1} = F_k x_k + w n_k \quad (3.19)$$

$$F_k = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

İkinci model, Wiener ivme modelidir. Bu modelde ivmedeki artışın beyaz gürültüye eşit olduğu kabul edilir [24]. Wiener ivme modeline ait durum uzayı Eşitlik 3.20'de verilmiştir.

$$x_{k+1} = F_k x_k + G_k w n_k \quad (3.20)$$

$$F_k = \begin{bmatrix} I & \Delta t \cdot I & (\Delta t^2/2) \cdot I \\ 0 & I & \Delta t \cdot I \\ 0 & 0 & I \end{bmatrix} \quad G_k = \begin{bmatrix} (\Delta t^2/2) \cdot I \\ \Delta t \cdot I \\ I \end{bmatrix}$$

3.4. Singer Hızlanma Modeli

Singer hızlanma modeli manevra yapan hedeflerin modellemesinde sıkça kullanılan popüler bir modeldir. Hedefin hızlanmasının ve hedefin manevrasının belirli bir zaman içerisinde olduğu kabul edilir. Eğer hedef t anında hızlanıyor ise, $t + \Delta t$ anında da hızlanmasının t anındaki hızlanmasına benzer olduğu kabul edilir [26]. Δt küçük bir zaman aralığını ifade etmektedir.

Bu modelde hedefin hızlanmasına ilişkin korelasyon fonksiyonu Eşitlik 3.21'deki gibi yazılır.

$$r(\Delta t) = E a(t) a(t + \Delta t) = \sigma^2 e^{-\alpha |\Delta t|} \quad (3.12)$$

σ^2 hedefin hızlanma varyansını, $a(t)$ hedefin ivmesini, α ise manevra zaman sabitinin tersini ($\alpha = 1/\Delta t_m$) ifade eder ve manevranın ne kadar sürdüğü ile ilişkilidir. $\alpha = 1/60$ yavaş dönüşü, $\alpha = 1/20$ kaçış manevrasını, $\alpha = 1$ atmosferik türbülansı ifade eder [26].

Hedefe ait hızlanma varyansı (σ^2) Eşitlik 3.22 yardımıyla hesaplanır.

$$\sigma^2 = \frac{a_{max}^2}{3} [1 + 4P_{max} - P_0] \quad (3.22)$$

Eşitlik 3.22'de a_{max} hedefin maksimum hızlanma miktarını ($-a_{max}$ ile $+a_{max}$ aralığında değer alabilir), P_{max} maksimum hızlanmaya ait olasılığı, P_0 ise hedefin hızlanmaya başlamadan önceki hızlanma olasılığını ifade eder.

Singer hızlanma modelinin ayrık zamandaki ifadesi, Eşitlik 3.20'den yararlanarak, Eşitlik 3.23'deki gibi yazılabilir.

$$x_{k+1} = \begin{bmatrix} I & \Delta t.I & (\Delta t^2/2).I \\ 0 & I & \Delta t.I \\ 0 & 0 & e^{-\alpha.\Delta t}I \end{bmatrix} x_k + \begin{bmatrix} (\Delta t^2/2).I \\ \Delta t.I \\ I \end{bmatrix} wn_k \quad (3.23)$$

Eğer manevra zaman sabiti Δt_m azalır, Singer hızlanma modeli sabit hız modeline yaklaşır. Manevra zaman sabiti Δt_m artarsa, Singer hızlanma modeli sabit ivme modeline yaklaşır [24].

Singer hızlanma modeli genel bir model olmasına rağmen sadece tekdüze ve hızlanan hedefler için kullanılabilir.

3.5. İki Boyutlu Hareket Modelleri

Daha önce belirtilen sabit hız, sabit ivme ve Singer hızlanma modelleri hedefin tek bir eksen boyunca yaptığı hareketi modellemekte olup, herhangi bir dönüş modelini ifade etmemektedir. Bu başlık altında koordineli dönüş modelleri incelenmektedir.

3.5.1. Dönüş oranı bilinen koordineli dönüş modeli

Koordineli dönüş modellerinde hedefin sabit hız, sabit irtifa ve sabit açısal dönüş oranı (ω) ile hareket ettiği varsayılır. Bu tip hareketler genellikle sivil hedeflere ait hareketlerdir.

Dönüş oranı bilinen koordineli dönüş modelinde açısal dönüş oranının (ω) bilindiği varsayılır ve durum vektörü Kartezyen koordinatlarda Eşitlik 3.9'daki gibi gösterilir.

Eşitlik 3.3 yardımıyla Eşitlik 3.24 yazılabilir [21]. Eşitlik 3.24'te yer alan $A(\omega)$ açısal dönüş oranına bağımlı durum geçiş matrisini ifade etmektedir.

$$\dot{x}(t) = (s_x, -\omega s_y, s_y, \omega s_x)'(t) + wn(t) = A(\omega)x + wn(t) \quad (3.24)$$

$$A(\omega) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\omega \\ 0 & 0 & 0 & 1 \\ 0 & \omega & 0 & 0 \end{bmatrix}$$

Yukarıda belirtilen koordineli dönüş modeli ω bilindiği sürece doğrusaldır [24].

Eşitlik 3.24'te verilen sürekli zaman koordineli dönüş modeli ayrık zamanda Eşitlik 3.25'teki gibi yazılır.

$$x_{k+1} = F_{ct}(\omega)x_k + wn_k \quad (3.25)$$

$$F_{ct}(\omega) = \begin{bmatrix} 1 & \frac{\sin\omega T}{\omega} & 0 & -\frac{1 - \cos\omega T}{\omega} \\ 0 & \cos\omega T & 0 & -\sin\omega T \\ 0 & \frac{1 - \cos\omega T}{\omega} & 1 & \frac{\sin\omega T}{\omega} \\ 0 & \sin\omega T & 0 & \cos\omega T \end{bmatrix}$$

Dönüş oranı (ω) hızın doğrusal olmayan bir fonksiyonu gibi yazılabilir. Dönüş oranına (ω) ait ifade Eşitlik 3.26'da verilmiştir.

$$\omega = \frac{\pm a_m}{\sqrt{(s_x)^2 + (s_y)^2}} \quad (3.26)$$

Eşitlik 3.26'da yer alan ifadede a_m manevra ivmelenmesini ifade etmekte olup, aldığı değerlere göre cismin hareket bilgisi aşağıdaki gibidir [4]:

- $a_m > 0$ saat yönünün tersi manevra,
- $a_m = 0$ düz uçuş (sabit hız modeli),
- $a_m < 0$ saat yönünde manevra.

Farklı doğrusal olmayan modeller, farklı manevra ivmelenmesi (a_m) ile hesaplanabilir [4].

3.5.2. Dönüş oranı bilinmeyen koordineli dönüş modeli

Dönüş oranı bilinen koordineli dönüş modelinde, dönüş oranı (ω) geçmiş değerler kullanılarak yeniden hesaplanmakta veya başlangıçta sabit bir değer atanmaktadır. Dönüş oranı bilinmeyen koordineli dönüş modelinde ise bu oranın modellenmesi ve her adımda yeniden tahmin edilmesi gerekmektedir. Modelleme Wiener işlemi veya birinci derece Markov işlemi ile yapılabilir. Bu işlemler ile k anındaki açısal dönüş oranı kullanılarak $k + 1$ anındaki açısal dönüş oranı tahmin edilebilir [4].

Wiener işlemi için Eşitlik 3.27 kullanılır.

$$\omega_{k+1} = \omega_k + wn_{\omega,k} \quad (3.27)$$

Birinci derece Markov işlemi için ise Eşitlik 3.28 kullanılır.

$$\omega_{k+1} = e^{-T/\tau_\omega} \omega_k + wn_{\omega,k} \quad (3.28)$$

Eşitlik 3.28'de yer alan τ_ω her bir tur için korelasyon zaman sabitini, wn beyaz gürültüyü ifade etmektedir.

4. ÇOKLU MODEL FİLTRELERİ

4.1. Çoklu Model Parçacık Filtresi (ÇMPF)

Hareketli hedefler nadiren tek bir model ile ifade edilirler, fakat birçok hedef rotaları boyunca birden fazla manevra çeşidi ile hareket ederler. Bu nedenle hedefin tam olarak ve doğru şekilde takip edilebilmesi için hedefin yapabileceği tüm olası manevraların takip algoritmasına tanıtılmış olması gerekmektedir. Bir ÇMPF'deki ana fikir, sisteme tanıtılmış tüm modellerin $k + 1$ anı için hesaplanması ve hedefe ait durumun tahmin adımının k anı için yapılmasıdır.

Bu tezde üç farklı ÇMPF incelenmiştir. Bu ÇMPF'lerden bir ve ikincisi sırasıyla [4; 5; 6] numaralı kaynaklardan alınmış iken, üçüncü yöntem bizim tarafımızdan önerilmektedir. Bu üç ÇMPF'nin ortak tarafı t anında sistemde tanımlı olan X adet modelin N adet parçacık için hesaplanmasıdır. Bu nedenle hesaplama aşamasında sistemde $X*N$ adet parçacık bulunmaktadır. Daha sonra elde edilen $X*N$ adetlik parçacık sayısı bir sonraki adımda kullanılmak üzere tekrar N adete düşürülmektedir.

4.1.1. Birinci algoritma

Bu algoritma Yıldırım ve Demirekler [4; 5] tarafından sunulmuştur. Bu algoritmaya göre, her zaman adımı içerisinde sistemde yer alan tüm parçacıklar tanımlanmış olan bütün sistem modelleri için ilerletilmektedirler. Bu işlem sonucunda elde edilen toplam parçacık sayısı başlangıçtaki parçacık sayısı (N) ile algoritmadaki model sayısının (X) çarpımı kadar olmaktadır. Elde edilen tüm parçacıklar uygunluklarına ve Markov geçiş olasılıklarına göre ağırlıklandırılmaktadırlar. Yeniden örnekleme adımına geçildiğinde ise elde edilen $X*N$ adetlik parçacık sayısı baskınlıklarına yani ağırlıklarına göre yeniden örneklenecek tekrar N ye düşürülmektedir. Yani $X*N$ parçacıktan daha önemli bölgede bulunan N adetlik parçacık seçilmektedir [7]. Algoritmanın her bir k anı için yaptığı işlemleri gösterir sözde kodu aşağıda verilmiştir.

$i=1:N$

$j=1:X$

- Önem yoğunluk fonksiyonundan parçacıkları oluştur:

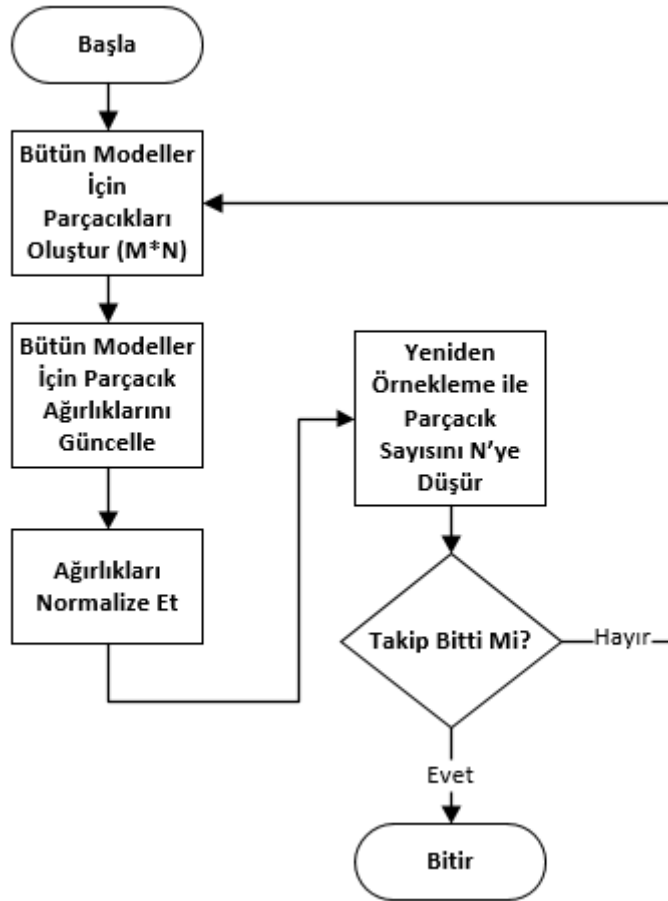
$$x_t^{(i)} \cong q(x_t^{(i,j)} | x_{t-1}^{(i)}, y_{1:t})$$

- Parçacıkların ağırlıklarını güncelle: $w_t^{(i,j)} \cong \frac{p(y_t | x_t^{(i,j)}) \cdot p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i,j)} | x_{1:t-1}^{(i)}, y_{1:t})} w_{t-1}^{(i,j)}$

- $M \times N$ parçacığın ağırlıklarını normalize et: $w_t^{(i,j)} = \frac{w_t^{(i,j)}}{\sum_{i=1}^X w_t^{(i,j)}}$

- Parçacıkları yeniden örnekleyerek toplam parçacık sayısını N 'ye düşür.

Algoritma 1'e ait akış diyagramı Şekil 4.1'de verilmiştir.



Şekil 4.1 Algoritma 1 Akış Diyagramı

4.1.2. İkinci algoritma

Bu algoritma Hong, Shi ve Chen [6] tarafından sunulmuştur. Bu algoritma tahmin adımına kadar algoritma 1'in yapısını kullanmaktadır. Her zaman adımı içerisinde sistemde yer alan tüm parçacıklar bütün sistem modelleri için ilerletilmektedirler. Tahmin işlemine geçildiğinde sistemde toplam $X*N$ adet parçacık ve hedefin konumuna ait N adet tahmin değeri bulunmaktadır. Hedefin konumuna ait tahmin değeri, yukarıda belirtilen N adet tahminin toplamı ile her bir tahmin için elde edilen ağırlıkların toplamının oranı ile bulunmaktadır. Algoritmanın her bir k anı için yapılan işlemlere ait sözde kod aşağıda verilmiştir.

i=1:N

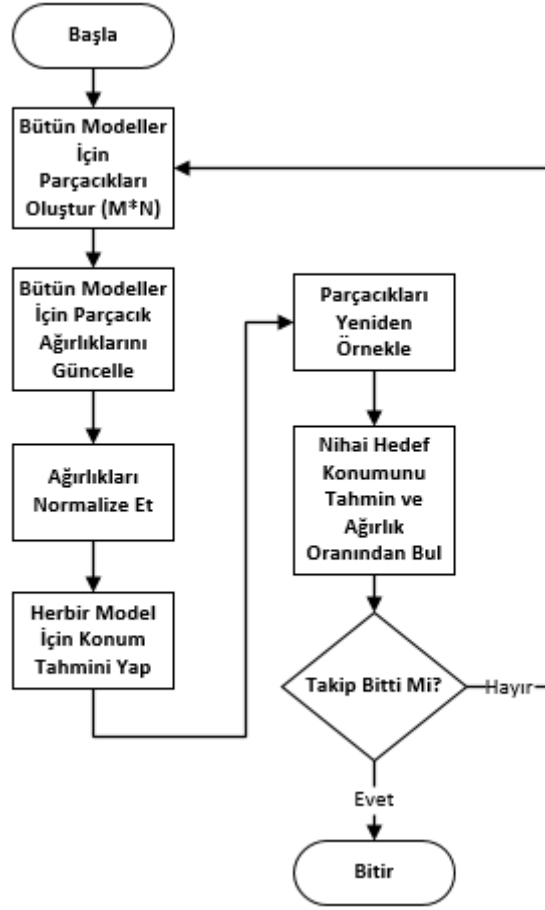
j=1:X

- Önem yoğunluk fonksiyonundan parçacıkları oluştur:

$$x_t^{(i)} \cong q(x_t^{(i,j)} | x_{t-1}^{(i)}, y_{1:t})$$

- Parçacıkların ağırlıklarını güncelle: $w_t^{(i,j)} \cong \frac{p(y_t | x_t^{(i,j)}) \cdot p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i,j)} | x_{1:t-1}^{(i)}, y_{1:t})} w_{t-1}^{(i,j)}$
- $M*N$ parçacığın ağırlıklarını normalize et: $w_t^{(i,j)} = \frac{w_t^{(i,j)}}{\sum_{i=1}^X w_t^{(i,j)}}$
- Her bir model için hedef konumu olan x_t 'yi tahmin et.
- Parçacıkları yeniden örnekle.
- Nihai hedef konumunu hesapla: $x_t = (x_t^1 + \dots + x_t^j) / (w_t^1 + \dots + w_t^j)$

Algoritma 2'ye ait akış diyagramı Şekil 4.2'de verilmiştir.



Şekil 4.2 Algoritma 2 Akış Diyagramı

4.1.3. Üçüncü algoritma

Bu algoritma Algoritma 2'ye alternatif olarak kullanılmak üzere tarafımızdan önerilmiştir. Algoritma 3 tahmin adımına kadar Algoritma 2'nin yapısını kullanmakla beraber, Algoritma 2'den farklı olarak en az hata miktarına sahip model bir sonraki adım için kullanılmak üzere seçilir. İkinci algoritmadaki gibi her zaman adımı içerisinde sistemde yer alan tüm parçacıklar bütün sistem modelleri için ilerletilir ve tahmin işlemine geçildiğinde sistemde toplam $X*N$ adet parçacık ve hedefin konumuna ait N adet tahmin değeri bulunur. t anı için elde edilen N adet tahminden hangisinin doğru olduğunu tespit edebilmek için $t-1$ anındaki tahminler ile $t-1$ anındaki gözlem değeri arasındaki hata miktarına bakılır. En az hata miktarına sahip modelin bir sonraki adımda da gelme olasılığının yüksek olacağı varsayımından hareketle ötürü t anında bu modelin tahmin değeri kullanılır.

Algoritmanın her bir k anı için yaptığı işlemleri gösterir sözde kodu aşağıda verilmiştir.

$i=1:N$

$j=1:X$

- Önem yoğunluk fonksiyonundan parçacıkları oluştur:

$$x_t^{(i)} \cong q(x_t^{(i)} | x_{t-1}^{(i)}, y_{1:t})$$

- Parçacıkların ağırlıklarını güncelle: $w_t^{(i,j)} \cong \frac{p(y_t | x_t^{(i,j)}) \cdot p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i,j)} | x_{1:t-1}^{(i)}, y_{1:t})} w_{t-1}^{(i,j)}$

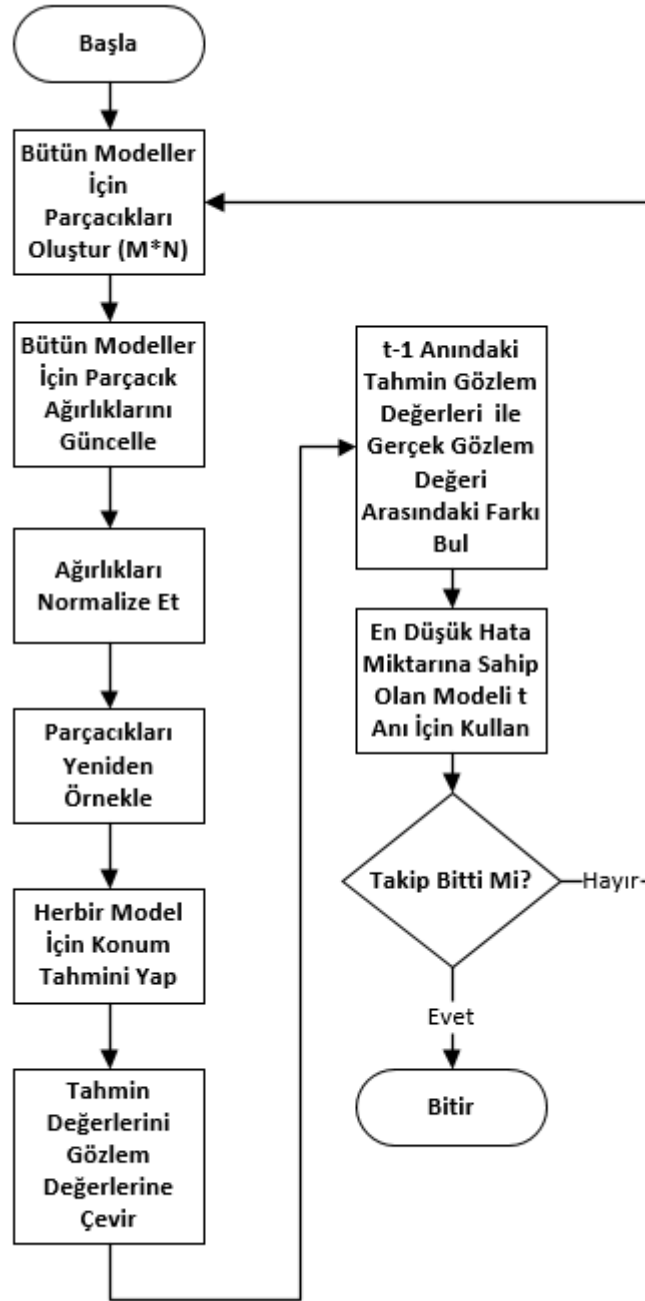
- $M*N$ parçacığın ağırlıklarını normalize et: $w_t^{(i,j)} = \frac{w_t^{(i,j)}}{\sum_{i=1}^X w_t^{(i,j)}}$

- Parçacıkları yeniden örnekle.

- Her model için elde edilen tahmin değerinden gözlem değerini (x ve y koordinatlarından açı ve mesafe bilgisi) çek: $\theta = \tan^{-1}\left(\frac{x}{y}\right)$, $R = \sqrt{x^2 + y^2}$

- $t-1$ anındaki mevcut gözlem değeri ile model tahminlerinden elde edilen gözlem değerleri arasındaki hata miktarlarını hesapla.
- En düşük hata miktarına sahip modeli t anı için kullanılmak üzere seç.

Algoritma 3'e ait akış diyagramı Şekil 4.3'te verilmiştir.



Şekil 4.3 Algoritma 3 Akış Diyagramı

4.2. Etkileşimli Çoklu Model (EÇM)

Durum kestiriminde kullanılan bir diğer yöntem ise etkileşimli çoklu model (EÇM) yöntemidir. Bu yöntemin çoklu model olarak tanımlanmasının nedeni sistemde yer alan sonlu sayıdaki modeller arasında geçiş yaparak sistem parametrelerini yani

durum kestirimini yapabilmesidir. Bu algoritmada durum kestirimi için Kalman filtresinden yararlanılmakta ve bu amaç ile kullanılabilir birden fazla Kalman filtresi türü bulunmaktadır.

EÇM algoritması dört adımdan meydana gelmektedir. Bunlar etkileşim adımı, model eşleme filtresi adımı, model olasılığının güncellenmesi adımı ve kestirim ve kovaryans kombinasyonu adımıdır [27].

4.2.1. Etkileşim adımı

Filtre güncellemesi öncesinde, durum kestirimi ve durum kovaryans değerleri bu adımda hesaplanır. İlgili hesaplamalar Eşitlik 4.1 ve Eşitlik 4.2’de verilmiştir [27; 28].

$$\hat{x}^{oj}(k-1|k-1) = \sum_{i=1}^r \hat{x}^i(k-1|k-1) \cdot \mu_{ij}(k-1|k-1) \quad (4.1)$$

$$P^{oj}(k-1|k-1) = \sum_{i=1}^r \mu_{ij}(k-1|k-1) \cdot \{P^i(k-1|k-1) + [\hat{x}^i(k-1|k-1) - \hat{x}^{oj}(k-1|k-1)] [\hat{x}^i(k-1|k-1) - \hat{x}^{oj}(k-1|k-1)]^T\} \quad (4.2)$$

Eşitlik 4.1’de x durum kestirimini, k süreyi; Eşitlik 4.2’de ise P durum kovaryans matrisini, μ model olasılığını ifade eder. Model olasılığı Eşitlik 4.3’te verilmiştir.

$$\mu_{ij}(k-1|k-1) = P\{M_i(k-1) | M_j(k), Z^{k-1}\} = \frac{1}{c} p_{ij} \mu_j(k-1) \quad (4.3)$$

Eşitlik 4.3’te yer alan M modu, c normalizasyon faktörünü ifade eder. Normalizasyon faktörüne ait eşitlik Eşitlik 4.4’te verilmiştir.

$$\bar{c}_j = \sum_{i=1}^r p_{ij} \mu_j(k-1) \quad (4.4)$$

4.2.2. Model eşleme filtresi

Eşitlik 4.1 ve Eşitlik 4.2'den elde edilen durum kestirimi ve durum kovaryans değerleri Kalman filtresinde kullanılarak, k anı için $M_j(k)$ modeli, $x^j(k|k)$ durum kestirimi ve $P^j(k|k)$ kovaryans değeri eşlenir [27]. Eşleme işlemi ile $\lambda_j(k)$ model benzerliği hesaplanır. Model benzerliği Eşitlik 4.5'te verilmiştir.

$$\lambda_j(k) = |S_j(k)|^{-1/2} \exp \left\{ -\frac{1}{2} z_j^T(k) S_j^{-1}(k) z_j(k) \right\} \quad (4.5)$$

Eşitlik 4.5'te yer alan z inovasyonu, S ise kovaryans matrisini ifade etmektedir.

4.2.3. Model olasılığının güncelleştirilmesi

Model olasılığı güncelleme işlemi Eşitlik 4.6 ile yapılır.

$$\mu_{ij}(k) = \frac{1}{c} \Lambda_j(k) \sum_{i=1}^r p_{ij} \mu_i(k-1) = \frac{1}{c} \Lambda_j(k) \bar{c}_j \quad (4.6)$$

Eşitlik 4.6'da yer alan normalizasyon faktörü Eşitlik 4.7'deki gibi yazılabilir.

$$c = \sum_{j=1}^r \Lambda_j(k) \bar{c}_j \quad (4.7)$$

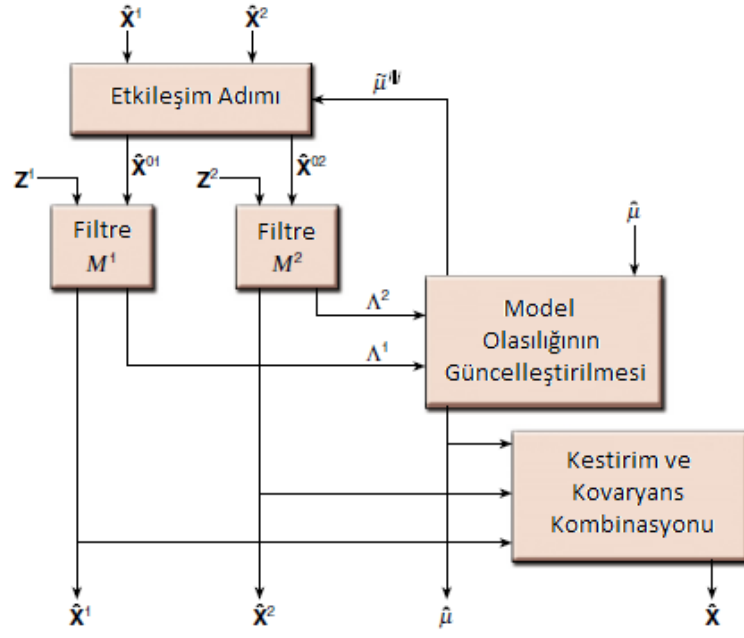
4.2.4. Kestirim ve kovaryans kombinasyonu

Güncelleştirme işleminden sonra elde edilen kestirim ve kovaryans matrisi sırasıyla Eşitlik 4.8 ve Eşitlik 4.9'da verilmiştir.

$$\hat{x}(k|k) = \sum_{j=1}^r \hat{x}^j(k|k) \cdot \mu_j(k) \quad (4.8)$$

$$P(k|k) = \sum_{j=1}^r \mu_j(k) \cdot \left\{ P^j(k|k) + [\hat{x}^j(k|k) - \hat{x}(k|k)] [\hat{x}^j(k|k) - \hat{x}(k|k)]^T \right\} \quad (4.9)$$

İki filtreli bir EÇM'ye ait blok diyagram Şekil 4.4'te verilmiştir.



Şekil 4.4 EÇM Blok Diyagramı [29]

5. AĞIRLIKLANDIRILMIŞ İSTATİSTİKSEL MODEL SEÇİMİ (AİMS)

Hareketli hedeflerin takibinin gerçek zamanlı yapılabilmesi için hedef takibinde sıklıkla tercih edilen ÇMPF algoritmasında kullanılan parçacık sayısının ve/veya parçacık filtresinde her bir hareket modeli için gerçekleştirilen tahmin işlemlerinin optimum değerde tutulması gerekmektedir. Bu amaç ile parçacık filtresine entegre edilmiş optimizasyon algoritmaları kullanılabilir. Bu çalışmada çoklu model parçacık filtrelerinde model tahmini işleminin en kısa sürede, en az işlem ile yapılması hedeflenmiş ve yeni bir yaklaşım olan ağırlıklandırılmış istatistiksel model seçimi (AİMS) algoritması tarafımızdan sunulmuştur. AİMS algoritması hedefe ait hareket modellerinden en uygununu seçerek performans ve hata miktarı arasındaki dengeyi kurmaya çalışır.

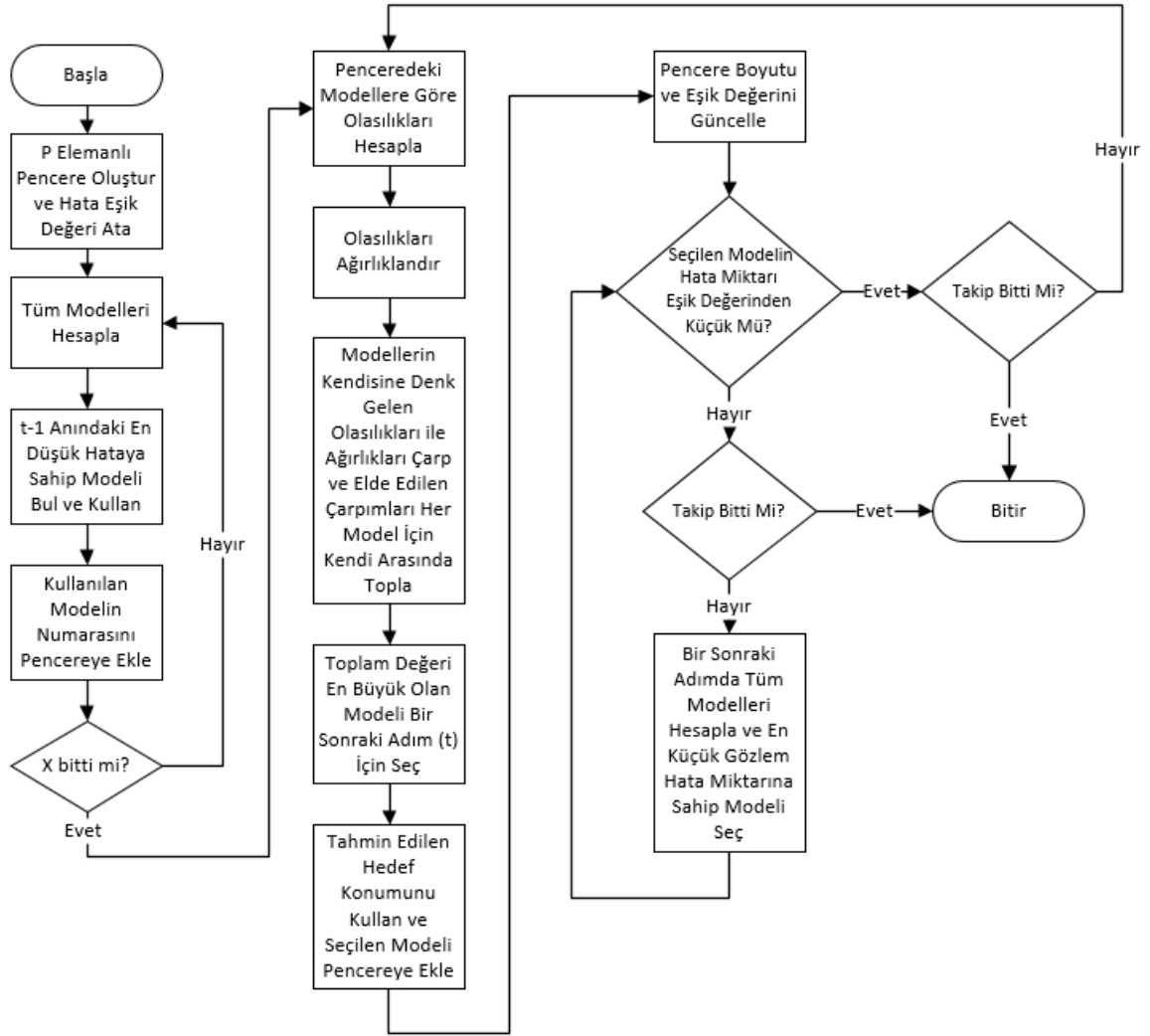
Bir ÇMPF'de X adet model ve N adet parçacık olduğu düşünülürse, toplam yapılması gereken işlem sayısı $X*N$ adet olacaktır. AİMS ile bu sayı N 'ye düşürülmeye çalışılmakta yani bir zaman adımında toplam N adet işlem yapılmaya çalışılmaktadır. Bu da kullanılan ÇMPF'nin yaklaşık X kat daha hızlı çalışmasını sağlamaktadır. AİMS algoritmasında model tahminlerine ilişkin hata değerleri her bir zaman adımında dinamik olarak ayarlanan bir eşik değeri ile kontrol altında tutulmaya çalışılmaktadır. Aksi takdirde AİMS algoritması ÇMPF'de hatalı model seçimine neden olmakta ve hedef takibinde hata miktarı artmaktadır.

Bu algoritmada izlenen adımlar sırasıyla aşağıda verilmiştir:

1. AİMS için en az P elemanlı pencere oluşturulur ve hata eşik değeri atanır.
2. Parçacık filtresini model sayısı kadar (X kere) çalıştırılır.
 - a. Bu süre zarfı içerisinde bütün modeller her bir zaman adımı içerisinde hesaplanır.
 - b. $t-1$ anındaki gözlem değerine göre en az hataya sahip model t anında kullanılır. Bu adımda hiçbir cismin hareketini bir adımda tamamlayamayacağı varsayılır ve bu nedenle $t-1$ anındaki hata miktarı t anındaki tahmin için kullanılabilir.

- c. Her zaman adımı için tercih edilen modelin numarası 1nci maddede oluşturulan pencereye depolanır.
3. İlk X adet zaman adımı tamamlandığında $X+1$ 'nci adım için AİMS algoritması kullanılır.
 - a. Pencere içerisindeki modellerin kullanılma sayılarına göre olasılıkları belirlenir.
 - b. Pencere içerisindeki en son seçilen modelin ağırlığı en büyük, ilk seçilen modelin ağırlığı en küçük olacak şekilde ardışık ağırlıklar atanır.
 - c. Modellerin kendisine denk gelen ağırlıklar ile model olasılıkları çarpılır ve bu çarpım sonuçları her model için toplanır.
 - d. Toplam değeri en büyük olan model bir sonraki adım için seçilir ve sadece bu modelin hesaplamaları yapılır.
4. Tahmin edilen hedef konumu kullanılır ve seçilen model pencereye kaydedilir.
5. Pencere boyutu ve eşik değeri güncellenir.
6. $t-1$ anındaki gerçek gözlem değeri ile $t-1$ anındaki tahmin edilen gözlem değeri arasındaki hata miktarı hesaplanır ve eşik değeri ile karşılaştırılır.
7. Üçüncü adımda yapılan karşılaştırma sonucunda, hesaplanan hata miktarı eşik değerinden küçük ise üçüncü adıma geçilerek işleme devam edilir. Eğer büyükse bir sonraki adımda tüm modeller hesaplanır.

AİMS algoritmasına ait akış diyagramı Şekil 5.1'de verilmiştir.



Şekil 5.1 AİMS Algoritması Akış Şeması

AİMS algoritmasında hata eşik değeri her zaman en küçük hata miktarına eşitlenmeye çalışılmaktadır. Eğer mevcut hata eşik değeri o adım için hesaplanan gözlem hata miktarından büyük ise, o adımdaki gözlem hata miktarı bir sonraki adım için hata eşik değeri olarak kullanılır. Mevcut hata eşik değeri o adım için hesaplanan hata miktarından küçük ise, bir sonraki adımdaki karşılaştırmalar mevcut hata eşik değeri ile devam eder. Fakat bu işlemler sırasında hata eşik değeri her zaman, elde edilen en küçük hata miktarına eşitlenmektedir. Gözlem hata miktarı bir süre sonra hep hata eşik değerinin üzerinde kalmaya başlayacağından, bu durum bir süre sonra AİMS algoritmasının çalışmamasına ve geriye kalan tüm adımlar için tüm dinamik modellerin hesaplanmasına yol

açacaktır. Bunu durumu önlemek için hata eşik değerinin belirli bir değerin altına düşmesini engellemek gerekmektedir ki bunun için bir alt eşik değeri kriteri kullanılmaktadır.

Pencere boyutu ise hesaplanan gözlem hata miktarının hata eşik değeri ile yapılan karşılaştırma sonucuna göre değişmektedir. Eğer hesaplanan gözlem hata miktarı hata eşik değerinden küçük ise pencere boyutu büyütülür, büyük ise küçültülür.

AİMS algoritmasının pencere boyutu ve hata eşik değeri için hazırlanmış, yarı dinamik ve dinamik olmak üzere iki farklı uygulama yöntemi kullanılmıştır. Yarı dinamik yöntemde alt eşik değeri ve pencere boyutu sabit katsayılar yardımı ile hesaplanır. Dinamik yöntemde ise alt eşik değeri ve pencere boyutu hesaplanan gözlem hata değeri ile eşik değerinin karşılaştırılması sonucuna göre artırılıp azaltılmaktadır.

Tam dinamik yöntemde pencere boyutunun hesaplanması işlemi hata eşik değeri ile t anındaki hatanın karşılaştırılması ile yapılmaktadır. Hatanın hata eşik değerinden büyük olduğu durumda pencere boyutu Eşitlik 5.1'e göre küçültülmektedir.

$$\text{pencere boyutu} - (\text{pencere boyutu}/\text{model sayısı}) \quad (5.1)$$

Hatanın hata eşik değerinden küçük olduğu durumda ise pencere boyutu Eşitlik 5.2'ye göre büyütülmektedir.

$$\text{pencere boyutu} + (\text{pencere boyutu}/\text{model sayısı}) \quad (5.2)$$

Eşitlik 5.1 ve Eşitlik 5.2'den elde edilen sonuçlar en yakın tam sayıya yuvarlanmaktadır.

Tam dinamik yöntemde hata eşik değerinin hesaplanması işlemi $t-1$ anındaki hata miktarı ile t anındaki hata miktarının karşılaştırılması ile yapılmaktadır. Eğer $t-1$ anındaki hata miktarı t anındaki hata miktarından büyük ise hata eşik değeri t anındaki hata miktarına eşitlenmektedir. Tam tersi durumda ise aynı hata eşik değeri kullanılmaya devam edilmektedir.

Yarı dinamik yöntemde ise pencere boyutunun ve hata eşik değerinin hesaplanması sırasında bazı sabit parametreler kullanılmaktadır. Pencere boyutu hesaplamalarında pencere boyutu önceden belirlenen sabit bir katsayı ile çarpılarak büyütülmekte veya ilgili katsayı ile bölünerek küçültülmektedir. Hata eşik değeri de aynı şekilde hesaplanmaktadır.

AİMS algoritmasının çalışmasına ilişkin bir örnek aşağıda verilmiştir.

$$\begin{array}{cccc|c|c} M_1 & M_2 & M_3 & M_4 & \dots & M_P \\ W_1 & W_2 & W_3 & W_4 & & W_P \end{array}$$

W_P model ağırlıklarını ($W_P > \dots > W_4 > W_3 > W_2 > W_1$ olacak şekilde), M_P ise model türünü ifade eder.

Çizelge 5.1'de örnek bir pencere verilmiştir.

Çizelge 5.1 Model Ağırlık Çizelgesi

Model Türü	M_1	M_2	M_1	M_3	M_4	M_3	M_2	M_3	M_4	M_1	?
Model Ağırlığı	1	2	3	4	5	6	7	8	9	10	-

Yukarıdaki örneğe bakıldığında 10 elemanlı bir pencere için elde edilen olasılıklar aşağıda verilmiştir:

- Model 1 (M_1): Pencere içerisinde 3 adet bulunduğu için olasılığı 0,3.
- Model 2 (M_2): Pencere içerisinde 2 adet bulunduğu için olasılığı 0,2.
- Model 3 (M_3): Pencere içerisinde 3 adet bulunduğu için olasılığı 0,3.
- Model 4 (M_4): Pencere içerisinde 2 adet bulunduğu için olasılığı 0,2.

Yukarıda elde edilen olasılık değerleri tek başlarına bir şey ifade etmeyeceklerdir. Çünkü örnekte görüldüğü gibi modellere ait olasılık değerleri bazen birbirine eşit sonuçlar vermektedir. Birbirine eşit sonuçlar vermediği durumlarda ise modellerden birisinin baskın hale gelmesi sonraki işlem adımlarında da baskın modelin seçilmesi nedeniyle hatalı model seçimine neden olacaktır. Bu nedenle elde edilen her bir olasılık model ağırlıkları ile çarpılmalıdır. Elde edilen olasılık

değerleri yerlerine yazılıp model ağırlığı ile çarpılırsa Çizelge 5.2'deki sonuçlar elde edilir.

Çizelge 5.2 Model Ağırlık Çizelgesi

Model Türü	M ₁	M ₂	M ₁	M ₃	M ₄	M ₃	M ₂	M ₃	M ₄	M ₁	?
Model Ağırlığı	1	2	3	4	5	6	7	8	9	10	-
Model Olasılığı	0,3	0,2	0,3	0,3	0,2	0,3	0,2	0,3	0,2	0,3	-
Ağırlıklandırılmış Model Katsayısı	0,3	0,4	0,9	1,2	1,0	1,8	1,4	2,4	1,8	3	-

Yukarıdaki pencerede görüldüğü gibi olasılık değerleri model ağırlıkları ile çarpılırsa ağırlıklandırılmış model olasılıkları elde edilir. Elde edilen bu değerler her bir model için kendi arasında toplanır. Bu toplamlar Çizelge 5.3'te verilmiştir.

Çizelge 5.3 Toplam Katsayı Değeri

Model Numarası	Toplam Katsayı
Model 1 (M ₁)	4,2
Model 2 (M ₂)	1,8
Model 3 (M ₃)	5,4
Model 4 (M ₄)	2,8

Yukarıdaki çizelge incelendiğinde bir sonraki zaman adımında olması gereken model 3 numaralı modeldir. Çünkü modellerin toplam katsayı değerlerine bakıldığında en yüksek değerli model 3 numaralı modeldir. Bu bilgiye göre bir sonraki adımda tüm modellerin hesaplanması yerine sadece 3 numaralı modelin hesaplanması yeterli olacaktır. Sistemde toplam 4 adet model olduğu düşünüldüğünde AİMS algoritmasının kullanıldığı zaman adımlarında zamandan yaklaşık 4 kat tasarruf edilmektedir.

6. UYGULAMA

Tezin uygulama aşamasında farklı senaryolar altında kullanılan ÇMPF'lerin; hedef hareketinin, gürültü miktarı artışının ve yeni bir yöntem olan AİMS'nin kullanılmasının hedef takibinin doğruluğu ve algoritma hızı üzerindeki etkileri araştırılmıştır. Bu kapsamda üç farklı senaryo oluşturulmuş ve bu senaryolarda üç farklı ÇMPF kullanılarak iki boyutlu düzlemde hedef takibi yapılmıştır. Dört numaralı başlık altında sunulan üç farklı ÇMPF'nin ikinci ve üçüncü algoritmasında, bu çalışmada önerilen AİMS yaklaşımı bir model seçim yöntemi olarak kullanılmıştır. Algoritma 1'in yapısında modellere ilişkin bir tahmin adımı bulunmamaktadır. Dolayısıyla Algoritma 1 yapısına AİMS yaklaşımı entegre edilememektedir. Yapılan çalışmalarda ÇMPF'ler birbiri ile karşılaştırılmış ve AİMS'nin bu ÇMPF'ler üzerindeki etkisi incelenmiştir.

Senaryolarda hedef için birbirinden farklı dinamik modeller kullanılmıştır. Bu şekilde kullanılan algoritmaların hedef takibi üzerindeki başarısı farklı modeller üzerinde denenmiştir.

Oluşturulan ÇMPF'lerin gürültü artışına vereceği tepkinin gözlemlenebilmesi için gözlem değerlerine ait gürültü varyansı sırası ile 0,01, 0,1, 0,15 ve 0,2 olarak seçilmiştir.

AİMS'e ait pencere boyutu ilk iki adım için 10, hata eşik değeri ise ilk dört (model sayısı) adım için 0,07 seçilmiştir. Belirtilen adım sayılarından sonra sistemde yer alan pencere boyutu ve hata eşik değeri her bir adım için hesaplanarak dinamik hale gelmektedir.

6.1. Kullanılan Materyal

Benzetimlerin gerçekleştirilmesi için aşağıda yer alan bilgisayar kullanılmıştır.

- Intel Core i7 İşlemci (M640 2.80 GHz)
- 6,00 GB RAM
- ATI Radeon HD 5650 Ekran Kartı

- 600 GB Sabit Disk
- Windows 7 (64 Bit) İşletim Sistemi

Benzetimlerin gerçekleştirilmesi için ise MATLAB R2012a (64 Bit) yazılımı kullanılmıştır.

6.2. Senaryo Parametreleri

Senaryolarda uygulanmak üzere genel olarak aşağıdaki parametreler kullanılmıştır.

- Örnekleme aralığı: 1 saniye;
- Gözlem gürültüsü varyansı: $\sigma^2 = 0,01, 0,1, 0,15$ ve $0,2$;
- Model seçimi için hata eşik değeri: $0,07$ (Başlangıç için);
- AİMS pencere genişliği: 10 (Başlangıç için);
- Kullanılan modeller: Sabit hız, sabit ivme ve dönüş oranı bilinen koordineli dönüş modeli (sol ve sağ olmak üzere iki adet);
- Parçacık sayıları: $1000, 2000, 3000, 4000, 5000$ ve 7000 .

Hedef ve gözlemciye ilişkin değişkenler her senaryoda farklı olduğu için bu parametreler ilgili senaryolar altında verilecektir.

Hedef takibinin değerlendirilmesi için verilen hata değerleri, gerçekleştirilen benzetimler çok sayıda çalıştırılarak ve her çalışmada elde edilen hata değerlerinin (gerçek değerler ile tahmin değerleri arasındaki farkın) ortalaması alınarak elde edilmiştir. Çünkü benzetim her çalıştırıldığında sistem rastgele gürültüye maruz bırakılmakta ve bu da elde edilen sonuçları etkilemektedir. Benzetim sayısının artırılması durumunda elde edilen sonuçların ortalamasının alınması ile daha doğru sonuçlar elde edilmektedir. Benzetim sayısı literatür incelenerek ve daha önceki çalışmalar göz önünde bulundurularak 100 seçilmiştir [4; 16; 30; 31; 32].

6.3. Karşılaştırma Kriterleri

Oluşturulan senaryolarda her bir zaman adımında elde edilen gerçek ve tahmin değerleri arasındaki hata miktarının bulunması ortalama karesel hata miktarı (RMSE) ile yapılmaktadır. Bu yöntem ile negatif veya pozitif çıkan hata miktarları birbirini etkilememektedir. Yani RMSE ile hata miktarı sayının işaretinden bağımsız hale gelmektedir. RMSE değerleri Eşitlik 6.1'de yer alan denklem kullanılarak yapılmaktadır.

$$RMSE = \sqrt{\frac{1}{K} \sum_{i=1}^K ((dx^{(i)} - dx^{gerçek})^2 + (dy^{(i)} - dy^{gerçek})^2)} \quad (6.1)$$

Birden fazla benzetim gerçekleştirildiğinde bu benzetimlerden elde edilen hataların ortalamasının alınması gerekmektedir. Bu nedenle Eşitlik 6.2 kullanılır [4; 7; 33; 34; 35].

$$RMSE = \sqrt{\frac{1}{L} \sum_{k=1}^L \frac{1}{K} \sum_{i=1}^K ((dx^{(i)} - dx^{gerçek})^2 + (dy^{(i)} - dy^{gerçek})^2)} \quad (6.2)$$

Eşitlik 6.1 ve Eşitlik 6.2'de yer alan K toplam adım sayısını ifade ederken L ise gerçekleştirilen benzetim sayısını ifade etmektedir. dx ve dy sırası ile x ve y koordinatlarındaki tahmin değerini, $dx^{gerçek}$ ve $dy^{gerçek}$ ise gerçek durumdaki x ve y koordinatlarını ifade etmektedir.

Her bir benzetimin işlem süresi bakımından karşılaştırma işlemi ise MATLAB'ta yer alan zaman işaretleyicileri yardımıyla yapılmıştır.

6.4. Benzetimlerde Kullanılan Senaryolar

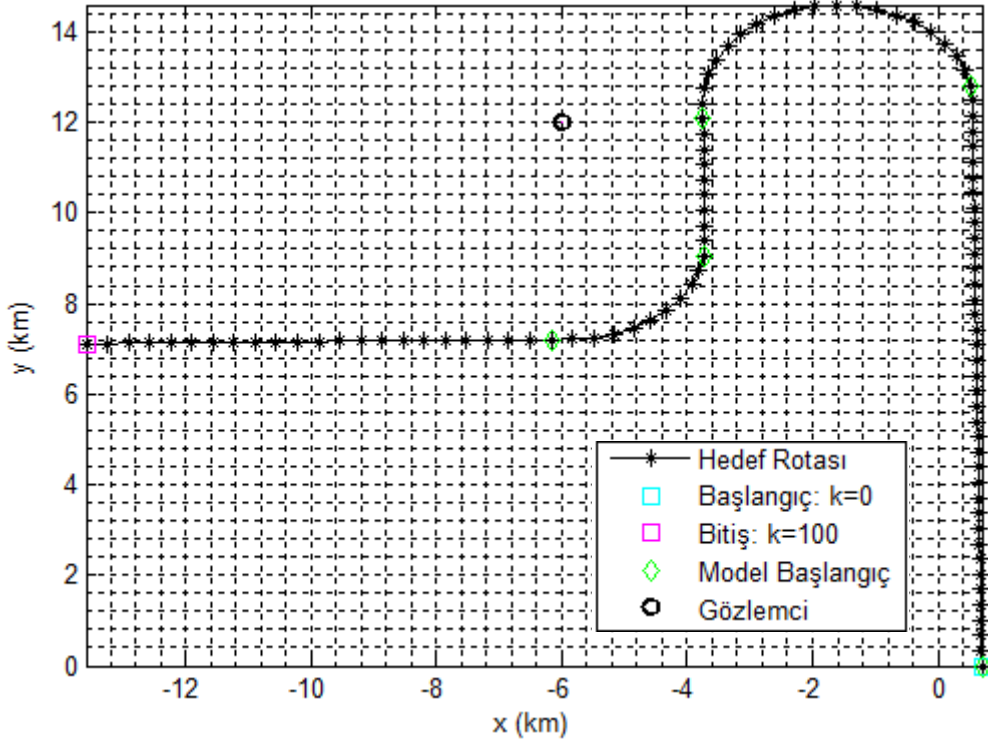
6.4.1. Senaryo 1

Senaryo 1'de iki boyutlu düzlemde sabit hız ve dönüş oranı bilinen koordineli dönüş modelleri kullanılarak hedefe ait basit bir rota oluşturulmuştur. Bu senaryoya göre hedef aşağıda yer alan hareketleri sırası ile gerçekleştirmektedir:

- 38 adım boyunca sabit hız,

- 20 adım boyunca sola koordineli dönüş,
- 9 adım boyunca sabit hız,
- 10 adım boyunca sağa koordineli dönüş,
- 23 adım boyunca sabit hız hareketini gerçekleştirmiştir.

Senaryo 1'e ait hedef rota bilgisi Şekil 6.1'de verilmiştir.



Şekil 6.1 Senaryo 1 Hedef Rota Bilgisi

Senaryo ile ilgili parametreler aşağıda verilmiştir.

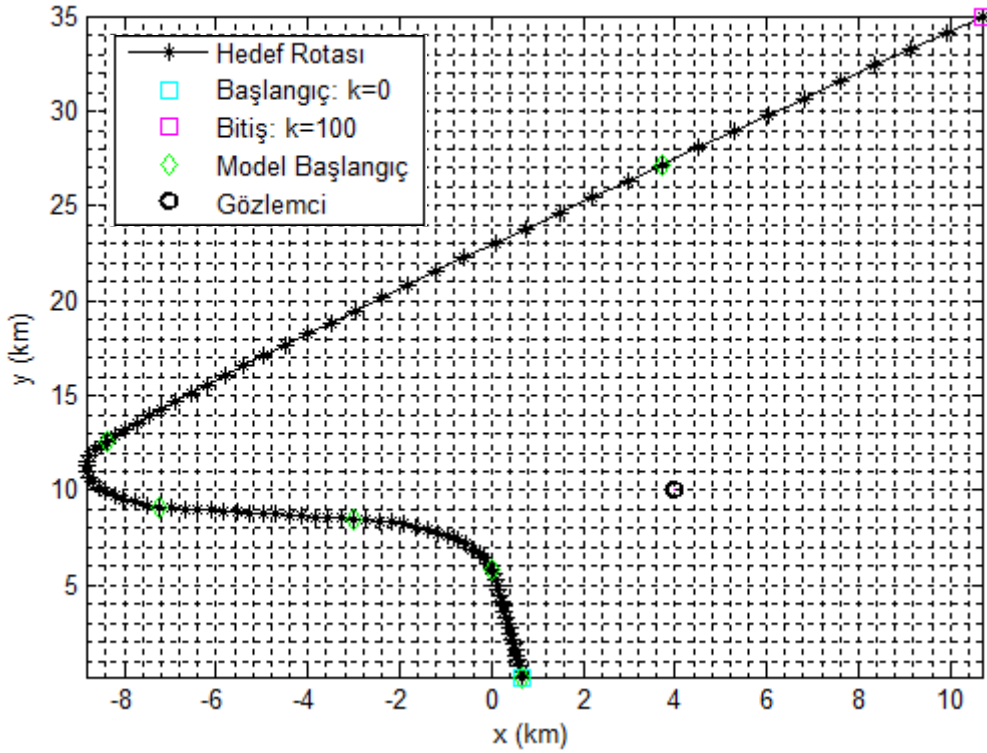
- Gözlemci koordinatı: (-6, 12);
- Hedef manevra alanı: 196 km^2 ;
- Toplam adım sayısı: 100;
- Kullanılan modeller: Sabit hız ve dönüş oranı bilinen koordineli dönüş modeli.

6.4.2. Senaryo 2

Senaryo 2'de iki boyutlu düzlemde sabit hız, sabit ivme ve dönüş oranı bilinen koordineli dönüş modelleri kullanılarak hedefe ait basit bir rota oluşturulmuştur. Bu senaryoya göre hedef aşağıda yer alan hareketleri sırası ile gerçekleştirmektedir:

- 20 adım boyunca sabit hız,
- 15 adım boyunca sola koordineli dönüş,
- 15 adım boyunca sabit hız,
- 15 adım boyunca sağa koordineli dönüş,
- 25 adım boyunca sabit ivme,
- 10 adım boyunca sabit hız hareketini gerçekleştirmiştir.

Senaryo 2'ye ait hedef rota bilgisi Şekil 6.2'de verilmiştir.



Şekil 6.2 Senaryo 2 Hedef Rota Bilgisi

Senaryo ile ilgili parametreler aşağıda verilmiştir.

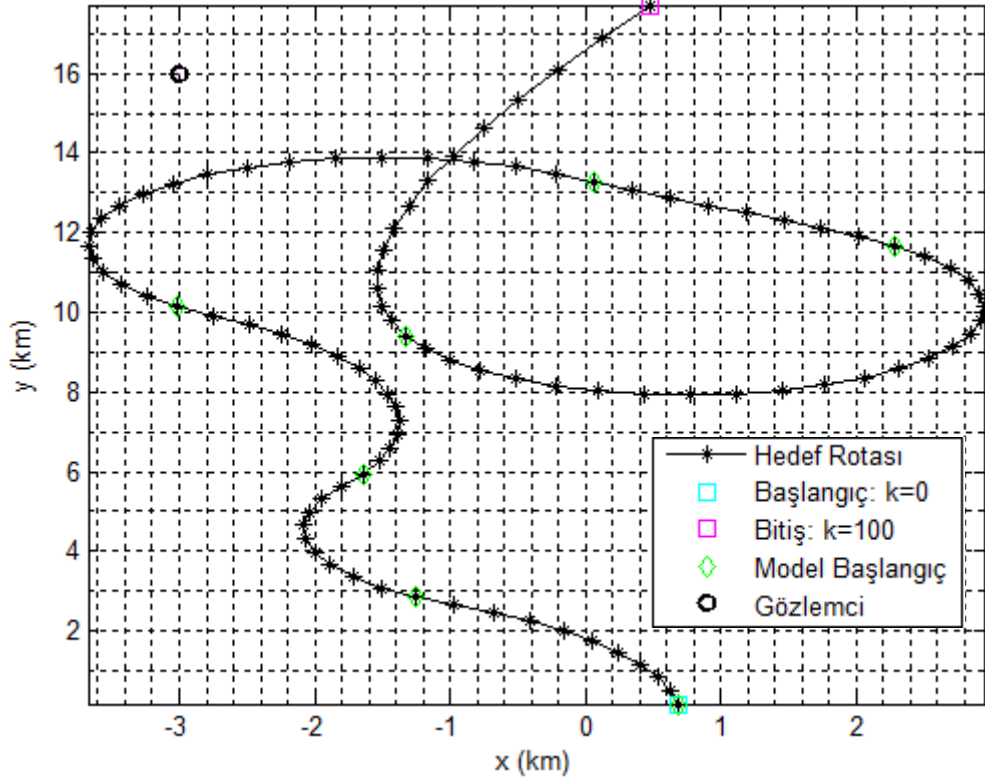
- Gözlemci koordinatı: (4, 10);
- Hedef manevra alanı: 665 km²;
- Toplam adım sayısı: 100;
- Kullanılan modeller: Sabit hız, sabit ivme ve dönüş oranı bilinen koordineli dönüş modeli (sol ve sağ olmak üzere iki adet).

6.4.3. Senaryo 3

Senaryo 3'te iki boyutlu düzlemde sabit hız, sabit ivme ve dönüş oranı bilinen koordineli dönüş modelleri kullanılarak hedefe ait basit bir rota oluşturulmuştur. Bu senaryoya göre hedef aşağıda yer alan hareketleri sırası ile gerçekleştirmektedir:

- 10 adım boyunca sola koordineli dönüş,
- 10 adım boyunca sağa koordineli dönüş,
- 14 adım boyunca sola koordineli dönüş,
- 20 adım boyunca sağa koordineli dönüş,
- 8 adım boyunca sabit hız,
- 23 adım boyunca sağa koordineli dönüş,
- 15 adım boyunca sabit ivme hareketini gerçekleştirmiştir.

Senaryo 3'e ait hedef rota bilgisi Şekil 6.3'te verilmiştir.



Şekil 6.3 Senaryo 3 Hedef Rota Bilgisi

Senaryo ile ilgili parametreler aşağıda verilmiştir.

- Gözlemci koordinatları: (-3, 16);
- Hedef manevra alanı: 116 km²;
- Toplam adım sayısı: 100;
- Kullanılan modeller: Sabit hız, sabit ivme ve dönüş oranı bilinen koordineli dönüş modeli (sol ve sağ olmak üzere iki adet).

6.5. Benzetim Sonuçları

Tüm senaryolar için gözlem gürültüsü varyansı 0,01, 0,1, 0,15 ve 0,2 seçilerek, dört farklı benzetim gerçekleştirilmiştir. Bu şekilde ÇMPF'nin gürültüye karşı olan dayanıklılığı test edilmiştir. 1000, 2000, 3000,4000, 5000 ve 7000 olmak üzere üç farklı parçacık sayısı seçilerek parçacık sayılarının sisteme olan etkisi araştırılmıştır.

Birbirinden farklı 3 adet senaryo, 6 farklı parçacık sayısı ve 4 farklı gürültü seviyesi için yapılan benzetimler; 3 adet ÇMPF ve 2 adet ÇMPF+AİMS algoritmaları için gerçekleştirilmiş olup, sonuçlar bu başlık altında verilmiştir.

6.5.1. Yarı dinamik yaklaşıma ait sonuçlar

Bu kısımda kullanılan AİMS algoritmasındaki pencere boyutu ve hata alt eşik değeri için yarı dinamik yöntem kullanılmıştır. Yarı dinamik yaklaşımda alt eşik değeri ve pencere boyutu sabit katsayılar yardımı ile hesaplanır.

Çizelge 6.1, Çizelge 6.2 ve Çizelge 6.3'te verilen değerler yarı dinamik yaklaşım için gerçekleştirilen benzetim sonuçları olup, benzetimler sırasıyla Senaryo 1, Senaryo 2 ve Senaryo 3 için uygulanmıştır.

Benzetimlerde ÇMPF başlığı altında anlatılan algoritmalar kullanılmış olup, kullanılan bu algoritmalarından ikisine AİMS algoritması uygulanmıştır. Parçacık sayısı ve gürültü varyansları yukarıda belirtilen değerlerde kullanılmıştır.

Çizelge 6.1, Çizelge 6.2 ve Çizelge 6.3'te yer alan t saniye cinsinde zamanı, RMSE ortalama karesel hatayı, N ise parçacık sayısını ifade etmektedir.

Çizelge 6.1 Senaryo 1 – Yarı Dinamik Yaklaşım Sonuçları

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 1	$\sigma^2=0,01$	0,1040	1,4255	0,0634	3,1843	0,0501	5,9043	0,0498	8,6347	0,0475	12,9441	0,0468	21,4765
	$\sigma^2=0,1$	0,7492	1,4273	0,5903	3,2552	0,5748	5,9386	0,5053	8,6866	0,5138	12,4655	0,5083	21,6032
	$\sigma^2=0,15$	1,4581	1,3711	1,1416	3,2077	1,1147	5,6699	1,0733	8,7828	1,0425	12,5262	0,9763	21,3828
	$\sigma^2=0,2$	2,8519	1,5060	1,8377	3,4697	1,6194	5,5431	1,6487	8,7207	1,6660	12,1270	1,4604	21,1736
Algoritma 2	$\sigma^2=0,01$	0,1379	0,7089	0,1234	1,2776	0,1084	1,9274	0,1031	2,5101	0,1025	3,1559	0,0988	4,3022
	$\sigma^2=0,1$	0,3413	0,6748	0,2851	1,1851	0,2840	1,8029	0,2901	2,3691	0,2795	3,0215	0,2820	3,2135
	$\sigma^2=0,15$	0,4850	0,6203	0,4594	1,1221	0,4519	1,6380	0,4211	2,1405	0,4293	2,6550	0,4133	3,6654
	$\sigma^2=0,2$	0,6784	0,5705	0,6024	1,0618	0,5957	1,5695	0,5890	2,1628	0,5773	2,6492	0,5808	3,4804
Algoritma 3	$\sigma^2=0,01$	0,1366	0,6994	0,1111	1,3241	0,1036	1,8533	0,1078	2,5416	0,1028	3,0531	0,1020	4,3493
	$\sigma^2=0,1$	0,3251	0,6489	0,3129	1,2558	0,2920	1,7338	0,2821	2,3985	0,2800	2,8468	0,2854	4,1204
	$\sigma^2=0,15$	0,4911	0,6359	0,4695	1,1396	0,4202	1,6750	0,4129	2,3359	0,4128	2,9849	0,4100	3,7910
	$\sigma^2=0,2$	0,6786	0,5934	0,6229	1,1612	0,6026	1,5975	0,6252	2,0788	0,5752	2,6452	0,5845	3,6150
Algoritma 2 + AİMS	$\sigma^2=0,01$	0,1642	0,4709	0,1362	0,7868	0,1318	1,1671	0,1234	1,4896	0,1216	1,9116	0,1224	2,6212
	$\sigma^2=0,1$	0,3302	0,5318	0,3050	0,9531	0,2952	1,3916	0,2815	1,8329	0,2929	2,3745	0,2788	3,2323
	$\sigma^2=0,15$	0,5358	0,5396	0,4391	0,9723	0,4633	1,4340	0,4511	1,8586	0,4252	2,4139	0,4297	3,3114
	$\sigma^2=0,2$	0,7130	0,5277	0,6830	0,9809	0,6389	1,4431	0,5859	1,8649	0,6797	2,4497	0,6262	3,2964
Algoritma 3 + AİMS	$\sigma^2=0,01$	0,1498	0,4396	0,1279	0,7655	0,1291	1,0972	0,1217	1,4289	0,1113	1,7037	0,1149	2,3994
	$\sigma^2=0,1$	0,3231	0,5349	0,3025	0,9297	0,3027	1,3543	0,2813	1,7077	0,2992	2,1921	0,2800	2,9762
	$\sigma^2=0,15$	0,5680	0,5331	0,4523	0,9489	0,4580	1,4410	0,4274	1,7400	0,4381	2,2183	0,4313	3,0374
	$\sigma^2=0,2$	0,7922	0,5254	0,6678	0,9468	0,6336	1,3738	0,6812	1,7761	0,6362	2,2096	0,6231	3,0121

Çizelge 6.2 Senaryo 2 – Yarı Dinamik Yaklaşım Sonuçları

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 1	$\sigma^2=0,01$	0,1355	1,3950	0,0805	3,1968	0,0701	5,8932	0,0621	8,5624	0,0697	12,8984	0,0571	21,8267
	$\sigma^2=0,1$	0,9545	2,1750	0,7078	3,3020	0,7191	8,3502	0,6127	8,7479	0,6188	13,0469	0,5979	22,0919
	$\sigma^2=0,15$	2,0408	1,6016	1,4906	4,0051	1,2522	5,6193	1,1610	8,5822	1,0869	12,5395	1,1280	21,5944
	$\sigma^2=0,2$	3,7174	1,9736	2,8289	3,9204	2,3329	7,1571	1,9317	8,5040	2,0022	12,4866	2,1384	21,2601
Algoritma 2	$\sigma^2=0,01$	0,2577	0,7157	0,2199	1,2758	0,2124	1,9353	0,2054	2,4477	0,2074	3,1650	0,2070	4,1833
	$\sigma^2=0,1$	0,4183	0,6813	0,3466	1,2048	0,3573	1,8455	0,3593	2,2857	0,3440	3,0196	0,3276	3,9661
	$\sigma^2=0,15$	0,5277	0,6351	0,5249	1,1914	0,4988	1,6820	0,5039	2,2044	0,5022	2,7238	0,4884	3,6837
	$\sigma^2=0,2$	0,8845	0,6252	0,7850	1,1535	0,7576	1,5992	0,7125	2,1097	0,6676	2,5732	0,7191	3,5886
Algoritma 3	$\sigma^2=0,01$	0,2436	0,7060	0,2182	1,2755	0,2063	2,0159	0,2146	2,4497	0,2225	2,9747	0,2032	4,1804
	$\sigma^2=0,1$	0,4428	0,6555	0,3394	1,2055	0,3545	1,8980	0,3505	2,2579	0,3554	3,1390	0,3461	3,9511
	$\sigma^2=0,15$	0,6367	0,6494	0,5457	1,1981	0,5105	1,7312	0,5033	2,2545	0,4881	2,8110	0,4974	3,8786
	$\sigma^2=0,2$	0,8059	0,6181	0,7468	1,1565	0,6971	1,6694	0,7003	2,1593	0,7169	2,8609	0,6985	3,7265
Algoritma 2 + AİMS	$\sigma^2=0,01$	0,2405	0,4369	0,2136	0,7542	0,2088	1,0894	0,2119	1,4002	0,1963	1,8593	0,1937	2,4973
	$\sigma^2=0,1$	0,3872	0,5200	0,3723	0,9291	0,3406	1,3331	0,3572	1,7785	0,3453	2,1938	0,3404	3,1960
	$\sigma^2=0,15$	0,6940	0,5324	0,5411	0,9559	0,5077	1,3732	0,5177	1,8126	0,4939	2,2396	0,4961	3,2358
	$\sigma^2=0,2$	0,8136	0,5428	0,7382	0,9671	0,7437	1,3949	0,6986	1,7820	0,7371	2,2531	0,6610	3,3156
Algoritma 3 + AİMS	$\sigma^2=0,01$	0,2161	0,4315	0,2033	0,7429	0,1965	1,0724	0,1977	1,4081	0,1932	1,7114	0,2012	2,2456
	$\sigma^2=0,1$	0,3849	0,5027	0,3805	0,9135	0,3470	1,3129	0,3737	1,7329	0,3410	2,1365	0,3532	2,9372
	$\sigma^2=0,15$	0,5989	0,5263	0,5171	0,9331	0,5340	1,3631	0,4870	1,7636	0,5073	2,1920	0,4743	2,9716
	$\sigma^2=0,2$	0,8312	0,5225	0,8266	0,9406	0,7462	1,3657	0,8201	1,7763	0,7050	2,1862	0,7815	2,9700

Çizelge 6.3 Senaryo 3 – Yarı Dinamik Yaklaşım Sonuçları

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 1	$\sigma^2=0,01$	0,2259	1,3814	0,1310	3,3227	0,1012	5,8019	0,0986	8,8755	0,0946	12,6163	0,0928	21,9441
	$\sigma^2=0,1$	1,2597	1,3586	1,0330	3,3096	0,8736	5,8333	0,8674	8,8771	0,8342	12,7216	0,7214	21,9434
	$\sigma^2=0,15$	2,8748	1,4716	2,0290	3,1826	1,8862	5,6207	1,7415	10,4953	1,6265	12,3564	1,6175	21,1936
	$\sigma^2=0,2$	5,5993	1,7706	2,8860	3,1059	2,7044	5,5141	2,6191	8,4283	2,5273	12,2517	2,3059	21,0791
Algoritma 2	$\sigma^2=0,01$	0,2541	0,6933	0,2316	1,2424	0,2171	1,8217	0,2119	2,3614	0,2115	3,0021	0,2021	4,1009
	$\sigma^2=0,1$	0,5311	0,6374	0,5030	1,1668	0,5082	1,7137	0,4830	2,2631	0,4839	2,8577	0,4838	3,7804
	$\sigma^2=0,15$	0,8901	0,6327	0,7532	1,1579	0,6951	1,6868	0,7208	2,1964	0,7010	2,7723	0,6810	3,8148
	$\sigma^2=0,2$	1,1428	0,5962	1,1134	1,1131	1,0299	1,6176	0,9849	2,0616	0,9910	2,6415	0,9276	3,6122
Algoritma 3	$\sigma^2=0,01$	0,2590	0,6858	0,2197	1,3054	0,2109	1,8078	0,2150	2,5179	0,2042	2,9503	0,1944	4,3019
	$\sigma^2=0,1$	0,5347	0,6362	0,5104	1,2410	0,4977	1,7061	0,4826	2,3652	0,4906	2,8138	0,4764	4,0644
	$\sigma^2=0,15$	0,8610	0,6185	0,7175	1,1308	0,7039	1,6484	0,7103	2,1496	0,6945	2,6806	0,6999	3,6851
	$\sigma^2=0,2$	1,1831	0,5977	1,1695	1,0797	1,1602	1,5964	1,0438	2,0702	0,9872	2,5837	0,9647	3,5766
Algoritma 2 + AİMS	$\sigma^2=0,01$	0,2606	0,5072	0,2171	0,8880	0,2144	1,3012	0,2048	1,6767	0,2011	2,0602	0,2065	2,7775
	$\sigma^2=0,1$	0,5122	0,5709	0,4919	1,0355	0,4907	1,5118	0,4904	1,9530	0,4624	2,4581	0,4824	3,3924
	$\sigma^2=0,15$	0,8212	0,5655	0,7636	1,0220	0,7097	1,4717	0,7144	1,9359	0,6855	2,4130	0,7061	3,3200
	$\sigma^2=0,2$	1,2277	0,5520	1,1248	1,0001	1,0801	1,4628	0,9780	1,8898	0,9769	2,3813	0,9732	3,2890
Algoritma 3 + AİMS	$\sigma^2=0,01$	0,2407	0,5012	0,2050	0,8551	0,2089	1,2442	0,1999	1,6058	0,1963	1,9679	0,1917	2,6645
	$\sigma^2=0,1$	0,5649	0,5624	0,4990	1,0127	0,5032	1,4913	0,4860	1,9145	0,4775	2,3938	0,4755	3,2603
	$\sigma^2=0,15$	0,8812	0,5634	0,7828	0,9942	0,6924	1,4490	0,6946	1,9017	0,6715	2,3112	0,6888	3,2035
	$\sigma^2=0,2$	1,1766	0,5346	1,0293	0,9937	1,1029	1,4064	1,0055	1,8608	0,9440	2,3076	0,9148	3,1695

Çizelge 6.1, Çizelge 6.2 ve Çizelge 6.3'te görüldüğü gibi Algoritma 1'de parçacık sayısı arttıkça işlem süresi uzamakta, gürültü miktarı arttıkça ise hata değeri artmaktadır. İlgili çizelgelerde Algoritma 2 ve Algoritma 3'ün ise Algoritma 1'e göre gürültü altında daha iyi çalıştığı görülmektedir.

AİMS yaklaşımının algoritmalar üzerindeki etkisini incelemek amacıyla bu yaklaşımın entegre edildiği Algoritma 2 ve 3'e ilişkin sonuçlar birbirleri ile karşılaştırılarak Çizelge 6.4, Çizelge 6.5 ve Çizelge 6.6'da verilmektedir. Algoritma 1'e AİMS yaklaşımının uygulanamaması nedeniyle Çizelge 6.4, Çizelge 6.5 ve Çizelge 6.6'daki karşılaştırma çizelgelerine Algoritma 1 eklenmemiştir.

Çizelge 6.4, Çizelge 6.5 ve Çizelge 6.6'da görüldüğü gibi her bir senaryo için;

- Algoritma 2 ve Algoritma 2+AİMS,
- Algoritma 2 ve Algoritma 3,
- Algoritma 3 ve Algoritma 3+ AİMS,
- Algoritma 2+ AİMS ve Algoritma 3+ AİMS,
- Algoritma 2 ve Algoritma 3+ AİMS,
- Algoritma 2+ AİMS ve Algoritma 3.

karşılaştırma sonuçları yer almaktadır. Karşılaştırmalar yüzde bazında olup, pozitif değerler sistemdeki iyileşmeyi, negatif olanlar ise sistemdeki kötüleşmeyi ifade etmektedir.

Tablolarda yer alan Algoritma A-B karşılaştırmaları Eşitlik 6.3'teki ifade ile hesaplanmıştır.

$$((\text{Algoritma A} - \text{Algoritma B})/\text{Algoritma A}) * 100 \quad (6.3)$$

Çizelge 6.4 Senaryo 1 – Yarı Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 2-2+AIMS	$\sigma^2=0,01$	-19,07	33,57	-10,37	38,42	-21,59	39,45	-19,69	40,66	-18,63	39,43	-23,89	39,07
	$\sigma^2=0,1$	3,25	21,19	-6,98	19,58	-3,94	22,81	2,96	22,63	-4,79	21,41	1,13	-0,59
	$\sigma^2=0,15$	-10,47	13,01	4,42	13,35	-2,52	12,45	-7,12	13,17	0,96	9,08	-3,97	9,66
	$\sigma^2=0,2$	-5,10	7,50	-13,38	7,62	-7,25	8,05	0,53	13,77	-17,74	7,53	-7,82	5,29
Algoritma 2-3	$\sigma^2=0,01$	0,94	1,34	9,97	-3,64	4,43	3,84	-4,56	-1,25	-0,29	3,26	-3,24	-1,09
	$\sigma^2=0,1$	4,75	3,84	-9,75	-5,97	-2,82	3,83	2,76	-1,24	-0,18	5,78	-1,21	-28,22
	$\sigma^2=0,15$	-1,26	-2,51	-2,20	-1,56	7,01	-2,26	1,95	-9,13	3,84	-12,43	0,80	-3,43
	$\sigma^2=0,2$	-0,03	-4,01	-3,40	-9,36	-1,16	-1,78	-6,15	3,88	0,36	0,15	-0,64	-3,87
Algoritma 3-3+AIMS	$\sigma^2=0,01$	-9,66	37,15	-15,12	42,19	-24,61	40,80	-12,89	43,78	-8,27	44,20	-12,65	44,83
	$\sigma^2=0,1$	0,62	17,57	3,32	25,97	-3,66	21,89	0,28	28,80	-6,86	23,00	1,89	27,77
	$\sigma^2=0,15$	-15,66	16,17	3,66	16,73	-9,00	13,97	-3,51	25,51	-6,13	25,68	-5,20	19,88
	$\sigma^2=0,2$	-16,74	11,46	-7,21	18,46	-5,14	14,00	-8,96	14,56	-10,61	16,47	-6,60	16,68
Algoritma 2+AIMS -3+AIMS	$\sigma^2=0,01$	8,77	6,65	6,09	2,71	2,05	5,99	1,38	4,07	8,47	10,88	6,13	8,46
	$\sigma^2=0,1$	2,15	-0,58	0,82	2,46	-2,54	2,68	0,07	6,83	-2,15	7,68	-0,43	7,92
	$\sigma^2=0,15$	-6,01	1,20	-3,01	2,41	1,14	-0,49	5,25	6,38	-3,03	8,10	-0,37	8,27
	$\sigma^2=0,2$	-11,11	0,44	2,23	3,48	0,83	4,80	-16,27	4,76	6,40	9,80	0,50	8,62
Algoritma 2-3+AIMS	$\sigma^2=0,01$	-8,63	37,99	-3,65	40,08	-19,10	43,07	-18,04	43,07	-8,59	46,02	-16,30	44,23
	$\sigma^2=0,1$	5,33	20,73	-6,10	21,55	-6,58	24,88	3,03	27,92	-7,05	27,45	0,71	7,38
	$\sigma^2=0,15$	-17,11	14,06	1,55	15,44	-1,35	12,03	-1,50	18,71	-2,05	16,45	-4,36	17,13
	$\sigma^2=0,2$	-16,77	7,91	-10,86	10,83	-6,36	12,47	-15,65	17,88	-10,20	16,59	-7,28	13,46
Algoritma 2+AIMS -3	$\sigma^2=0,01$	16,81	-48,52	18,43	-68,29	21,40	-58,80	12,64	-70,62	15,46	-59,71	16,67	-65,93
	$\sigma^2=0,1$	1,54	-22,02	-2,59	-31,76	1,08	-24,59	-0,21	-30,86	4,40	-19,89	-2,37	-27,48
	$\sigma^2=0,15$	8,34	-17,85	-6,92	-17,21	9,30	-16,81	8,47	-25,68	2,92	-23,65	4,58	-14,48
	$\sigma^2=0,2$	4,82	-12,45	8,80	-18,38	5,68	-10,70	-6,71	-11,47	15,37	-7,98	6,66	-9,67

Çizelge 6.5 Senaryo 2 – Yarı Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 2-2+AİMS	$\sigma^2=0,01$	6,67	38,95	2,86	40,88	1,69	43,71	-3,16	42,80	5,35	41,25	6,43	40,30
	$\sigma^2=0,1$	7,43	23,68	-7,41	22,88	4,67	27,76	0,58	22,19	-0,38	27,35	-3,91	19,42
	$\sigma^2=0,15$	-31,51	16,17	-3,09	19,77	-1,78	18,36	-2,74	17,77	1,65	17,78	-1,58	12,16
	$\sigma^2=0,2$	8,02	13,18	5,96	16,16	1,83	12,78	1,95	15,53	-10,41	12,44	8,08	7,61
Algoritma 2-3	$\sigma^2=0,01$	5,47	1,36	0,77	0,02	2,87	-4,16	-4,48	-0,08	-7,28	6,01	1,84	0,07
	$\sigma^2=0,1$	-5,86	3,79	2,08	-0,06	0,78	-2,84	2,45	1,22	-3,31	-3,95	-5,65	0,38
	$\sigma^2=0,15$	-20,66	-2,25	-3,96	-0,56	-2,35	-2,93	0,12	-2,27	2,81	-3,20	-1,84	-5,29
	$\sigma^2=0,2$	8,89	1,14	4,87	-0,26	7,99	-4,39	1,71	-2,35	-7,38	-11,18	2,86	-3,84
Algoritma 3-3+AİMS	$\sigma^2=0,01$	11,29	38,88	6,83	41,76	4,75	46,80	7,88	42,52	13,17	42,47	0,98	46,28
	$\sigma^2=0,1$	13,08	23,31	-12,11	24,22	2,12	30,83	-6,62	23,25	4,05	31,94	-2,05	25,66
	$\sigma^2=0,15$	5,94	18,96	5,24	22,12	-4,60	21,26	3,24	21,77	-3,93	22,02	4,64	23,38
	$\sigma^2=0,2$	-3,14	15,47	-10,69	18,67	-7,04	18,19	-17,11	17,74	1,66	23,58	-11,88	20,30
Algoritma 2+AİMS - 3+AİMS	$\sigma^2=0,01$	10,15	1,24	4,82	1,50	5,89	1,56	6,70	-0,56	1,58	7,95	-3,87	10,08
	$\sigma^2=0,1$	0,59	3,33	-2,20	1,68	-1,88	1,52	-4,62	2,56	1,25	2,61	-3,76	8,10
	$\sigma^2=0,15$	13,70	1,15	4,44	2,39	-5,18	0,74	5,93	2,70	-2,71	2,13	4,39	8,16
	$\sigma^2=0,2$	-2,16	3,74	-11,98	2,74	-0,34	2,09	-17,39	0,32	4,35	2,97	-18,23	10,42
Algoritma 2-3+AİMS	$\sigma^2=0,01$	16,14	39,71	7,55	41,77	7,49	44,59	3,75	42,47	6,85	45,93	2,80	46,32
	$\sigma^2=0,1$	7,98	26,21	-9,78	24,18	2,88	28,86	-4,01	24,19	0,87	29,25	-7,81	25,94
	$\sigma^2=0,15$	-13,49	17,13	1,49	21,68	-7,06	18,96	3,35	20,00	-1,02	19,52	2,89	19,33
	$\sigma^2=0,2$	6,03	16,43	-5,30	18,46	1,50	14,60	-15,10	15,80	-5,60	15,04	-8,68	17,24
Algoritma 2+AİMS -3	$\sigma^2=0,01$	-1,29	-61,59	-2,15	-69,12	1,20	-85,05	-1,27	-74,95	-13,35	-59,99	-4,90	-67,40
	$\sigma^2=0,1$	-14,36	-26,06	8,84	-29,75	-4,08	-42,37	1,88	-26,96	-2,92	-43,09	-1,67	-23,63
	$\sigma^2=0,15$	8,26	-21,98	-0,85	-25,34	-0,55	-26,07	2,78	-24,38	1,17	-25,51	-0,26	-19,87
	$\sigma^2=0,2$	0,95	-13,87	-1,16	-19,58	6,27	-19,68	-0,24	-21,17	2,74	-26,98	-5,67	-12,39

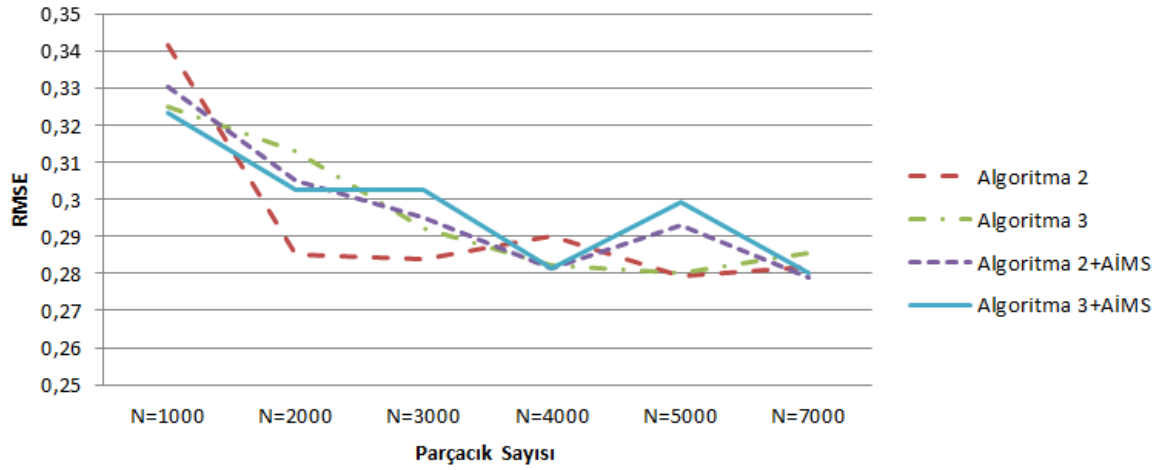
Çizelge 6.6 Senaryo 3 – Yarı Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 2-2+AIMS	$\sigma^2=0,01$	-2,56	26,84	6,26	28,53	1,24	28,57	3,35	29,00	4,92	31,37	-2,18	32,27
	$\sigma^2=0,1$	3,56	10,43	2,21	11,25	3,44	11,78	-1,53	13,70	4,44	13,98	0,29	10,26
	$\sigma^2=0,15$	7,74	10,62	-1,38	11,74	-2,10	12,75	0,89	11,86	2,21	12,96	-3,69	12,97
	$\sigma^2=0,2$	-7,43	7,41	-1,02	10,15	-4,87	9,57	0,70	8,33	1,42	9,85	-4,92	8,95
Algoritma 2-3	$\sigma^2=0,01$	-1,93	1,08	5,14	-5,07	2,86	0,76	-1,46	-6,63	3,45	1,73	3,81	-4,90
	$\sigma^2=0,1$	-0,68	0,19	-1,47	-6,36	2,07	0,44	0,08	-4,51	-1,38	1,54	1,53	-7,51
	$\sigma^2=0,15$	3,27	2,24	4,74	2,34	-1,27	2,28	1,46	2,13	0,93	3,31	-2,78	3,40
	$\sigma^2=0,2$	-3,53	-0,25	-5,04	3,00	-12,65	1,31	-5,98	-0,42	0,38	2,19	-4,00	0,99
Algoritma 3-3+AIMS	$\sigma^2=0,01$	7,07	26,92	6,69	34,50	0,95	31,18	7,02	36,22	3,87	33,30	1,39	38,06
	$\sigma^2=0,1$	-5,65	11,60	2,23	18,40	-1,11	12,59	-0,70	19,06	2,67	14,93	0,19	19,78
	$\sigma^2=0,15$	-2,35	8,91	-9,10	12,08	1,63	12,10	2,21	11,53	3,31	13,78	1,59	13,07
	$\sigma^2=0,2$	0,55	10,56	11,99	7,97	4,94	11,90	3,67	10,11	4,38	10,69	5,17	11,38
Algoritma 2+AIMS - 3+AIMS	$\sigma^2=0,01$	7,64	1,18	5,57	3,70	2,57	4,38	2,39	4,23	2,39	4,48	7,17	4,07
	$\sigma^2=0,1$	-10,29	1,49	-1,44	2,20	-2,55	1,36	0,90	1,97	-3,27	2,62	1,43	3,89
	$\sigma^2=0,15$	-7,31	0,37	-2,51	2,72	2,44	1,54	2,77	1,77	2,04	4,22	2,45	3,51
	$\sigma^2=0,2$	4,16	3,15	8,49	0,64	-2,11	3,86	-2,81	1,53	3,37	3,09	6,00	3,63
Algoritma 2-3+AIMS	$\sigma^2=0,01$	5,27	27,71	11,49	31,17	3,78	31,70	5,66	32,00	7,19	34,45	5,15	35,03
	$\sigma^2=0,1$	-6,36	11,77	0,80	13,21	0,98	12,98	-0,62	15,40	1,32	16,23	1,72	13,76
	$\sigma^2=0,15$	1,00	10,95	-3,93	14,14	0,39	14,10	3,63	13,42	4,21	16,63	-1,15	16,02
	$\sigma^2=0,2$	-2,96	10,33	7,55	10,73	-7,09	13,06	-2,09	9,74	4,74	12,64	1,38	12,26
Algoritma 2+AIMS -3	$\sigma^2=0,01$	0,61	-35,21	-1,20	-47,00	1,63	-38,93	-4,98	-50,17	-1,54	-43,20	5,86	-54,88
	$\sigma^2=0,1$	-4,39	-11,44	-3,76	-19,85	-1,43	-12,85	1,59	-21,11	-6,10	-14,47	1,24	-19,81
	$\sigma^2=0,15$	-4,85	-9,37	6,04	-10,65	0,82	-12,01	0,57	-11,04	-1,31	-11,09	0,88	-11,00
	$\sigma^2=0,2$	3,63	-8,28	-3,97	-7,96	-7,42	-9,13	-6,73	-9,55	-1,05	-8,50	0,87	-8,74

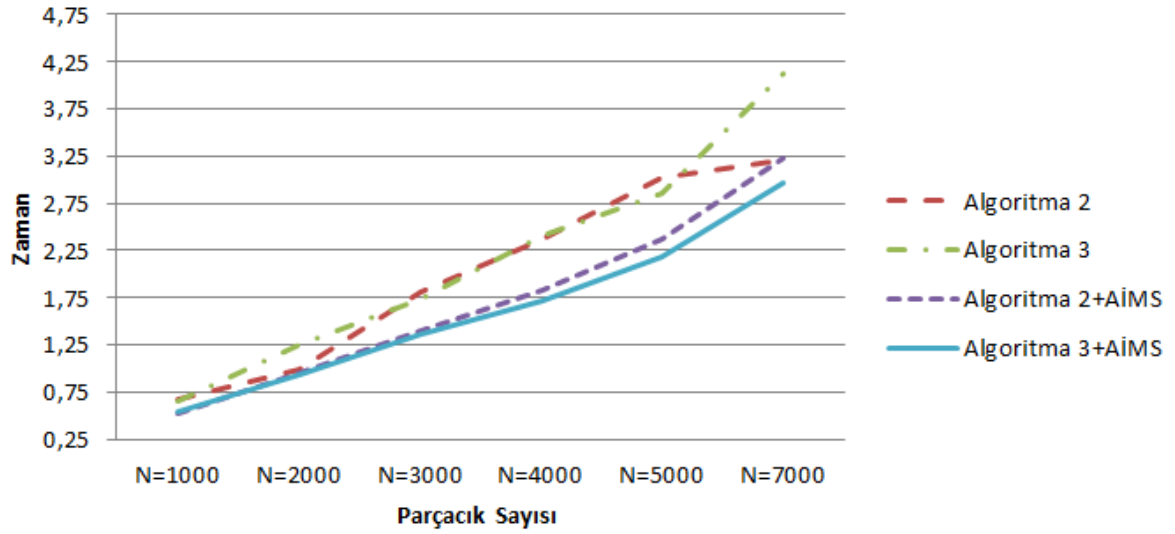
Senaryo 1 için Çizelge 6.4 incelendiğinde;

- Algoritma 2 ve Algoritma 2+AİMS için yapılan işlem süresi karşılaştırmalarda 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 38, 17, 11 ve 8 iyileşme sağlanmıştır. Fakat hata miktarlarında ise sırasıyla yüzde 18, 1, 3 ve 8 artış meydana gelmiştir.
- Algoritma 2 ve Algoritma 3 için yapılan işlem süresi ve hata karşılaştırmalarında (elde edilen sonuçların ortalaması alındığında) her iki algoritmanın birbirine yakın tepkiler gösterdiği görülmektedir.
- Algoritma 3 ve Algoritma 3+AİMS için yapılan işlem süresi karşılaştırmalarında 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 42, 24, 19 ve 15 iyileşme sağlanmıştır. Fakat hata miktarlarında ise sırasıyla yüzde 13, 1, 5 ve 9 artış meydana gelmiştir.
- Algoritma 2+AİMS ve Algoritma 3+AİMS için yapılan karşılaştırmalarda işlem süresi bakımından Algoritma 3'ün Algoritma 2'den ortalama yüzde 5 daha iyi olduğu, hata bakımından ise benzer tepki verdikleri görülmektedir.
- Algoritma 2 ve Algoritma 3+AİMS için yapılan işlem süresi karşılaştırmalarında Algoritma 3+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 42, 21, 15 ve 13 daha iyi olduğu görülmüştür. Fakat hata miktarlarında bakımından incelendiğinde Algoritma 2'nin sırasıyla ortalama yüzde 12, 1, 4 ve 11 daha iyi olduğu görülmüştür.
- Algoritma 2+AİMS ve Algoritma 3 için yapılan işlem süresi karşılaştırmalarında Algoritma 2+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 61, 26, 19 ve 11 daha iyi olduğu görülmüştür. Fakat hata miktarlarında bakımından incelendiğinde Algoritma 3'ün sırasıyla ortalama yüzde 16, 1, 4 ve 5 daha iyi olduğu görülmüştür.

Senaryo 1 için tüm parçacıklara ait 0,1 gürültü varyanslı benzetim sonuçlarına ait hata ve işlem süresi grafikleri sırasıyla Şekil 6.4 ve Şekil 6.5'te verilmiştir.



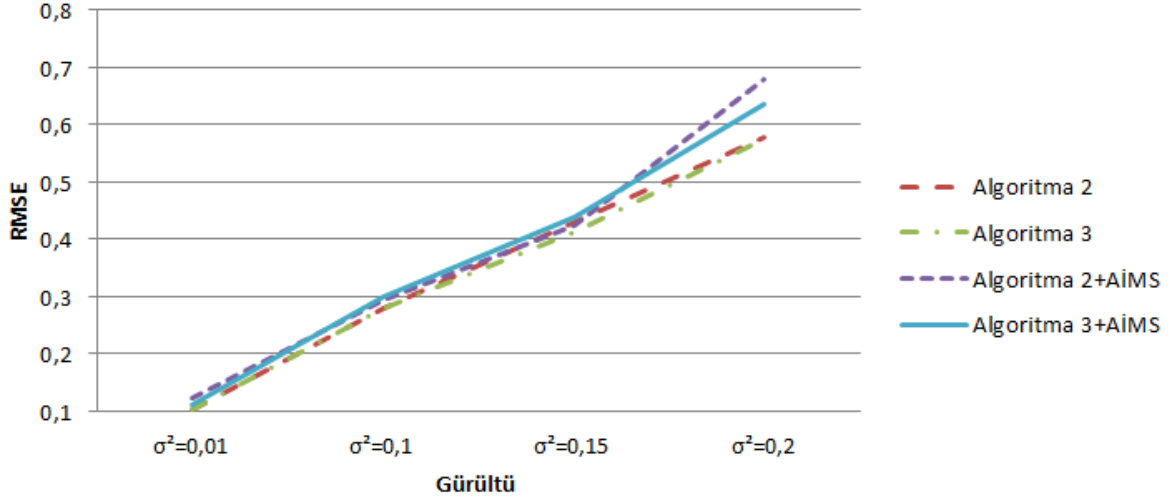
Şekil 6.4 Senaryo 1 Tüm Parçacıklar İçin Hata Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)



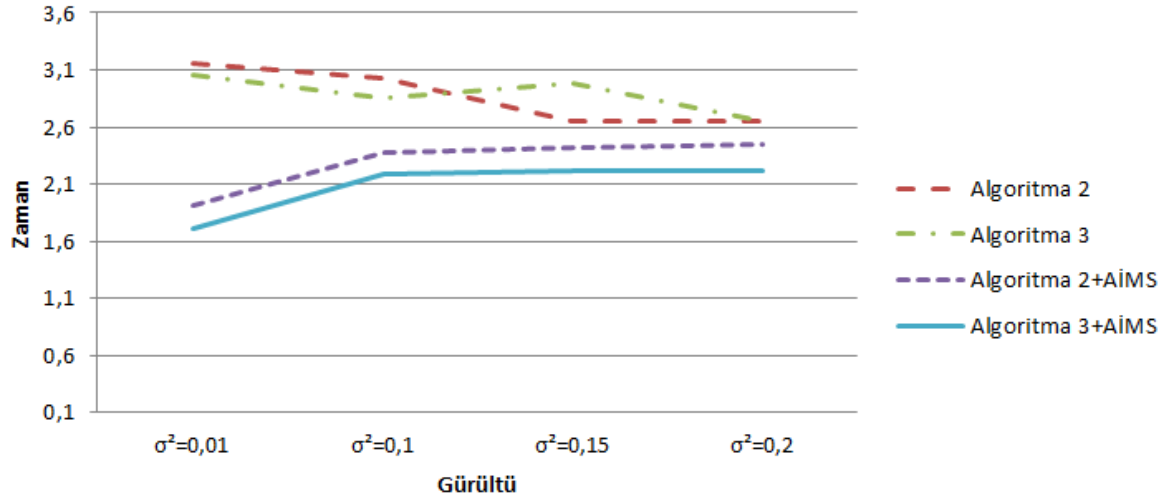
Şekil 6.5 Senaryo 1 Tüm Parçacıklar İçin Zaman Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)

Şekil 6.4 ve Şekil 6.5 incelendiğinde Algoritma 2 ve Algoritma 3'ün işlem süresi bakımından benzer tepkiler verdiği, hata bakımından ise Algoritma 2'nin Algoritma 3'ten daha iyi olduğu görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın Algoritma 2+AİMS için arttığı, işlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği görülmüştür. Algoritma 3+AİMS'in işlem süresi bakımından Algoritma 2+AİMS'ten daha iyi olduğu görülmektedir.

Senaryo 1 için tüm gürültü varyansları ve 5000 parçacık için hata ve işlem süresi grafikleri sırasıyla Şekil 6.6 ve Şekil 6.7’de verilmiştir.



Şekil 6.6 Senaryo 1 Tüm Gürültüler İçin Hata Grafiği (Yarı Dinamik ve N=5000)



Şekil 6.7 Senaryo 1 Tüm Gürültüler İçin Zaman Grafiği (Yarı Dinamik ve N=5000)

Şekil 6.6 ve Şekil 6.7 incelendiğinde Algoritma 2 ve Algoritma 3’ün hem işlem süresi hem de hata miktarı bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın her iki algoritma için çok fazla değişmediği, işlem süresi bakımından ise her iki algoritmanın pozitif

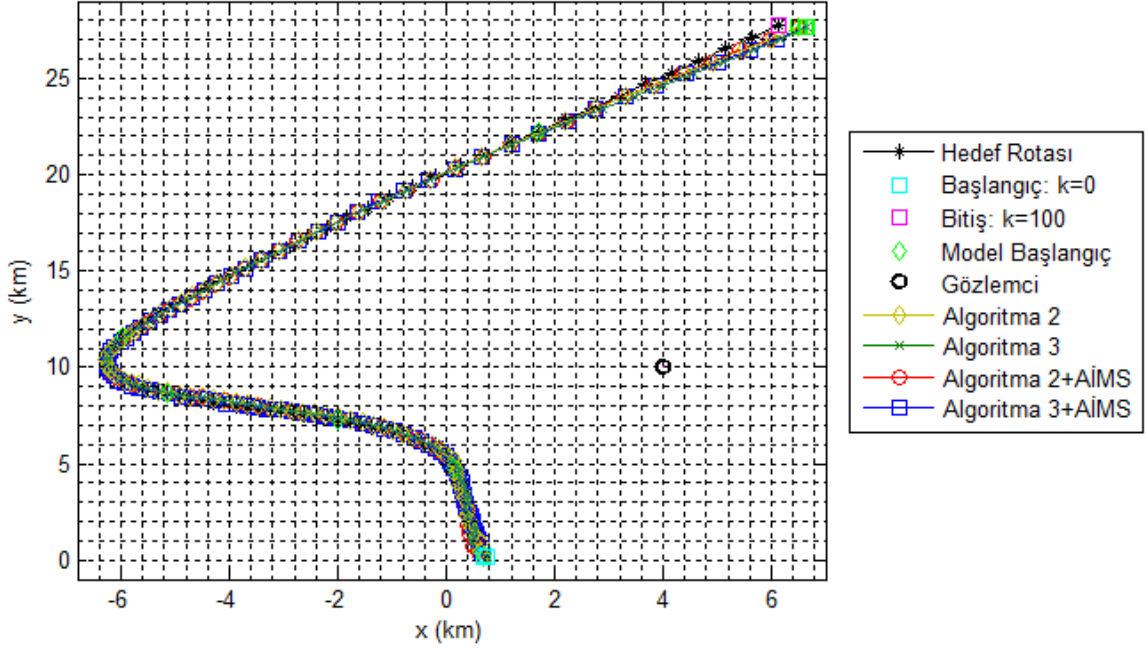
yönde etkilendiği görülmüştür. Algoritma 3+AİMS'in işlem süresi bakımından Algoritma 2+AİMS'ten daha iyi olduğu görülmektedir.

Senaryo 2 için Çizelge 6.5 incelendiğinde;

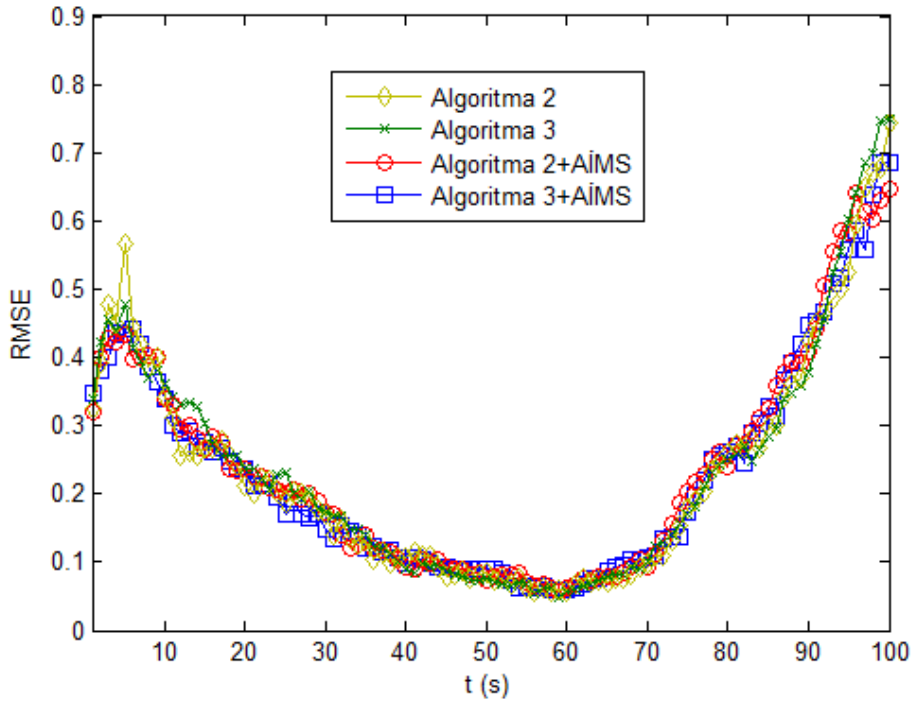
- Algoritma 2 ve Algoritma 2+AİMS için yapılan işlem süresi karşılaştırmalarında 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 41, 23, 17 ve 12 iyileşme sağlanmıştır. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.
- Algoritma 2 ve Algoritma 3 için yapılan işlem süresi ve hata karşılaştırmalarında (elde edilen sonuçların ortalaması alındığında) her iki algoritmanın birbirine yakın tepkiler gösterdiği görülmektedir.
- Algoritma 3 ve Algoritma 3+AİMS için yapılan işlem süresi karşılaştırmalarında 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 42, 24, 19 ve 15 iyileşme sağlanmıştır. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.
- Algoritma 2+AİMS ve Algoritma 3+AİMS için yapılan karşılaştırmalarda işlem süresi bakımından Algoritma 3'ün Algoritma 2'den ortalama yüzde 3 daha iyi olduğu, hata bakımından ise benzer tepki verdikleri görülmektedir.
- Algoritma 2 ve Algoritma 3+AİMS için yapılan işlem süresi karşılaştırmalarında Algoritma 3+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 43, 26, 19 ve 16 daha iyi olduğu görülmüştür. Fakat hata miktarlarında bakımından incelendiğinde Algoritma 2'nin 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 1, 2 ve 4 daha iyi olduğu, 0,01 gürültü seviyesi için ise yaklaşık yüzde 7 daha kötü olduğu görülmüştür.
- Algoritma 2+AİMS ve Algoritma 3 için yapılan işlem süresi karşılaştırmalarında Algoritma 2+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 69, 31, 23 ve 18 daha iyi olduğu

görülmüştür. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.

Senaryo 2 için gerçekleştirilen MATLAB benzetimine ilişkin hedef takip ve hata grafikleri Şekil 6.8 ve 6.9'da sırasıyla verilmiştir. İlgili benzetim gözlem gürültüsü varyansı $\sigma^2=0,1$ ve parçacık sayısı 5000 için gerçekleştirilmiştir.

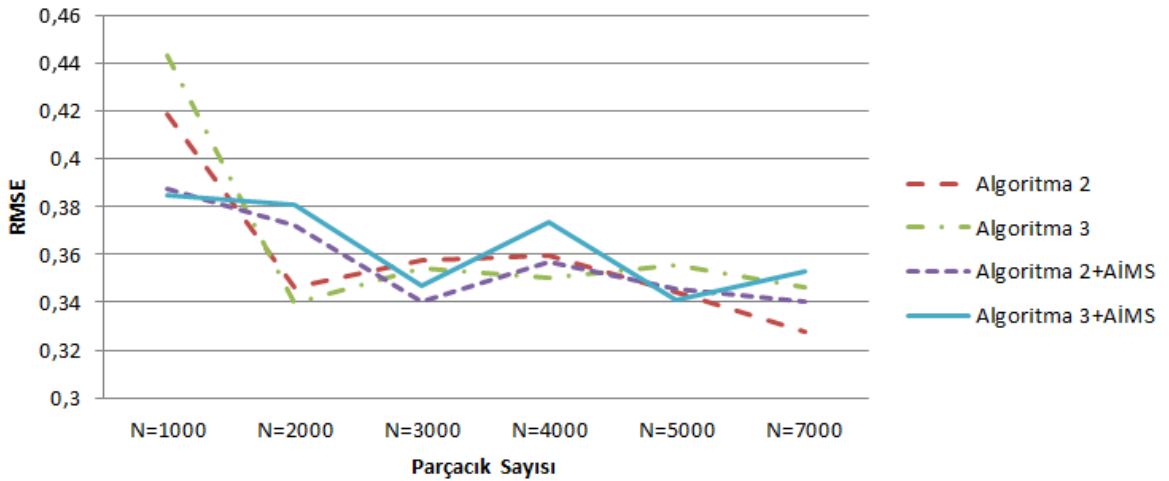


Şekil 6.8 Senaryo 2 Hedef Takibi (Yarı Dinamik, $\sigma^2=0,1$ ve $N=5000$)

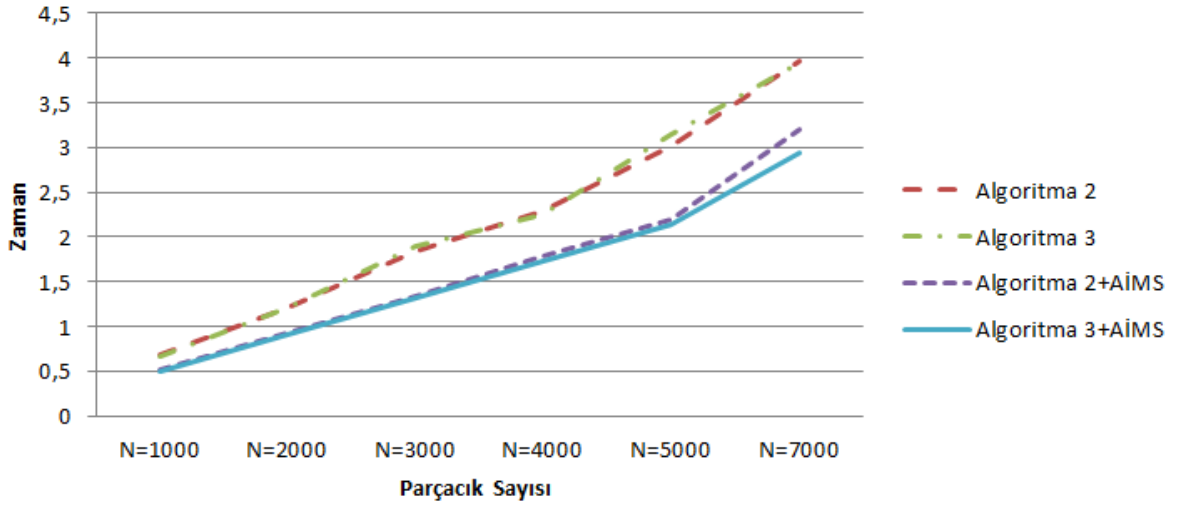


Şekil 6.9 Senaryo 2 RMSE Grafiği (Yarı Dinamik, $\sigma^2=0,1$ ve $N=5000$)

Senaryo 2 için tüm parçacıklara ait 0,1 gürültü varyanslı benzetim sonuçlarına ait hata ve işlem süresi grafikleri sırasıyla Şekil 6.10 ve Şekil 6.11'de verilmiştir.



Şekil 6.10 Senaryo 2 Tüm Parçacıklar İçin Hata Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)

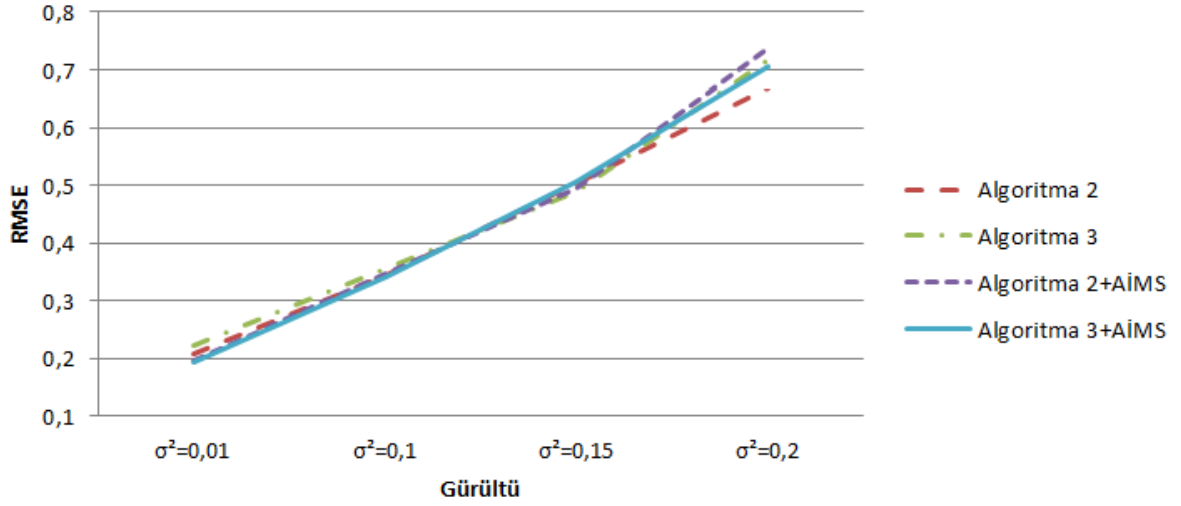


Şekil 6.11 Senaryo 2 Tüm Parçacıklar İçin Zaman Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)

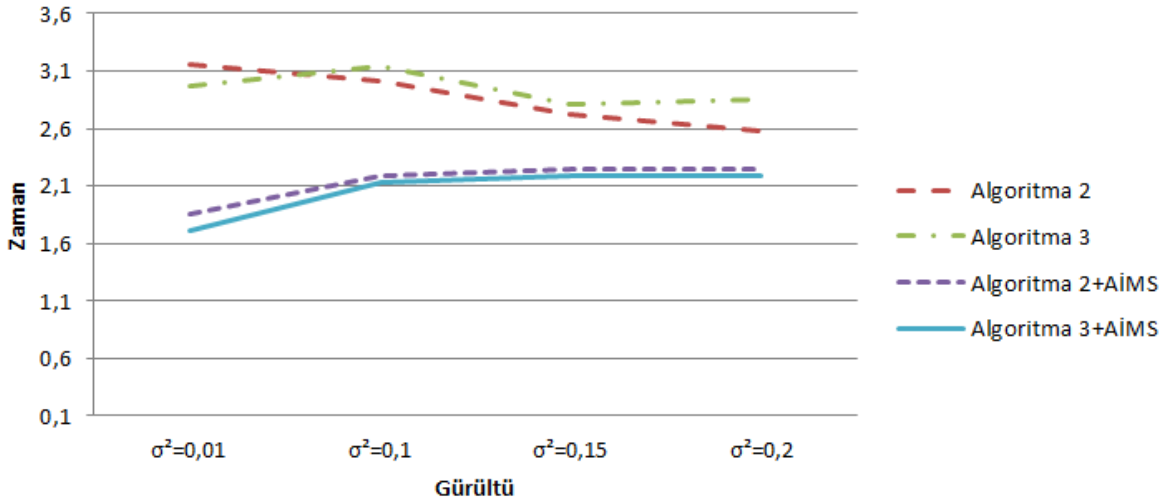
Şekil 6.8 ve Şekil 6.9 yer alan grafikler Şekil 6.10 ve Şekil 6.11 yardımıyla 5000 parçacık için incelendiğinde Algoritma 3+AİMS'in işlem süresi ve RMSE bakımından en iyi sonucu verdiği, Algoritma 3+AİMS'in Algoritma 3'ten daha düşük RMSE değerine sahip olduğu görülmektedir. Algoritma 2 ve Algoritma 2+AİMS ise 5000 parçacık için aynı RMSE değerine sahiptir.

Şekil 6.10 ve Şekil 6.11 incelendiğinde Algoritma 2 ve Algoritma 3'ün işlem süresi ve hata bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın Algoritma 3+AİMS için bazı parçacıklar için arttığı, Algoritma 2+AİMS için ise azaldığı görülmüştür. İşlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği ve Algoritma 3+AİMS'in Algoritma 2+AİMS'ten daha iyi olduğu görülmektedir.

Senaryo 2 için tüm gürültü varyansları ve 5000 parçacık için hata ve zaman grafikleri sırasıyla Şekil 6.12 ve Şekil 6.13'te verilmiştir.



Şekil 6.12 Senaryo 2 Tüm Gürültüler İçin Hata Grafiği (Yarı Dinamik ve N=5000)



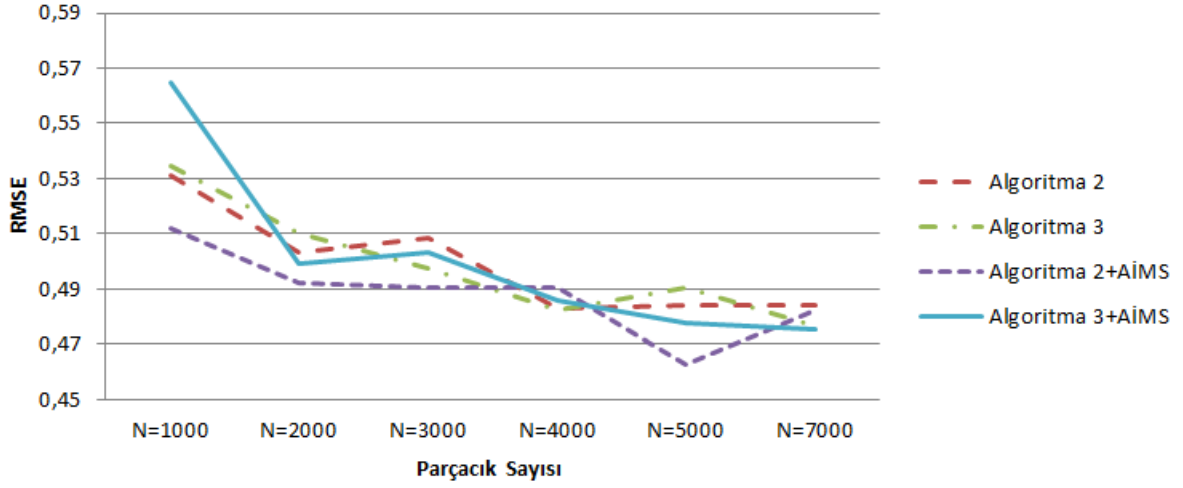
Şekil 6.13 Senaryo 2 Tüm Gürültüler İçin Zaman Grafiği (Yarı Dinamik ve N=5000)

Şekil 6.12 ve Şekil 6.13 incelendiğinde Algoritma 2 ve Algoritma 3'ün hem işlem süresi hem de hata miktarı bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın her iki algoritma için çok fazla değişmediği, işlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği görülmüştür. Algoritma 3+AİMS'in işlem süresi bakımından Algoritma 2+AİMS'ten daha iyi olduğu görülmektedir.

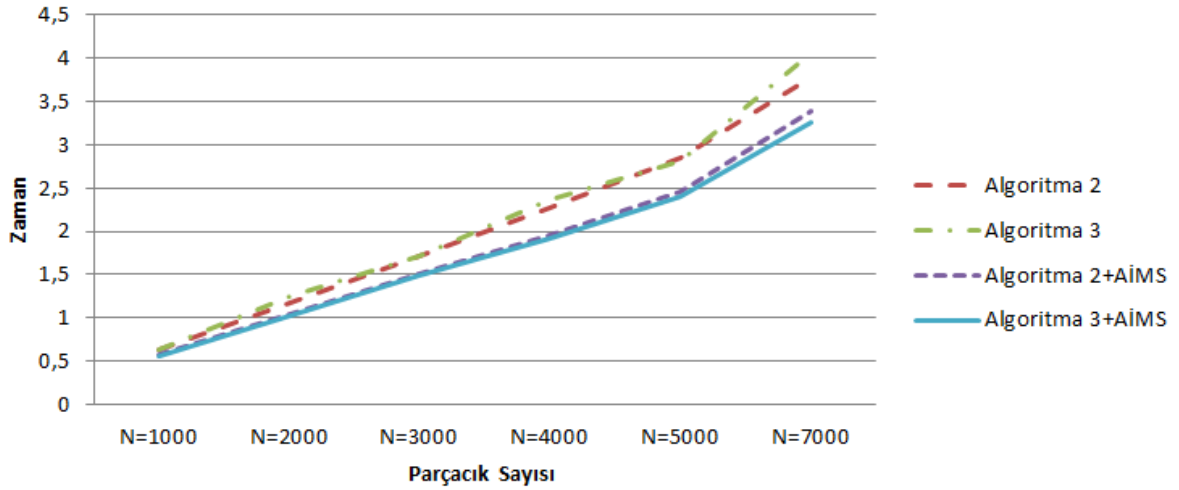
Senaryo 3 için Çizelge 6.6 incelendiğinde;

- Algoritma 2 ve Algoritma 2+AİMS için yapılan işlem süresi karşılaştırmalarında 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 29, 11, 12 ve 9 iyileşme sağlanmıştır. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.
- Algoritma 2 ve Algoritma 3 için yapılan işlem süresi ve hata karşılaştırmalarında (elde edilen sonuçların ortalaması alındığında) her iki algoritmanın birbirine yakın tepkiler gösterdiği görülmektedir.
- Algoritma 3 ve Algoritma 3+AİMS için yapılan işlem süresi karşılaştırmalarda 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 33, 16, 11 ve 10 iyileşme sağlanmıştır. Tüm gürültü seviyeleri için hata miktarları incelendiğinde yaklaşık yüzde 2 iyileşme olduğu görülmüştür.
- Algoritma 2+AİMS ve Algoritma 3+AİMS için yapılan karşılaştırmalarda işlem süresi bakımından Algoritma 3'ün Algoritma 2'den ortalama yüzde 3 daha iyi olduğu, hata bakımından ise benzer tepki verdikleri görülmektedir.
- Algoritma 2 ve Algoritma 3+AİMS için yapılan işlem süresi karşılaştırmalarda Algoritma 3+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 32, 13, 14 ve 11 daha iyi olduğu görülmüştür. Fakat hata miktarlarında bakımından incelendiğinde Algoritma 2'nin 0,1, 0,15 ve 0,2 gürültü seviyeleri için hatanın Algoritma 3+AİMS ile aynı olduğu, 0,01 gürültü seviyesi için ise Algoritma 2'nin yaklaşık yüzde 6 daha iyi olduğu görülmüştür.
- Algoritma 2+AİMS ve Algoritma 3 için yapılan işlem süresi karşılaştırmalarda Algoritma 2+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 44, 16, 10 ve 8 daha iyi olduğu görülmüştür. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.

Senaryo 3 için tüm parçacıklara ait 0,1 gürültü varyanslı benzetim sonuçlarına ait hata ve işlem süresi grafikleri sırasıyla Şekil 6.14 ve Şekil 6.15'te verilmiştir.



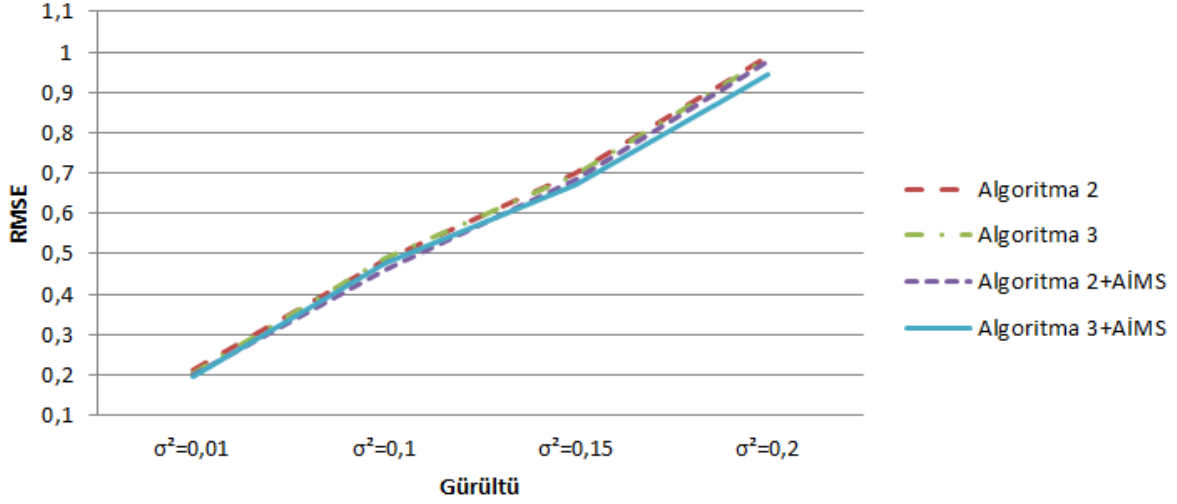
Şekil 6.14 Senaryo 3 Tüm Parçacıklar İçin Hata Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)



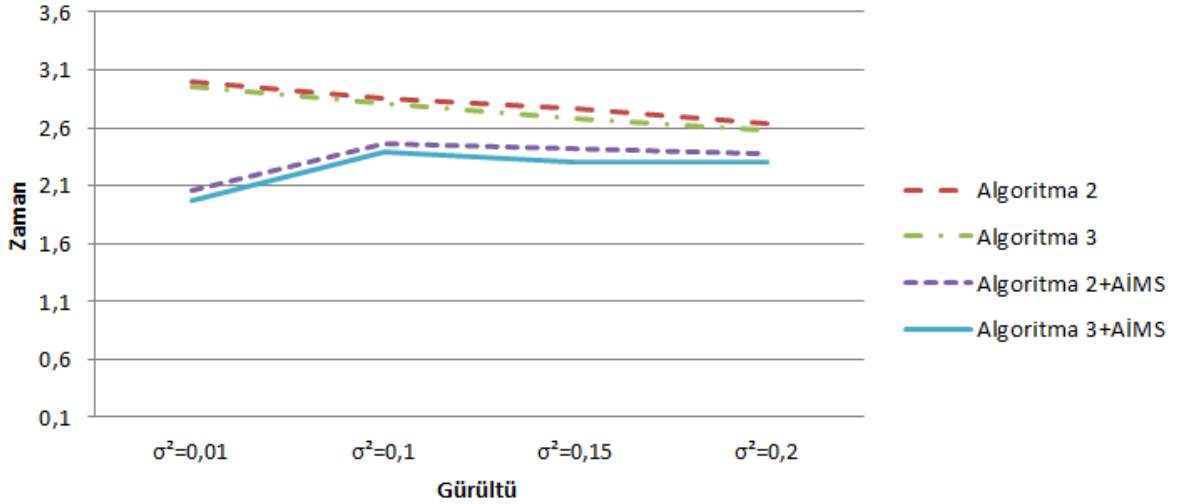
Şekil 6.15 Senaryo 3 Tüm Parçacıklar İçin Zaman Grafiği (Yarı Dinamik ve $\sigma^2=0,1$)

Şekil 6.14 ve Şekil 6.15 incelendiğinde Algoritma 2 ve Algoritma 3'ün işlem süresi ve hata bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın Algoritma 3+AİMS çok fazla değişmediği, Algoritma 2+AİMS için ise azaldığı görülmüştür. İşlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği, Algoritma 3+AİMS'in Algoritma 2+AİMS'ten daha iyi olduğu görülmektedir.

Senaryo 3 için tüm gürültü varyansları ve 5000 parçacık için hata ve işlem süresi grafikleri sırasıyla Şekil 6.16 ve Şekil 6.17’de verilmiştir.



Şekil 6.16 Senaryo 3 Tüm Gürültüler İçin Hata Grafiği (Yarı Dinamik ve N=5000)



Şekil 6.17 Senaryo 3 Tüm Gürültüler İçin Zaman Grafiği (Yarı Dinamik ve N=5000)

Şekil 6.16 ve 6.17 incelendiğinde Algoritma 2 ve Algoritma 3’ün hem işlem süresi hem de hata miktarı bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın her iki algoritma için çok fazla değişmediği, işlem süresi bakımından ise her iki algoritmanın pozitif yönde

etkilendiđi grlmŖtr. Algoritma 3+AİMS'in zaman bakımından Algoritma 2+AİMS'ten daha iyi olduđu grlmektedir.

6.5.2. Dinamik yaklaŖıma ait sonular

Bu kısımda kullanılan AİMS algoritmasındaki pencere boyutu ve hata alt eŖik deđeri iin dinamik yntem kullanılmıŖtır. Dinamik yaklaŖımda alt eŖik deđeri ve pencere boyutu dinamik olarak her bir zaman adımımda yeniden hesaplanmaktadır.

izelge 6.7, izelge 6.8 ve izelge 6.9'da verilen deđerler dinamik pencere boyutu ve hata eŖik deđerleri iin gerekleŖtirilen benzetim sonuları olup, benzetimler sırasıyla Senaryo 1, Senaryo 2 ve Senaryo 3 iin uygulanmıŖtır.

Benzetimlerde MPF baŖlıđı altında anlatılan algoritmalar kullanılmıŖ olup, kullanılan bu algoritmalarından ikisine AİMS algoritması uygulanmıŖtır. Paracık sayısı ve grlt varyansları yukarıda belirtilen deđerlerde kullanılmıŖtır.

izelge 6.7, izelge 6.8 ve izelge 6.9'da yer alan t saniye cinsinde zamanı, RMSE ortalama karesel hatayı, N ise paracık sayısını ifade etmektedir.

Çizelge 6.7 Senaryo 1 –Dinamik Yaklaşım Sonuçları

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 1	$\sigma^2=0,01$	0,1040	1,4255	0,0634	3,1843	0,0501	5,9043	0,0498	8,6347	0,0475	12,9441	0,0468	21,4765
	$\sigma^2=0,1$	0,7492	1,4273	0,5903	3,2552	0,5748	5,9386	0,5053	8,6866	0,5138	12,4655	0,5083	21,6032
	$\sigma^2=0,15$	1,4581	1,3711	1,1416	3,2077	1,1147	5,6699	1,0733	8,7828	1,0425	12,5262	0,9763	21,3828
	$\sigma^2=0,2$	2,8519	1,5060	1,8377	3,4697	1,6194	5,5431	1,6487	8,7207	1,6660	12,1270	1,4604	21,1736
Algoritma 2	$\sigma^2=0,01$	0,1379	0,7089	0,1234	1,2776	0,1084	1,9274	0,1031	2,5101	0,1025	3,1559	0,0988	4,3022
	$\sigma^2=0,1$	0,3413	0,6748	0,2851	0,9866	0,2840	1,8029	0,2901	2,3691	0,2795	3,0215	0,2820	3,2135
	$\sigma^2=0,15$	0,4850	0,6203	0,4594	1,1221	0,4519	1,6380	0,4211	2,1405	0,4293	2,6550	0,4133	3,6654
	$\sigma^2=0,2$	0,6784	0,5705	0,6024	1,0618	0,5957	1,5695	0,5890	2,1628	0,5773	2,6492	0,5808	3,4804
Algoritma 3	$\sigma^2=0,01$	0,1366	0,6994	0,1111	1,3241	0,1036	1,8533	0,1078	2,5416	0,1028	3,0531	0,1020	4,3493
	$\sigma^2=0,1$	0,3251	0,6489	0,3129	1,2558	0,2920	1,7338	0,2821	2,3985	0,2800	2,8468	0,2854	4,1204
	$\sigma^2=0,15$	0,4911	0,6359	0,4695	1,1396	0,4202	1,6750	0,4129	2,3359	0,4128	2,9849	0,4100	3,7910
	$\sigma^2=0,2$	0,6786	0,5934	0,6229	1,1612	0,6026	1,5975	0,6252	2,0788	0,5752	2,6452	0,5845	3,6150
Algoritma 2 + AİMS	$\sigma^2=0,01$	0,1396	0,4625	0,1379	0,7042	0,1155	1,1660	0,1099	1,5071	0,1074	1,8586	0,1098	2,4214
	$\sigma^2=0,1$	0,3350	0,5265	0,3167	0,8818	0,3074	1,3854	0,2898	1,8354	0,2790	2,2631	0,2771	3,1843
	$\sigma^2=0,15$	0,4964	0,5577	0,4661	0,9831	0,4408	1,4118	0,4430	1,8962	0,4282	2,2417	0,4143	2,9461
	$\sigma^2=0,2$	0,6695	0,5375	0,6573	0,9557	0,6288	1,4512	0,5786	1,8465	0,5861	2,2330	0,5703	2,5921
Algoritma 3 + AİMS	$\sigma^2=0,01$	0,1454	0,4566	0,1247	0,7630	0,1171	1,1865	0,0927	1,7397	0,1092	1,7467	0,1313	2,1456
	$\sigma^2=0,1$	0,3132	0,5427	0,3111	0,9965	0,2910	1,4440	0,2838	1,8651	0,1092	1,7467	0,2836	3,1214
	$\sigma^2=0,15$	0,5302	0,5531	0,4358	1,0190	0,4434	1,3800	0,4375	1,9204	0,4254	2,2969	0,4229	3,2880
	$\sigma^2=0,2$	0,7962	0,4929	0,6744	0,9549	0,6883	1,4770	0,6506	1,9152	0,6303	2,1592	0,5809	3,0475

Çizelge 6.8 Senaryo 2 – Dinamik Yaklaşım Sonuçları

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 1	$\sigma^2=0,01$	0,1355	1,3950	0,0805	3,1968	0,0701	5,8932	0,0621	8,5624	0,0697	12,8984	0,0571	21,8267
	$\sigma^2=0,1$	0,9545	2,1750	0,7078	3,3020	0,7191	8,3502	0,6127	8,7479	0,6188	13,0469	0,5979	22,0919
	$\sigma^2=0,15$	2,0408	1,6016	1,4906	4,0051	1,2522	5,6193	1,1610	8,5822	1,0869	12,5395	1,1280	21,5944
	$\sigma^2=0,2$	3,7174	1,9736	2,8289	3,9204	2,3329	7,1571	1,9317	8,5040	2,0022	12,4866	2,1384	21,2601
Algoritma 2	$\sigma^2=0,01$	0,2577	0,7157	0,2199	1,2758	0,2124	1,9353	0,2054	2,4477	0,2074	3,1650	0,2070	4,1833
	$\sigma^2=0,1$	0,4183	0,6813	0,3466	1,2048	0,3573	1,8455	0,3593	2,2857	0,3440	3,0196	0,3276	3,9661
	$\sigma^2=0,15$	0,5277	0,6351	0,5249	1,1914	0,4988	1,6820	0,5039	2,2044	0,5022	2,7238	0,4884	3,6837
	$\sigma^2=0,2$	0,8845	0,6252	0,7850	1,1535	0,7576	1,5992	0,7125	2,1097	0,6676	2,5732	0,7191	3,5886
Algoritma 3	$\sigma^2=0,01$	0,2436	0,7060	0,2182	1,2755	0,2063	2,0159	0,2146	2,4497	0,2225	2,9747	0,2032	4,1804
	$\sigma^2=0,1$	0,4428	0,6555	0,3394	1,2055	0,3545	1,8980	0,3505	2,2579	0,3554	3,1390	0,3461	3,9511
	$\sigma^2=0,15$	0,6367	0,6494	0,5457	1,1981	0,5105	1,7312	0,5033	2,2545	0,4881	2,8110	0,4974	3,8786
	$\sigma^2=0,2$	0,8059	0,6181	0,7468	1,1565	0,6971	1,6694	0,7003	2,1593	0,7169	2,8609	0,6985	3,7265
Algoritma 2 + AİMS	$\sigma^2=0,01$	0,2333	0,4472	0,2222	0,7913	0,2115	1,0794	0,1998	1,4190	0,2031	1,7907	0,2013	2,7532
	$\sigma^2=0,1$	0,3807	0,5258	0,3577	0,9543	0,3654	1,2545	0,3646	1,7974	0,3478	2,2024	0,3633	3,0969
	$\sigma^2=0,15$	0,5774	0,5422	0,5327	0,9827	0,5304	1,4163	0,4904	1,9325	0,4928	2,2102	0,4924	3,1797
	$\sigma^2=0,2$	0,9244	0,5414	0,7848	0,9584	0,7322	1,4266	0,7242	1,8708	0,7102	2,3064	0,6520	3,1707
Algoritma 3 + AİMS	$\sigma^2=0,01$	0,2508	0,5962	0,2103	0,7942	0,1973	1,2575	0,2039	1,5890	0,2066	2,0913	0,1943	2,7624
	$\sigma^2=0,1$	0,4003	0,5471	0,3568	0,9953	0,3678	1,1169	0,3693	1,8529	0,3563	2,2969	0,3468	3,3467
	$\sigma^2=0,15$	0,5850	0,5130	0,5512	0,8034	0,5047	1,2958	0,5078	1,7449	0,4917	2,1922	0,4957	3,5265
	$\sigma^2=0,2$	0,9693	0,4165	0,7541	1,0110	0,7627	1,3726	0,7231	1,9063	0,7263	2,4183	0,6806	2,9985

Çizelge 6.9 Senaryo 3 – Dinamik Yaklaşım Sonuçları

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 1	$\sigma^2=0,01$	0,2259	1,3814	0,1310	3,3227	0,1012	5,8019	0,0986	8,8755	0,0946	12,6163	0,0928	21,9441
	$\sigma^2=0,1$	1,2597	1,3586	1,0330	3,3096	0,8736	5,8333	0,8674	8,8771	0,8342	12,7216	0,7214	21,9434
	$\sigma^2=0,15$	2,8748	1,4716	2,0290	3,1826	1,8862	5,6207	1,7415	10,4953	1,6265	12,3564	1,6175	21,1936
	$\sigma^2=0,2$	5,5993	1,7706	2,8860	3,1059	2,7044	5,5141	2,6191	8,4283	2,5273	12,2517	2,3059	21,0791
Algoritma 2	$\sigma^2=0,01$	0,2541	0,6933	0,2316	1,2424	0,2171	1,8217	0,2119	2,3614	0,2115	3,0021	0,2021	4,1009
	$\sigma^2=0,1$	0,5311	0,6374	0,5030	1,1668	0,5082	1,7137	0,4830	2,2631	0,4839	2,8577	0,4838	3,7804
	$\sigma^2=0,15$	0,8901	0,6327	0,7532	1,1579	0,6951	1,6868	0,7208	2,1964	0,7010	2,7723	0,6810	3,8148
	$\sigma^2=0,2$	1,1428	0,5962	1,1134	1,1131	1,0299	1,6176	0,9849	2,0616	0,9910	2,6415	0,9276	3,6122
Algoritma 3	$\sigma^2=0,01$	0,2590	0,6858	0,2197	1,3054	0,2109	1,8078	0,2150	2,5179	0,2042	2,9503	0,1944	4,3019
	$\sigma^2=0,1$	0,5347	0,6362	0,5104	1,2410	0,4977	1,7061	0,4826	2,3652	0,4906	2,8138	0,4764	4,0644
	$\sigma^2=0,15$	0,8610	0,6185	0,7175	1,1308	0,7039	1,6484	0,7103	2,1496	0,6945	2,6806	0,6999	3,6851
	$\sigma^2=0,2$	1,1831	0,5977	1,1695	1,0797	1,1602	1,5964	1,0438	2,0702	0,9872	2,5837	0,9647	3,5766
Algoritma 2 + AİMS	$\sigma^2=0,01$	0,2768	0,5527	0,2334	0,9940	0,2066	1,3597	0,2066	1,8203	0,2056	2,1834	0,2061	3,0242
	$\sigma^2=0,1$	0,5476	0,6083	0,4872	1,0816	0,4861	1,5182	0,4870	2,0023	0,4710	2,5156	0,4717	3,4567
	$\sigma^2=0,15$	0,8582	0,5576	0,7702	1,0610	0,7310	1,5189	0,6741	1,9970	0,6939	2,4221	0,6816	3,3481
	$\sigma^2=0,2$	1,2311	0,5533	1,1288	1,0381	0,9899	1,5168	1,0104	1,7123	1,0120	2,3126	0,9128	3,4886
Algoritma 3 + AİMS	$\sigma^2=0,01$	0,2605	0,5406	0,2251	0,9479	0,2066	1,3246	0,2157	1,7680	0,2091	2,0404	0,1872	3,2045
	$\sigma^2=0,1$	0,5182	0,5823	0,5058	1,0013	0,4921	1,5299	0,4857	2,0024	0,4816	1,7613	0,4882	2,5310
	$\sigma^2=0,15$	0,8843	0,5240	0,7524	0,7375	0,6864	1,5108	0,7086	1,9629	0,6760	2,4217	0,6659	3,3286
	$\sigma^2=0,2$	1,2023	0,5516	1,1399	0,7758	1,0072	1,4658	0,9903	1,7307	1,0456	1,7234	1,0515	3,1746

Çizelge 6.7, Çizelge 6.8 ve Çizelge 6.9'da görüldüğü gibi Algoritma 1'de parçacık sayısı arttıkça işlem süresi uzamakta, gürültü miktarı arttıkça ise hata değeri artmaktadır. İlgili çizelgelerde Algoritma 2 ve Algoritma 3'ün ise Algoritma 1'e göre daha kararlı çalıştığı görülmektedir. Buna ek olarak Algoritma 1'e AİMS algoritmasının uygulanamaması nedeniyle Çizelge 6.10, Çizelge 6.11 ve Çizelge 6.12'deki karşılaştırma çizelgelerine Algoritma 1 eklenmemiştir.

Çizelge 6.10, Çizelge 6.11 ve Çizelge 6.12'de görüldüğü gibi her bir senaryo için;

- Algoritma 2 ve Algoritma 2+AİMS,
- Algoritma 2 ve Algoritma 3,
- Algoritma 3 ve Algoritma 3+AİMS,
- Algoritma 2+AİMS ve Algoritma 3+AİMS,
- Algoritma 2 ve Algoritma 3+AİMS,
- Algoritma 2+AİMS ve Algoritma 3.

Karşılaştırmaları yapılmıştır. Karşılaştırmalar yüzde bazında olup, pozitif değerler sistemdeki iyileşmeyi, negatif olanlar ise sistemdeki kötüleşmeyi ifade etmektedir.

Tablolarda yer alan Algoritma A-B karşılaştırmaları Eşitlik 6.3'teki ifade ile hesaplanmıştır.

Çizelge 6.10 Senaryo 1 –Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algorithm 2-2+AiMS	$\sigma^2=0,01$	-1,23	34,76	-11,75	44,88	-6,55	39,50	-6,60	39,96	-4,78	41,11	-11,13	43,72
	$\sigma^2=0,1$	1,85	21,98	-11,08	25,59	-8,24	23,16	0,10	22,53	0,18	25,10	1,74	0,91
	$\sigma^2=0,15$	-2,35	10,09	-1,46	12,39	2,46	13,81	-5,20	11,41	0,26	15,57	-0,24	19,62
	$\sigma^2=0,2$	1,31	5,78	-9,11	9,99	-5,56	7,54	1,77	14,62	-1,52	15,71	1,81	25,52
Algorithm 2-3	$\sigma^2=0,01$	0,94	1,34	9,97	-3,64	4,43	3,84	-4,56	-1,25	-0,29	3,26	-3,24	-1,09
	$\sigma^2=0,1$	4,75	3,84	-9,75	-5,97	-2,82	3,83	2,76	-1,24	-0,18	5,78	-1,21	-28,22
	$\sigma^2=0,15$	-1,26	-2,51	-2,20	-1,56	7,01	-2,26	1,95	-9,13	3,84	-12,43	0,80	-3,43
	$\sigma^2=0,2$	-0,03	-4,01	-3,40	-9,36	-1,16	-1,78	-6,15	3,88	0,36	0,15	-0,64	-3,87
Algorithm 3-3+AiMS	$\sigma^2=0,01$	-6,44	34,72	-12,24	42,38	-13,03	35,98	14,01	31,55	-6,23	42,79	-28,73	50,67
	$\sigma^2=0,1$	3,66	16,37	0,58	20,65	0,34	16,71	-0,60	22,24	61,00	38,64	0,63	24,25
	$\sigma^2=0,15$	-7,96	13,02	7,18	10,58	-5,52	17,61	-5,96	17,79	-3,05	23,05	-3,15	13,27
	$\sigma^2=0,2$	-17,33	16,94	-8,27	17,77	-14,22	7,54	-4,06	7,87	-9,58	18,37	0,62	15,70
Algorithm 2+AiMS - 3+AiMS	$\sigma^2=0,01$	-4,15	1,28	9,57	-8,35	-1,39	-1,76	15,65	-15,43	-1,68	6,02	-19,58	11,39
	$\sigma^2=0,1$	6,51	-3,08	1,77	-13,01	5,34	-4,23	2,07	-1,62	60,86	22,82	-2,35	1,98
	$\sigma^2=0,15$	-6,81	0,82	6,50	-3,65	-0,59	2,25	1,24	-1,28	0,65	-2,46	-2,08	-11,61
	$\sigma^2=0,2$	-18,92	8,30	-2,60	0,08	-9,46	-1,78	-12,44	-3,72	-7,54	3,30	-1,86	-17,57
Algorithm 2-3+AiMS	$\sigma^2=0,01$	-5,44	35,59	-1,05	40,28	-8,03	38,44	10,09	30,69	-6,54	44,65	-32,89	50,13
	$\sigma^2=0,1$	8,23	19,58	-9,12	15,91	-2,46	19,91	2,17	21,27	60,93	42,19	-0,57	2,87
	$\sigma^2=0,15$	-9,32	10,83	5,14	9,19	1,88	15,75	-3,89	10,28	0,91	13,49	-2,32	10,30
	$\sigma^2=0,2$	-17,36	13,60	-11,95	10,07	-15,54	5,89	-10,46	11,45	-9,18	18,50	-0,02	12,44
Algorithm 2+AiMS -3	$\sigma^2=0,01$	2,15	-51,22	19,43	-88,03	10,30	-58,95	1,91	-68,64	4,28	-64,27	7,10	-79,62
	$\sigma^2=0,1$	2,96	-23,25	1,20	-42,41	5,01	-25,15	2,66	-30,68	-0,36	-25,79	-3,00	-29,40
	$\sigma^2=0,15$	1,07	-14,02	-0,73	-15,92	4,67	-18,64	6,79	-23,19	3,60	-33,15	1,04	-28,68
	$\sigma^2=0,2$	-1,36	-10,40	5,23	-21,50	4,17	-10,08	-8,05	-12,58	1,86	-18,46	-2,49	-39,46

Çizelge 6.11 Senaryo 2 –Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi

	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 2-2+AİMS	$\sigma^2=0,01$	9,47	37,52	-1,05	37,98	0,42	44,23	2,73	42,03	2,07	43,42	2,75	34,19
	$\sigma^2=0,1$	8,99	22,82	-3,20	20,79	-2,27	32,02	-1,48	21,36	-1,10	27,06	-10,90	21,92
	$\sigma^2=0,15$	-9,42	14,63	-1,49	17,52	-6,34	15,80	2,68	12,33	1,87	18,86	-0,82	13,68
	$\sigma^2=0,2$	-4,51	13,40	0,03	16,91	3,35	10,79	-1,64	11,32	-6,38	10,37	9,33	11,65
Algoritma 2-3	$\sigma^2=0,01$	5,47	1,36	0,77	0,02	2,87	-4,16	-4,48	-0,08	-7,28	6,01	1,84	0,07
	$\sigma^2=0,1$	-5,86	3,79	2,08	-0,06	0,78	-2,84	2,45	1,22	-3,31	-3,95	-5,65	0,38
	$\sigma^2=0,15$	-20,66	-2,25	-3,96	-0,56	-2,35	-2,93	0,12	-2,27	2,81	-3,20	-1,84	-5,29
	$\sigma^2=0,2$	8,89	1,14	4,87	-0,26	7,99	-4,39	1,71	-2,35	-7,38	-11,18	2,86	-3,84
Algoritma 3-3+AİMS	$\sigma^2=0,01$	-2,96	15,55	3,62	37,73	4,36	37,62	4,99	35,13	7,15	29,70	4,38	33,92
	$\sigma^2=0,1$	9,60	16,54	-5,13	17,44	-3,75	41,15	-5,36	17,94	-0,25	26,83	-0,20	15,30
	$\sigma^2=0,15$	8,12	21,00	-1,01	32,94	1,14	25,15	-0,89	22,60	-0,74	22,01	0,34	9,08
	$\sigma^2=0,2$	-20,28	32,62	-0,98	12,58	-9,41	17,78	-3,26	11,72	-1,31	15,47	2,56	19,54
Algoritma 2+AİMS - 3+AİMS	$\sigma^2=0,01$	-7,50	-33,32	5,36	-0,37	6,71	-16,50	-2,05	-11,98	-1,72	-16,79	3,48	-0,33
	$\sigma^2=0,1$	-5,15	-4,05	0,25	-4,30	-0,66	10,97	-1,29	-3,09	-2,44	-4,29	4,54	-8,07
	$\sigma^2=0,15$	-1,32	5,39	-3,47	18,25	4,85	8,51	-3,55	9,71	0,22	0,81	-0,67	-10,91
	$\sigma^2=0,2$	-4,86	23,07	3,91	-5,49	-4,17	3,79	0,15	-1,90	-2,27	-4,85	-4,39	5,43
Algoritma 2-3+AİMS	$\sigma^2=0,01$	2,68	16,70	4,37	37,75	7,11	35,02	0,73	35,08	0,39	33,92	6,14	33,97
	$\sigma^2=0,1$	4,30	19,70	-2,94	17,39	-2,94	39,48	-2,78	18,94	-3,58	23,93	-5,86	15,62
	$\sigma^2=0,15$	-10,86	19,23	-5,01	32,57	-1,18	22,96	-0,77	20,84	2,09	19,52	-1,49	4,27
	$\sigma^2=0,2$	-9,59	33,38	3,94	12,35	-0,67	14,17	-1,49	9,64	-8,79	6,02	5,35	16,44
Algoritma 2+AİMS -3	$\sigma^2=0,01$	-4,41	-57,87	1,80	-61,19	2,46	-86,76	-7,41	-72,64	-9,55	-66,12	-0,94	-51,84
	$\sigma^2=0,1$	-16,31	-24,67	5,12	-26,32	2,98	-51,30	3,87	-25,62	-2,19	-42,53	4,73	-27,58
	$\sigma^2=0,15$	-10,27	-19,77	-2,44	-21,92	3,75	-22,23	-2,63	-16,66	0,95	-27,18	-1,02	-21,98
	$\sigma^2=0,2$	12,82	-14,17	4,84	-20,67	4,79	-17,02	3,30	-15,42	-0,94	-24,04	-7,13	-17,53

Çizelge 6.12 Senaryo 3 –Dinamik Yaklaşım İçin Karşılaştırma Çizelgesi

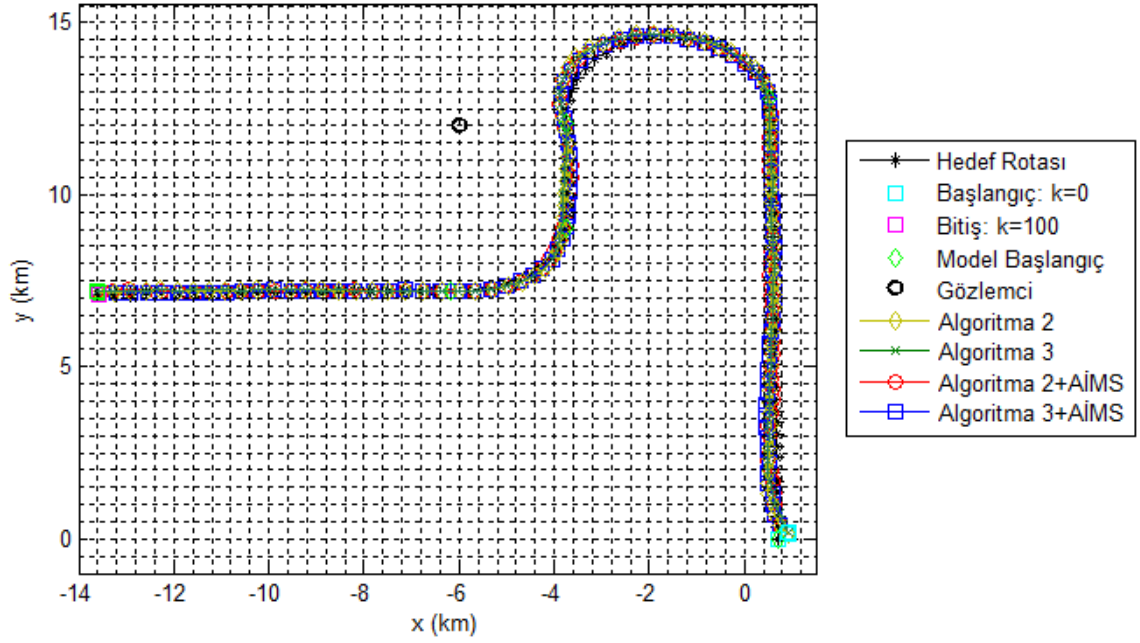
	Gürültü Varyansı	N=1000		N=2000		N=3000		N=4000		N=5000		N=7000	
		RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Algoritma 2-2+AİMS	$\sigma^2=0,01$	-8,93	20,28	-0,78	19,99	4,84	25,36	2,50	22,91	2,79	27,27	-1,98	26,26
	$\sigma^2=0,1$	-3,11	4,57	3,14	7,30	4,35	11,41	-0,83	11,52	2,67	11,97	2,50	8,56
	$\sigma^2=0,15$	3,58	11,87	-2,26	8,37	-5,16	9,95	6,48	9,08	1,01	12,63	-0,09	12,23
	$\sigma^2=0,2$	-7,73	7,20	-1,38	6,74	3,88	6,23	-2,59	16,94	-2,12	12,45	1,60	3,42
Algoritma 2-3	$\sigma^2=0,01$	-1,93	1,08	5,14	-5,07	2,86	0,76	-1,46	-6,63	3,45	1,73	3,81	-4,90
	$\sigma^2=0,1$	-0,68	0,19	-1,47	-6,36	2,07	0,44	0,08	-4,51	-1,38	1,54	1,53	-7,51
	$\sigma^2=0,15$	3,27	2,24	4,74	2,34	-1,27	2,28	1,46	2,13	0,93	3,31	-2,78	3,40
	$\sigma^2=0,2$	-3,53	-0,25	-5,04	3,00	-12,65	1,31	-5,98	-0,42	0,38	2,19	-4,00	0,99
Algoritma 3-3+AİMS	$\sigma^2=0,01$	-0,58	21,17	-2,46	27,39	2,04	26,73	-0,33	29,78	-2,40	30,84	3,70	25,51
	$\sigma^2=0,1$	3,09	8,47	0,90	19,32	1,13	10,33	-0,64	15,34	1,83	37,40	-2,48	37,73
	$\sigma^2=0,15$	-2,71	15,28	-4,86	34,78	2,49	8,35	0,24	8,69	2,66	9,66	4,86	9,67
	$\sigma^2=0,2$	-1,62	7,71	2,53	28,15	13,19	8,18	5,13	16,40	-5,92	33,30	-9,00	11,24
Algoritma 2+AİMS - 3+AİMS	$\sigma^2=0,01$	5,89	2,19	3,56	4,64	0,00	2,58	-4,40	2,87	-1,70	6,55	9,17	-5,96
	$\sigma^2=0,1$	5,37	4,27	-3,82	7,42	-1,23	-0,77	0,27	0,00	-2,25	29,98	-3,50	26,78
	$\sigma^2=0,15$	-3,04	6,03	2,31	30,49	6,10	0,53	-5,12	1,71	2,58	0,02	2,30	0,58
	$\sigma^2=0,2$	2,34	0,31	-0,98	25,27	-1,75	3,36	1,99	-1,07	-3,32	25,48	-15,20	9,00
Algoritma 2-3+AİMS	$\sigma^2=0,01$	-2,52	22,03	2,81	23,70	4,84	27,29	-1,79	25,13	1,13	32,03	7,37	21,86
	$\sigma^2=0,1$	2,43	8,64	-0,56	14,18	3,17	10,73	-0,56	11,52	0,48	38,37	-0,91	33,05
	$\sigma^2=0,15$	0,65	17,18	0,11	36,31	1,25	10,43	1,69	10,63	3,57	12,65	2,22	12,75
	$\sigma^2=0,2$	-5,21	7,48	-2,38	30,30	2,20	9,38	-0,55	16,05	-5,51	34,76	-13,36	12,11
Algoritma 2+AİMS -3	$\sigma^2=0,01$	6,43	-24,08	5,87	-31,33	-2,08	-32,96	-4,07	-38,32	0,68	-35,12	5,68	-42,25
	$\sigma^2=0,1$	2,36	-4,59	-4,76	-14,74	-2,39	-12,38	0,90	-18,12	-4,16	-11,85	-1,00	-17,58
	$\sigma^2=0,15$	-0,33	-10,92	6,84	-6,58	3,71	-8,53	-5,37	-7,64	-0,09	-10,67	-2,68	-10,07
	$\sigma^2=0,2$	3,90	-8,02	-3,61	-4,01	-17,20	-5,25	-3,31	-20,90	2,45	-11,72	-5,69	-2,52

Senaryo 1 için Çizelge 6.10 incelendiğinde;

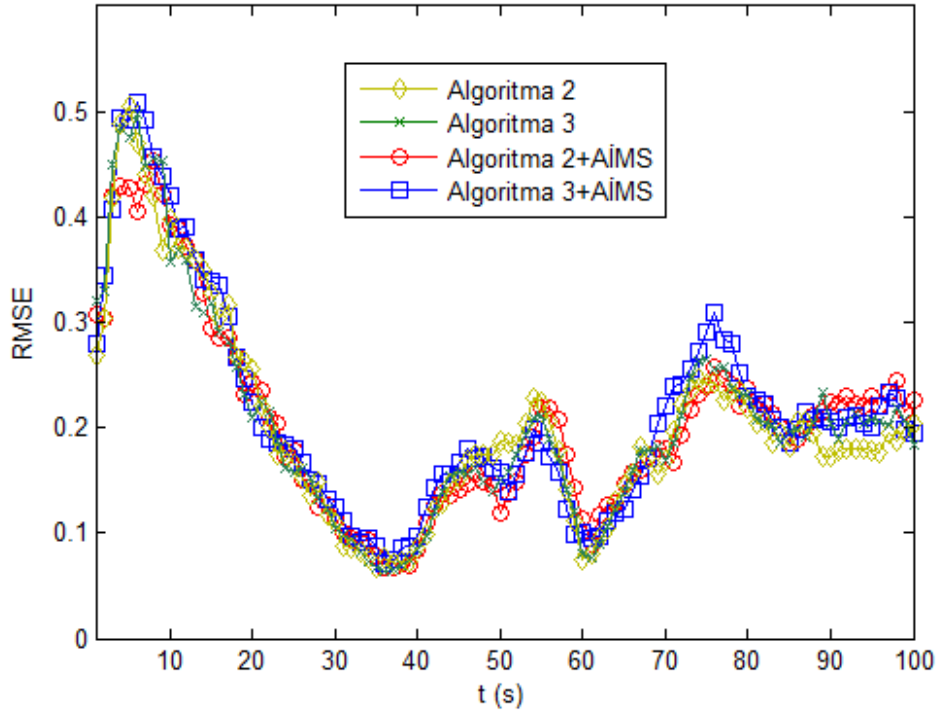
- Algoritma 2 ve Algoritma 2+AİMS için yapılan işlem süresi karşılaştırmalarında 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 40, 19, 13 ve 13 iyileşme sağlanmıştır. Fakat hata miktarlarında ise sırasıyla yüzde 7, 2, 1 ve 1 artış meydana gelmiştir.
- Algoritma 2 ve Algoritma 3 için yapılan hata karşılaştırmalarında Algoritma 3'ün yaklaşık yüzde 2 daha iyi olduğu görülmektedir. İşlem süresi karşılaştırmalarında (elde edilen sonuçların ortalaması alındığında) her iki algoritmanın birbirine yakın tepkiler gösterdiği görülmektedir.
- Algoritma 3 ve Algoritma 3+AİMS için yapılan işlem zamanı karşılaştırmalarda 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 39, 23, 15 ve 14 iyileşme sağlanmıştır. Fakat hata miktarlarında ise 0,01, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla yüzde 8, 3 ve 8 artış meydana gelmiştir. 0,1 gürültü seviyesi için ise yüzde 10 düşüş meydana gelmiştir.
- Algoritma 2+AİMS ve Algoritma 3+AİMS için yapılan karşılaştırmalarda işlem zamanı bakımından 0,1 gürültü seviyesi için Algoritma 3'ün Algoritma 2'den ortalama yüzde 12 daha iyi olduğu, 0,2 gürültü seviyesi için Algoritma 2'nin Algoritma 3'ten ortalama yüzde 8 daha iyi olduğu görülmektedir. Diğer gürültü seviyeleri için zaman aynı kalmıştır. Hata bakımından ise benzer tepkiler verdikleri görülmektedir.
- Algoritma 2 ve Algoritma 3+AİMS için yapılan işlem zamanı karşılaştırmalarında Algoritma 3+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 39, 2, 11 ve 11 daha iyi olduğu görülmüştür. Fakat hata miktarları bakımından incelendiğinde 0,01, 0,15 ve 0,2 gürültü seviyeleri için Algoritma 2'nin sırasıyla ortalama yüzde 7, 1 ve 10 daha iyi olduğu görülmüştür. 0,1 gürültü seviyesi için Algoritma 3+AİMS'in yaklaşık yüzde 9 daha iyi olduğu görülmektedir.
- Algoritma 2+AİMS ve Algoritma 3 için yapılan işlem süresi karşılaştırmalarda Algoritma 2+AİMS'in 0,01, 0,1 ve 0,15 gürültü seviyeleri için sırasıyla ortalama yüzde 68, 29, 22 ve 18 daha iyi olduğu görülmüştür.

Fakat hata miktarlarında bakımından incelendiğinde Algoritma 3'ün sırasıyla ortalama yüzde 7, 1 ve 2 daha iyi olduğu görülmüştür. 0,2 gürültü seviyesi için hatada farklılık oluşmamıştır.

Senaryo 1 için gerçekleştirilen MATLAB benzetimine ilişkin hedef takip ve hata grafikleri Şekil 6.18 ve 6.19'da sırasıyla verilmiştir. İlgili benzetim gözlem gürültüsü varyansı $\sigma^2=0,1$ ve parçacık sayısı 5000 için gerçekleştirilmiştir.

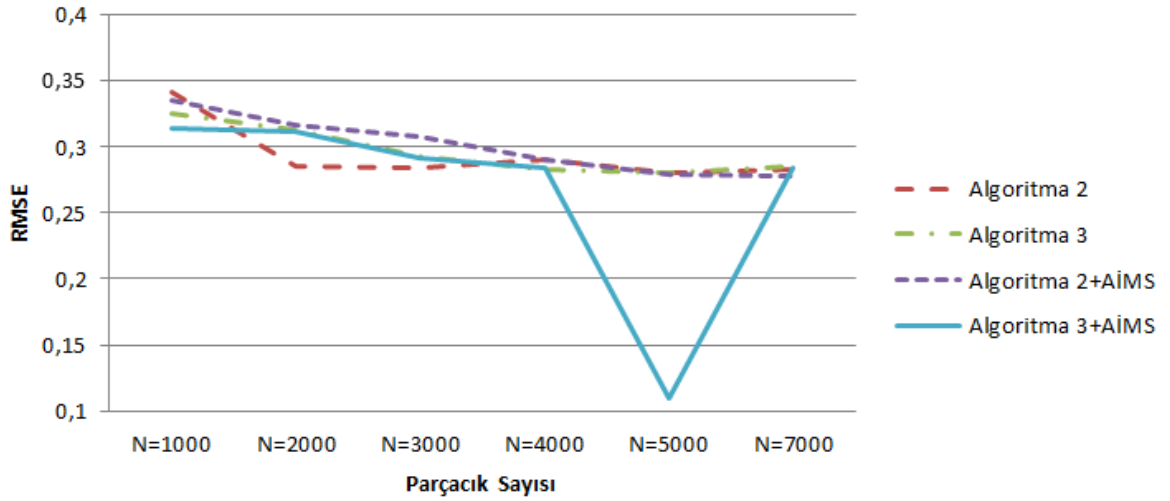


Şekil 6.18 Senaryo 1 Hedef Takibi (Dinamik, $\sigma^2=0,1$ ve $N=5000$)

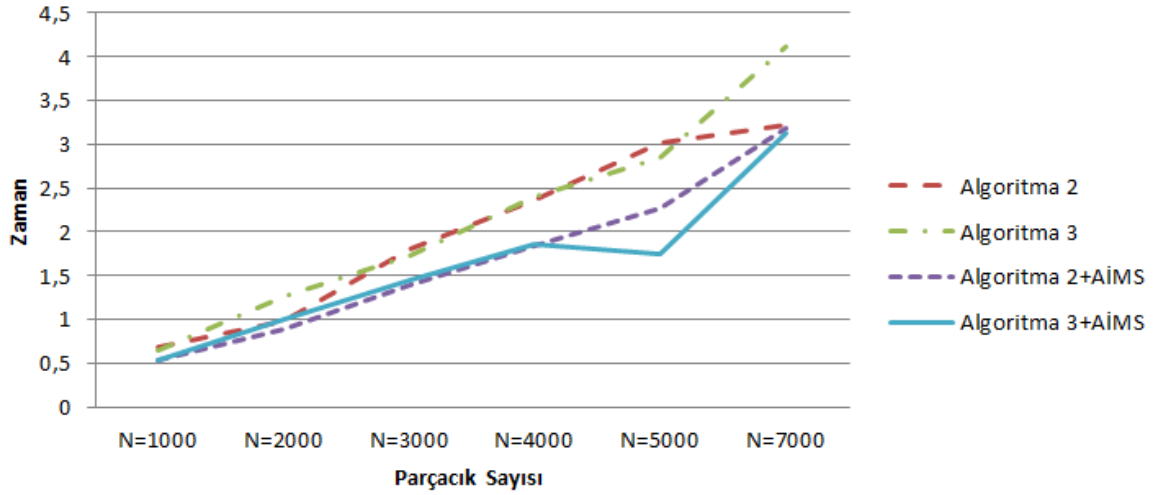


Şekil 6.19 Senaryo 2 RMSE Grafiği (Dinamik, $\sigma^2=0,1$ ve $N=5000$)

Senaryo 1 için tüm parçacıklara ait 0,1 gürültü varyanslı benzetim sonuçlarına ait hata ve işlem süresi grafikleri sırasıyla Şekil 6.20 ve Şekil 6.21'de verilmiştir.



Şekil 6.20 Senaryo 1 Tüm Parçacıklar İçin Hata Grafiği (Dinamik ve $\sigma^2=0,1$)

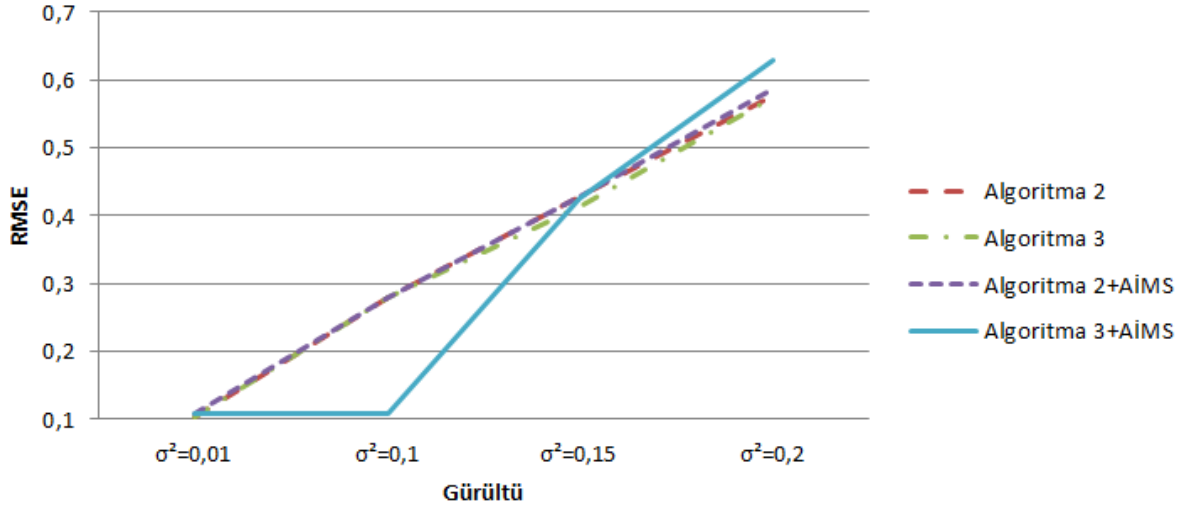


Şekil 6.21 Senaryo 1 Tüm Parçacıklar İçin Zaman Grafiği (Dinamik ve $\sigma^2=0,1$)

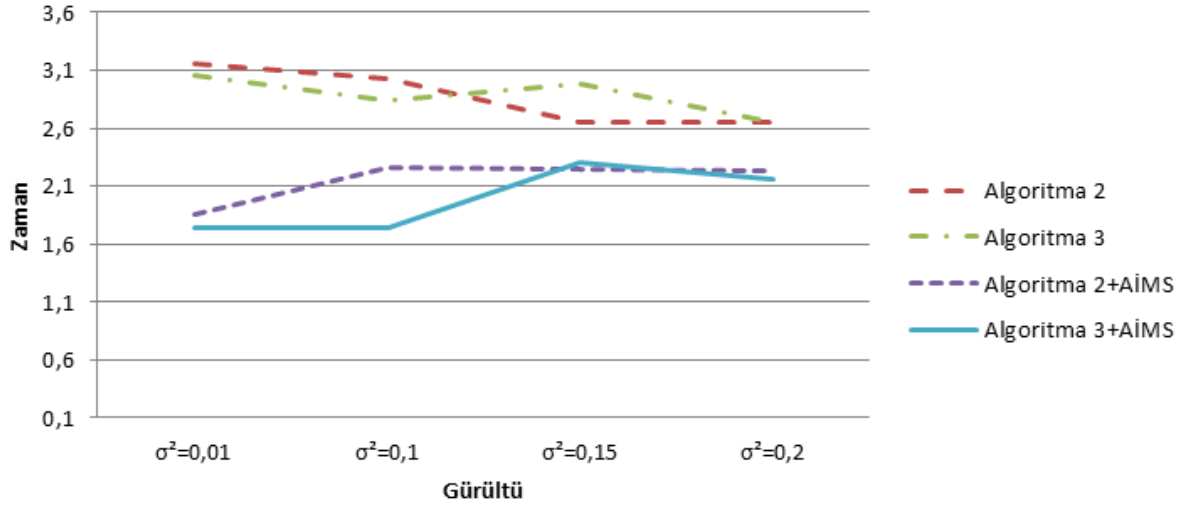
Şekil 6.18 ve Şekil 6.19’te yer alan grafikler Şekil 6.20 ve Şekil 6.21 yardımıyla 5000 parçacık için incelendiğinde Algoritma 3+AİMS’in işlem süresi ve RMSE bakımından en iyi sonucu verdiği, Algoritma 3+AİMS’in Algoritma 3’ten daha düşük RMSE ve işlem süresi değerine sahip olduğu görülmektedir. Algoritma 2 ve Algoritma 2+AİMS ise 5000 parçacık için aynı RMSE değerine sahiptir.

Şekil 6.20 ve Şekil 6.21 incelendiğinde Algoritma 2 ve Algoritma 3’ün işlem süresi ve hata bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın 5000 parçacıkta Algoritma 3+AİMS için azaldığı, Algoritma 2+AİMS için belirgin bir değişim görülmemiştir. İşlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği, Algoritma 3+AİMS’in Algoritma 2+AİMS’ten daha iyi olduğu görülmektedir.

Senaryo 1 için tüm gürültü varyansları ve 5000 parçacık için hata ve işlem süresi grafikleri sırasıyla Şekil 6.22 ve Şekil 6.23’te verilmiştir.



Şekil 6.22 Senaryo 1 Tüm Gürültüler İçin Hata Grafiği (Dinamik ve N=5000)



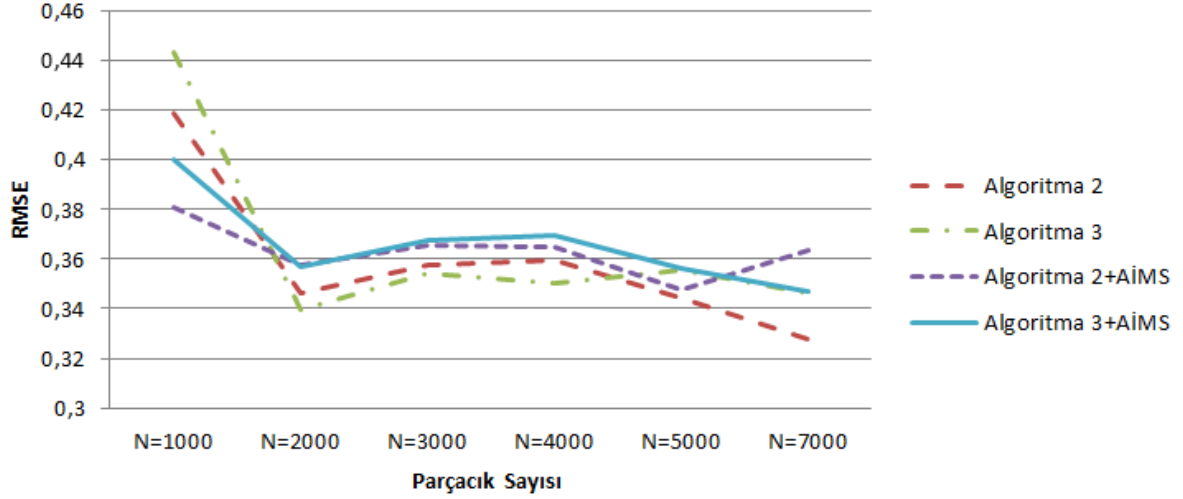
Şekil 6.23 Senaryo 1 Tüm Gürültüler İçin Zaman Grafiği (Dinamik ve N=5000)

Şekil 6.22 ve 6.23 incelendiğinde Algoritma 2 ve Algoritma 3'ün hem işlem süresi hem de hata miktarı bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın Algoritma 2+AİMS için çok fazla değişmediği, Algoritma 3+AİMS'te ise azaldığı görülmektedir. İşlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği, Algoritma 3+AİMS'in Algoritma 2+AİMS'ten daha iyi olduğu görülmektedir.

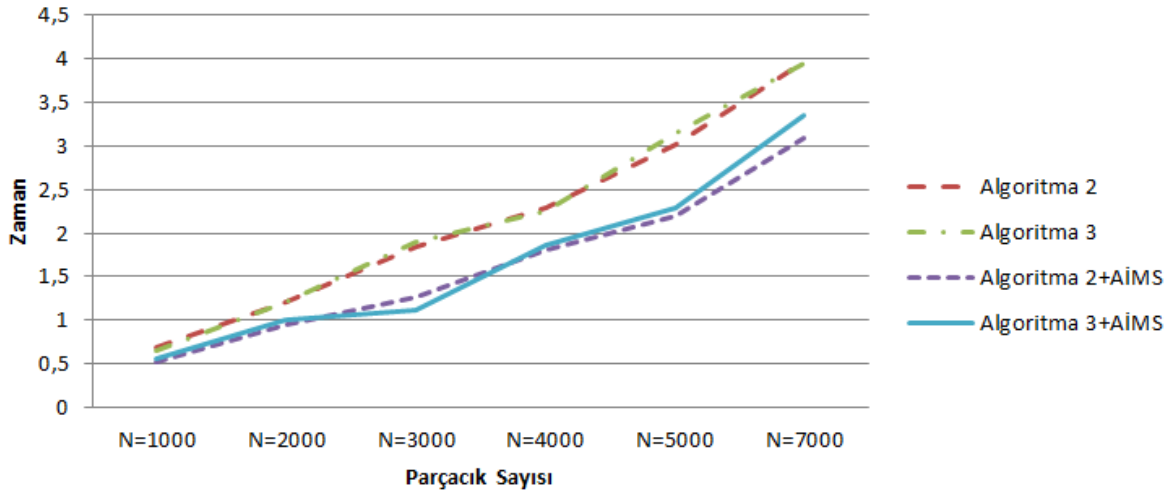
Senaryo 2 için Çizelge 6.11 incelendiğinde;

- Algoritma 2 ve Algoritma 2+AİMS için yapılan işlem süresi karşılaştırmalarında 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 39, 24, 15 ve 12 iyileşme sağlanmıştır. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.
- Algoritma 2 ve Algoritma 3 için yapılan işlem süresi ve hata karşılaştırmalarında (elde edilen sonuçların ortalaması alındığında) her iki algoritmanın birbirine yakın tepkiler gösterdiği görülmektedir.
- Algoritma 3 ve Algoritma 3+AİMS için yapılan işlem süresi karşılaştırmalarında 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 31, 22, 22 ve 18 iyileşme sağlanmıştır. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.
- Algoritma 2+AİMS ve Algoritma 3+AİMS için yapılan karşılaştırmalarda her iki algoritmanın da işlem süresi ve hata bakımından ise benzer tepki verdikleri görülmektedir.
- Algoritma 2 ve Algoritma 3+AİMS için yapılan işlem süresi karşılaştırmalarında Algoritma 3+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 32, 22, 19 ve 15 daha iyi olduğu görülmüştür. Fakat hata miktarlarında bakımından incelendiğinde Algoritma 2'nin 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 2, 2 ve 1 daha iyi olduğu, 0,01 gürültü seviyesi için ise yaklaşık yüzde 3 daha kötü olduğu görülmüştür.
- Algoritma 2+AİMS ve Algoritma 3 için yapılan işlem süresi karşılaştırmalarda Algoritma 2+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 66, 33, 21 ve 18 daha iyi olduğu görülmüştür. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.

Senaryo 2 için tüm parçacıklara ait 0,1 gürültü varyanslı benzetim sonuçlarına ait hata ve işlem süresi grafikleri sırasıyla Şekil 6.24 ve Şekil 6.25’de verilmiştir.



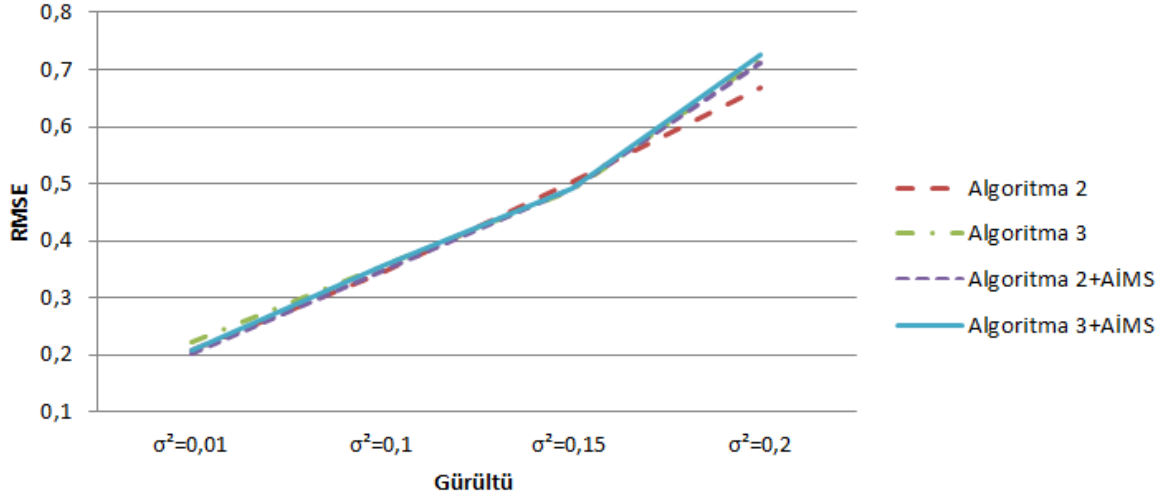
Şekil 6.24 Senaryo 2 Tüm Parçacıklar İçin Hata Grafiği (Dinamik ve $\sigma^2=0,1$)



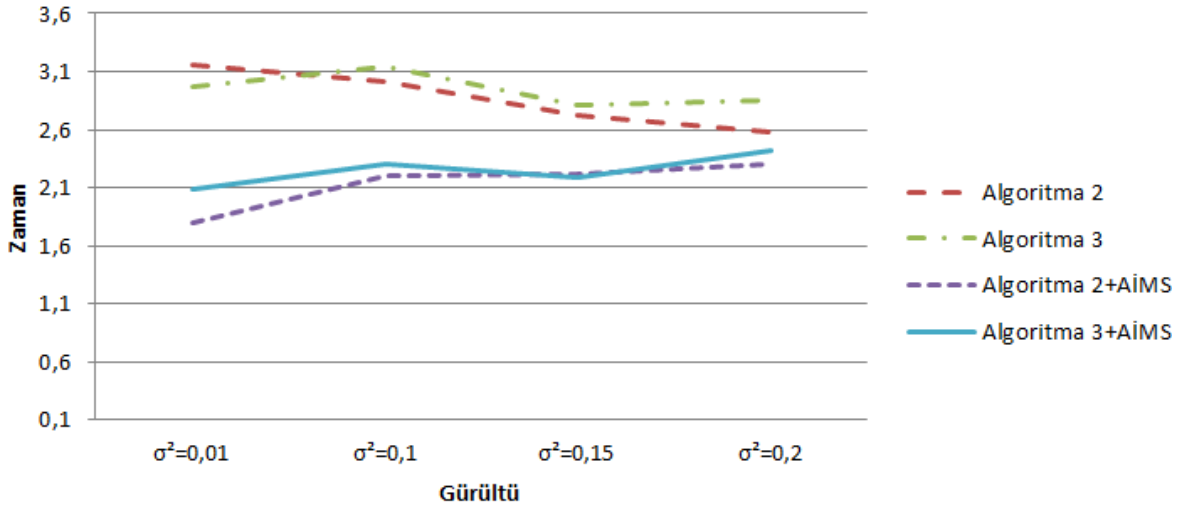
Şekil 6.25 Senaryo 2 Tüm Parçacıklar İçin Zaman Grafiği (Dinamik ve $\sigma^2=0,1$)

Şekil 6.24 ve Şekil 6.25 incelendiğinde Algoritma 2 ve Algoritma 3’ün işlem süresi ve hata bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın her iki algoritma için arttığı, işlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği ve benzer tepkiler verdiği görülmüştür.

Senaryo 2 için tüm gürültü varyansları ve 5000 parçacık için hata ve işlem süresi grafikleri sırasıyla Şekil 6.26 ve 6.27’de verilmiştir.



Şekil 6.26 Senaryo 2 Tüm Gürültüler İçin Hata Grafiği (Dinamik ve N=5000)



Şekil 6.27 Senaryo 2 Tüm Gürültüler İçin Zaman Grafiği (Dinamik ve N=5000)

Şekil 26 ve Şekil 27 incelendiğinde Algoritma 2 ve Algoritma 3'ün hata miktarı bakımından benzer tepkiler verdiği, işlem süresi bakımından ise Algoritma 2'nin Algoritma 3'ten daha iyi olduğu görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın her iki algoritma için çok fazla değişmediği

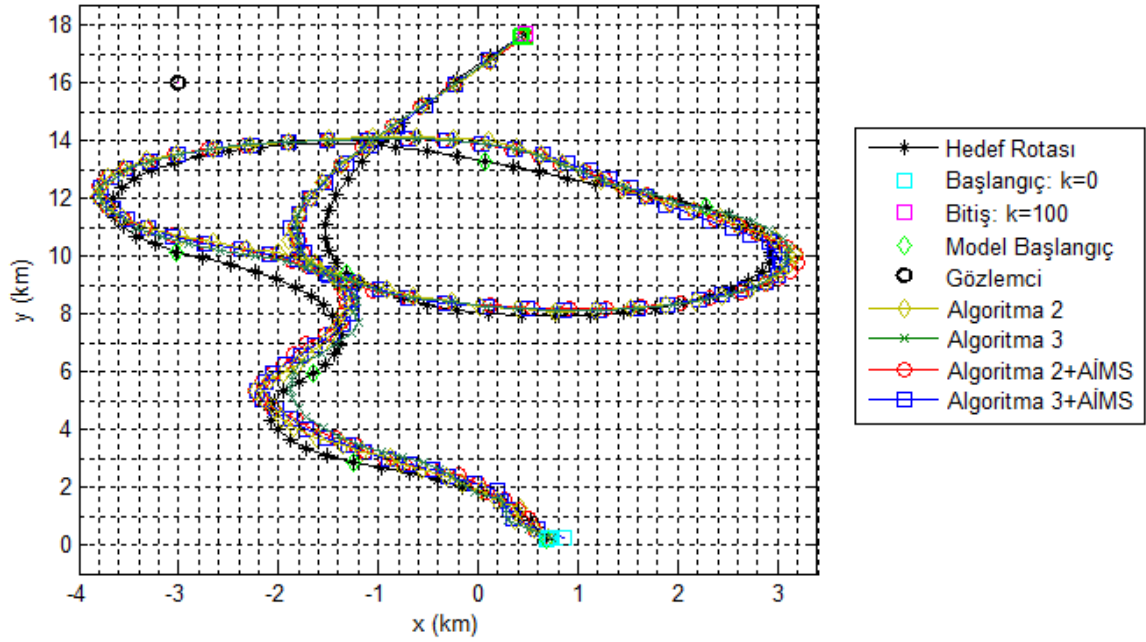
görülmektedir. İşlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği, Algoritma 2+AİMS'in Algoritma 3+AİMS'ten daha iyi olduğu görülmektedir.

Senaryo 3 için Çizelge 6.12 incelendiğinde;

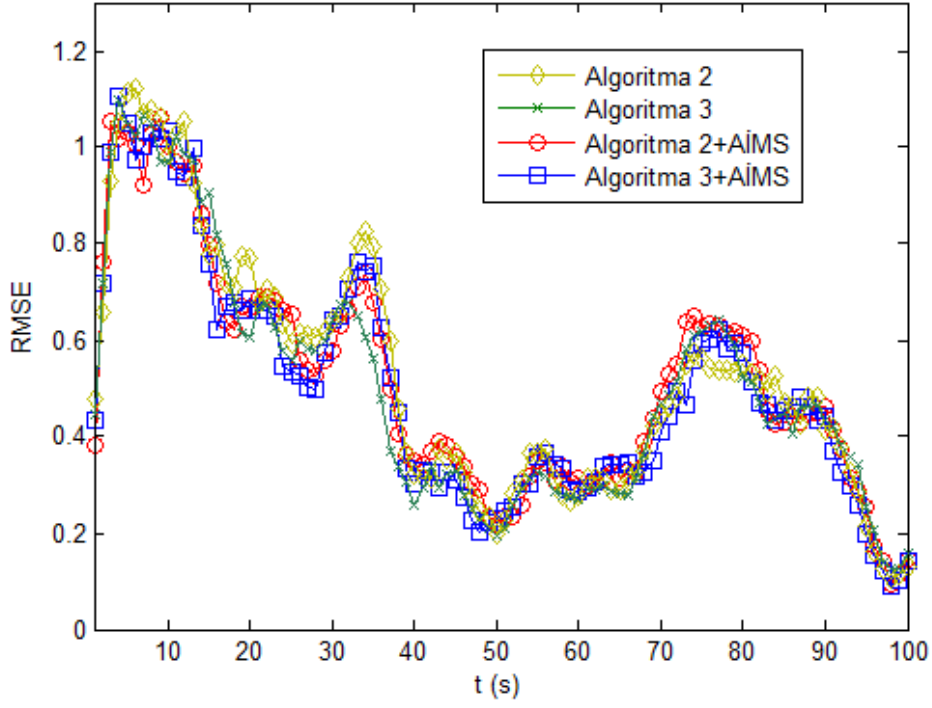
- Algoritma 2 ve Algoritma 2+AİMS için yapılan işlem süresi karşılaştırmalarında 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 23, 9, 10 ve 8 iyileşme sağlanmıştır. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.
- Algoritma 2 ve Algoritma 3 için yapılan işlem süresi ve hata karşılaştırmalarında (elde edilen sonuçların ortalaması alındığında) her iki algoritmanın birbirine yakın tepkiler gösterdiği görülmektedir.
- Algoritma 3 ve Algoritma 3+AİMS için yapılan karşılaştırmalarda 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için işlem süresi bakımından sırasıyla ortalama yüzde 26, 21, 14 ve 17 iyileşme sağlanmıştır. Tüm gürültü seviyeleri için hata miktarlarının aynı kaldığı görülmektedir.
- Algoritma 2+AİMS ve Algoritma 3+AİMS için yapılan karşılaştırmalarda 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için işlem süresi bakımından sırasıyla ortalama yüzde 2, 11, 6 ve 10 iyileşme sağlanmıştır. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.
- Algoritma 2 ve Algoritma 3+AİMS için yapılan işlem süresi karşılaştırmalarında Algoritma 3+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü seviyeleri için sırasıyla ortalama yüzde 25, 19, 16 ve 18 daha iyi olduğu görülmüştür. Fakat hata miktarlarında bakımından incelendiğinde Algoritma 2'nin 0,01, 0,1 ve 0,15 gürültü seviyeleri için hatanın Algoritma 3+AİMS'e göre yaklaşık yüzde 1 daha düşük olduğu, 0,2 gürültü seviyesi için ise Algoritma 3+AİMS'in yaklaşık yüzde 4 daha iyi olduğu görülmüştür.
- Algoritma 2+AİMS ve Algoritma 3 için yapılan işlem süresi karşılaştırmalarında Algoritma 2+AİMS'in 0,01, 0,1, 0,15 ve 0,2 gürültü

seviyeleri için sırasıyla ortalama yüzde 34, 13, 9 ve 8 daha iyi olduğu görülmüştür. Hata miktarları incelendiğinde ve hataların ortalaması alındığında ise hatanın aynı kaldığı görülmüştür.

Senaryo 3 için gerçekleştirilen MATLAB benzetimine ilişkin hedef takip ve hata grafikleri Şekil 6.28 ve 6.29'da sırasıyla verilmiştir. İlgili benzetim gözlem gürültüsü varyansı $\sigma^2=0,15$ ve parçacık sayısı 5000 için gerçekleştirilmiştir.

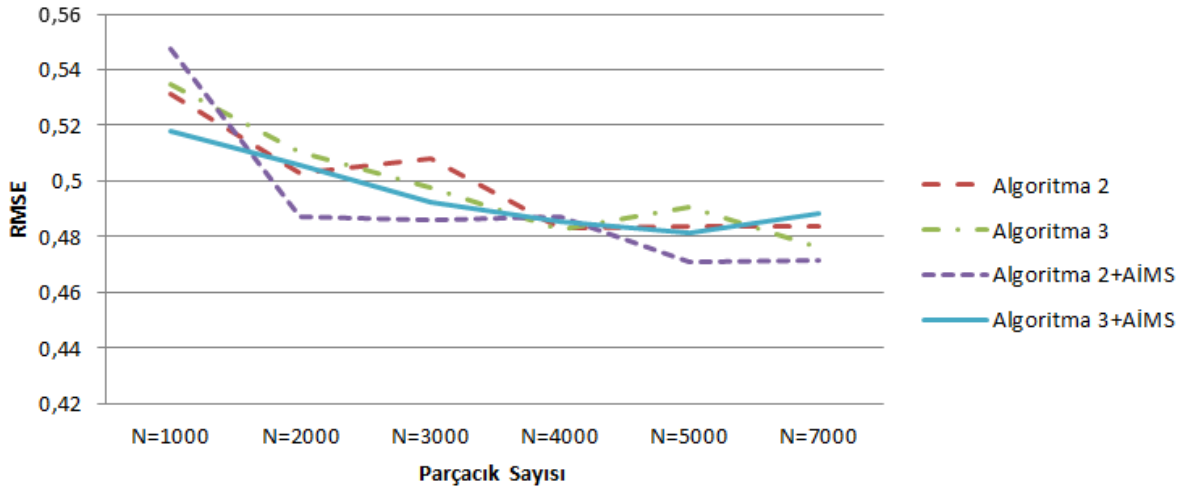


Şekil 6.28 Senaryo 3 Hedef Takibi (Dinamik, $\sigma^2=0,15$ ve $N=5000$)

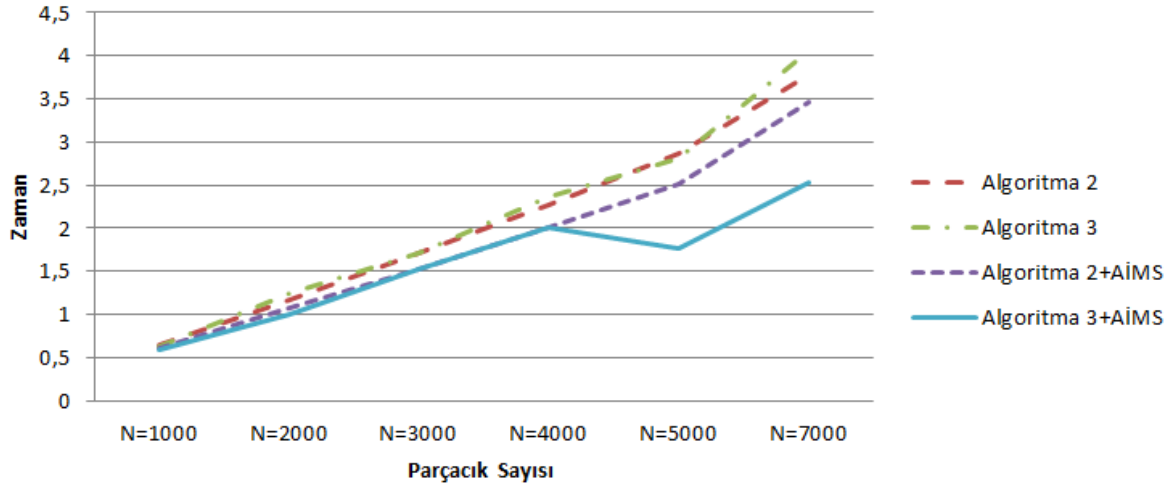


Şekil 6.29 Senaryo 3 RMSE Grafiği (Dinamik, $\sigma^2=0,15$ ve $N=5000$)

Senaryo 3 için tüm parçacıklara ait 0,1 gürültü varyanslı benzetim sonuçlarına ait hata ve işlem süresi grafikleri sırasıyla Şekil 6.30 ve Şekil 6.31’de verilmiştir.



Şekil 6.30 Senaryo 3 Tüm Parçacıklar İçin Hata Grafiği (Dinamik ve $\sigma^2=0,1$)

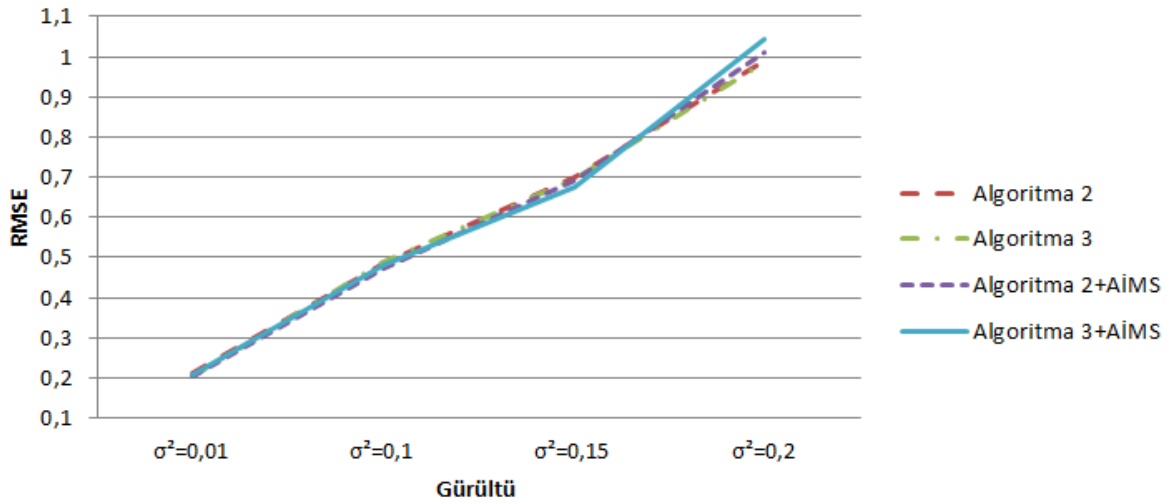


Şekil 6.31 Senaryo 3 Tüm Parçacıklar İçin Zaman Grafiği (Dinamik ve $\sigma^2=0,1$)

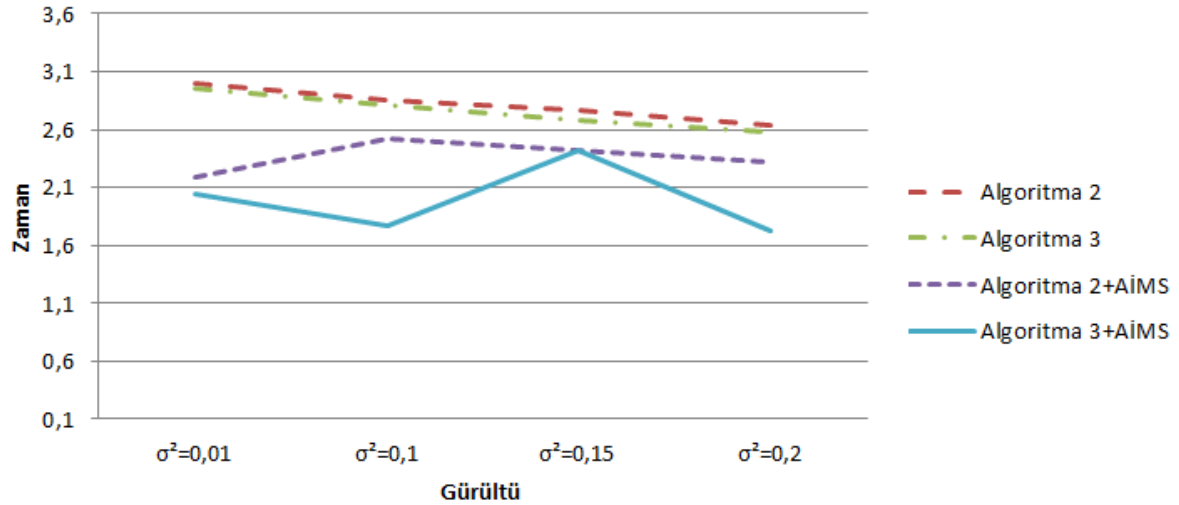
Şekil 6.28 ve Şekil 6.29'te yer alan grafikler Şekil 6.30 ve Şekil 6.31 yardımıyla 5000 parçacık için incelendiğinde RMSE bakımından Algoritma 2+AİMS'in, işlem süresi bakımından ise Algoritma 3+AİMS'in en iyi sonucu verdiği görülmektedir. AİMS algoritması uygulanan Algoritma 2+AİMS ve Algoritma 3+AİMS'in Algoritma 2 ve Algoritma 3'ten daha iyi RMSE ve işlem süresi değerine sahip olduğu görülmektedir.

Şekil 6.30 ve Şekil 6.31 incelendiğinde Algoritma 2 ve Algoritma 3'ün işlem süresi ve hata bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın her iki algoritma için azaldığı, işlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği görülmüştür. Algoritma 3+AİMS'in işlem süresi bakımından Algoritma 2+AİMS'ten daha iyi olduğu görülmektedir.

Senaryo 3 için tüm gürültü varyansları ve 5000 parçacık için hata ve işlem süresi grafikleri sırasıyla Şekil 6.32 ve 6.33'te verilmiştir.



Şekil 6.32 Senaryo 3 Tüm Gürültüler İçin Hata Grafiği (Dinamik ve N=5000)



Şekil 6.33 Senaryo 3 Tüm Gürültüler İçin Zaman Grafiği (Dinamik ve N=5000)

Şekil 32 ve Şekil 33 incelendiğinde Algoritma 2 ve Algoritma 3'ün hem işlem süresi hem de hata miktarı bakımından benzer tepkiler verdiği görülmektedir. Her iki algoritmaya da AİMS algoritması uygulandığında hatanın Algoritma 2+AİMS için çok fazla değişmediği, Algoritma 3+AİMS'te ise azaldığı görülmektedir. İşlem süresi bakımından ise her iki algoritmanın pozitif yönde etkilendiği, Algoritma 3+AİMS'in Algoritma 2+AİMS'ten daha iyi olduğu görülmektedir.

6.5.3. EÇM filtresi

Durum kestiriminde kullanılan ÇMPF'nin hata ve işlem süresi bakımından performansı bir diğer durum kestirim filtresi olan EÇM filtresi ile karşılaştırılmıştır. İki filtre arasındaki temel fark EÇM'nin Kalman tabanlı olması, ÇMPF'nin ise parçacık filtresi tabanlı olmasıdır.

Bu kısımda kullanılan EÇM filtresi Senaryo 1, Senaryo 2 ve Senaryo 3 için uygulanmış olup elde edilen sonuçlar Çizelge 6.13'te verilmiştir. Çizelge 6.13'te yer alan σ^2 gürültü varyansını, t saniye cinsinden zamanı, RMSE ortalama karesel hatayı ifade etmektedir.

Çizelge 6.13 EÇM Filtresi Sonuçları

	$\sigma^2=0,01$		$\sigma^2=0,1$		$\sigma^2=0,15$		$\sigma^2=0,2$	
	RMSE	t	RMSE	t	RMSE	t	RMSE	t
Senaryo 1	0,0591	10,4173	0,0996	10,4225	0,1441	10,5038	0,1868	10,4286
Senaryo 2	0,0544	13,8339	0,0959	13,8029	0,1429	13,7972	0,1863	13,9398
Senaryo 3	0,0568	10,6757	0,1057	10,6866	0,1483	10,6604	0,1926	10,7205

Çizelge 6.13'te yer alan değerler Çizelge 6.1, Çizelge 6.2, Çizelge 6.3, Çizelge 6.7, Çizelge 6.8, Çizelge 6.9 ile karşılaştırıldığında tüm senaryolar için işlem süresi açısından ÇMPF'ye ait hem Algoritma 2 ve Algoritma 3'ün hem de Algoritma 2+AİMS ve Algoritma 3+AİMS'in daha iyi sonuçlar verdiği görülmektedir. Fakat hata miktarları incelendiğinde E.Ç.M. filtresinin daha iyi olduğu görülmüştür.

7. SONUÇ VE ÖNERİLER

Bu tezde hareketli bir hedef, radardan elde edilen gözlem bilgilerine dayanılarak; sabit hız, sabit ivme ve koordineli dönüş modelleri kullanılarak oluşturulan çeşitli senaryolar ile ÇMPF kullanılarak takip edilmiştir. Tezin temel amacı kullanılan ÇMPF'nin hata miktarında olumsuz bir etki yaratmadan, ÇMPF'yi hızlandırmaktır. Bu amaçla ağırlıklandırılmış istatistiksel model seçimi (AİMS) adı verilen yeni bir yaklaşım geliştirilerek bu yaklaşımın, bir tanesi yine bu çalışmada oluşturulan yeni bir ÇMPF algoritması olmak üzere, iki ÇMPF algoritması üzerindeki etkisi araştırılmıştır. Yapılan çalışmalarda, algoritmaların gürültüye olan dayanıklılığı varyansı 0,01, 0,1, 0,15 ve 0,2 olan dört farklı gözlem gürültüsü üzerinde test edilmiştir. Radarın hedef takip başarısını inceleyebilmek için üç farklı senaryo oluşturulmuş ve tüm karşılaştırmalar bu üç senaryo üzerinde gerçekleştirilmiştir.

Bu çalışmada önerilen AİMS yaklaşımı, ÇMPF'lerde en son kullanılan P adet modeli kapsayan bir pencere üzerinde, model olasılıklarının ağırlıklandırılması ile en uygun modeli seçer. AİMS algoritması ile her zaman adımında sistemde tanımlanmış bütün modellerin hesaplanması yerine, sadece bir model hesaplanmaya çalışılmaktadır. Bu da işlem zamanının kısılmasına ve zamanın etkin kullanılmasını sağlamaktadır.

AİMS algoritması sadece model seçimi yapabilme olanağı sağlayabilen ÇMPF algoritmalarında çalışabilmektedir. Bu nedenle AİMS'nin etkisi Hong, Shi ve Chen [6] tarafından ortaya konan Algoritma 2 üzerinde ve bu çalışmada oluşturulan yeni bir ÇMPF algoritması (Algoritma 3) üzerinde denenerek, işlem süresi ve tahmin hatası parametreleri kullanılarak değerlendirilmiştir. Algoritma 1 parçacık seçimi üzerine kurulu bir algoritmadır ve model seçim yaklaşımı içermemektedir. Bu nedenle AİMS algoritması Algoritma 1 üzerinde uygulanamamıştır. Ancak yine de Algoritma 1, bu çalışmada üretilen Algoritma 3'ün performansını değerlendirmek amacıyla kullanılmıştır.

AİMS yaklaşımı olmaksızın Algoritma 1, Algoritma 2 ve Algoritma 3'e ilişkin sonuçlar incelendiğinde Algoritma 1'in hata miktarının düşük varyanslı gürültülerde

diğer algoritmalara göre daha az olduđu, fakat gürültü varyansı arttırıldıkça hata miktarının diğer algoritmalarından daha çok arttığı tespit edilmiştir. Bu da Algoritma 1'in diğer algoritmalarla karşılaştırıldığında gürültüye karşı dayanıksız olduğunu göstermektedir. Algoritma 2 ve 3'ün ise tüm senaryo ve gürültü varyansları için benzer tepkiler verdiği görülmüştür.

AİMS algoritması Algoritma 2 ve Algoritma 3'e entegre edilerek, yarı dinamik ve dinamik olmak üzere iki yaklaşımla kullanılmıştır. AİMS yönteminin yarı dinamik yaklaşımla uygulandığı benzetimlerde gürültü varyans değerine ve kullanılan senaryonun zorluğuna göre hata miktarı ve işlem süresi değişkenlik göstermektedir. Yarı dinamik AİMS yönteminin uygulandığı Algoritma 2 ve Algoritma 3'te Senaryo 1 ve Senaryo 2 için birbirine benzer etkiler görülmüş olup; 0,01, 0,1, 0,15 ve 0,2 gürültü varyansları için işlem süresinde sırasıyla; %41, %22, %17 ve %13 oranlarında azalma sağlanmıştır. Aynı gürültü değerleri için hata miktarında sırasıyla; %15, %1, %4 ve %8 oranında artış gözlenmiştir. Senaryo 3'ün diğer senaryolardan daha zor olması nedeniyle hata miktarı ve işlem süreleri diğer senaryolardan farklıdır. 0,01, 0,1, 0,15 ve 0,2 gürültü varyansları için sırasıyla %31, %14, %12 ve %10 oranlarında işlem süresinde azalma elde edilmiştir. Aynı değerler için hata miktarında belirgin bir artış veya düşüş gözlenmemiştir.

Algoritma 2 ve 3'e, dinamik yaklaşımla AİMS yöntemi uygulandığında; Senaryo 1 ve Senaryo 2 için yine birbirine benzer sonuçlar alınmış; 0,01, 0,1, 0,15 ve 0,2 gürültü varyansları için işlem süresinde sırasıyla %37, %21, %16 ve %10 oranlarında azalma sağlanmıştır. Aynı gürültü değerleri için hata miktarında sırasıyla %2, %0, %1 ve %3 oranında artış görülmüştür. Senaryo 3 ise hata miktarı ve işlem süreleri açısından diğer senaryolardan farklıdır. 0,01, 0,1, 0,15 ve 0,2 gürültü varyansları için sırasıyla %25, %15, %12 ve %12 oranlarında işlem süresinde azalma elde edilmiştir. Aynı değerler için hata miktarında belirgin bir artış veya düşüş gözlenmemiştir.

Elde edilen sonuçlar değerlendirildiğinde AİMS yaklaşımının işlem süresini azaltmadaki başarımı oldukça yüksektir. Algoritma 2 ve 3'ün AİMS ile birlikte

kullanımı bu algoritmaları %40'lara varan oranlarda daha hızlı hale getirmektedir. Bununla beraber tahmin hatasında görülen artış ise işlem süresinde elde edilen başarımlarına kıyasla oldukça düşük değerlerdir.

AİMS yönteminin uygulandığı algoritmalar kendi içinde değerlendirildiğinde ise hem işlem süresindeki azalma hem de hata miktarındaki değişim dikkate alındığında, Algoritma 3+AİMS algoritmasının diğerlerinden daha iyi sonuçlar verdiği görülmektedir.

ÇMPF'lerden elde edilen sonuçların bir başka çoklu model filtresi olan ve literatürde sıklıkla kullanılan EÇM filtresi ile karşılaştırması yapılmıştır. Fakat bu filtre bir Kalman filtresidir. Tüm senaryo ve gürültü varyansları için ÇMPF'ye ait algoritmalar EÇM filtresiyle karşılaştırıldığında ise hata miktarı bakımından EÇM'nin daha iyi olduğu görülmekle beraber, işlem süresi bakımından, özellikle AİMS yöntemini kullanan algoritmaların EÇM'ye göre daha hızlı işlem yaptığı görülmüştür.

Sonuç olarak yapılan tez çalışması ile sunulan AİMS algoritmasının işlem süresi üzerindeki etkinliği gösterilmiştir. Önerilen yaklaşım ve algoritmanın kullanımı, hareketli hedeflerin gerçek zamanlı takibinde tercih edilebilir. İleride yapılacak çalışmalarda iki boyuttan üç boyuta geçiş, AİMS algoritmasında yer alan pencereye ait eleman sayısı optimizasyonu ile AİMS algoritmasıyla birlikte kullanılan ve yine bu çalışmada önerilmiş olan ÇMPF'nin geliştirilmesi üzerinde çalışılabilir.

KAYNAKLAR LİSTESİ

- [1] <http://www.radartutorial.eu/index.tr.html>
- [2] Hammersley, J. M. and Morton, K. W., Poor Man's Monte Carlo, Journal of the Royal Statistical Society, Series B (Methodological), 16(1):23-38, 1954
- [3] Gordon N.J., Salmond D.J. and Smith A.F.M., Novel Approach to Nonlinear / Non-Gaussian Bayesian State Estimation. IEE-Proceedings-F, 140, 107-113, 1993
- [4] Berkin YILDIRIM, A Comparative Evaluation Of Conventional And Particle Filter Based Radar Target Tracking, METU, Kasım 2007
- [5] Berkin Yıldırım, Mübeccel Demirekler, Çoklu Model Parçacık Filtresi ve Etkileşimli Çoklu Model İKF'nin Manevra Yapan Hedef Takibinde Karşılaştırması, IEEE 978-1-4244-1999-9/08, 2008
- [6] Shaohua Hong, Zhiguo Shi, Kangsheng Chen, Easy-hardware-implementation MMPF for Maneuvering Target Tracking: Algorithm and Architecture, J Sign Process Syst 61:259–269, 2010
- [7] R. Karlsson, and N. Bergman, "Auxiliary Particle Filters for Tracking a Maneuvering Target", Proceedings of the 39.th IEEE Conference on Decision and Control, pages 3891–3895, Sydney, Australia, Aralık 2000
- [8] R. Karlsson, "Simulation Based Methods for Target Tracking", M.Sc. Thesis, Linköpings Universitet, Sweden, 2002
- [9] Muhammed Fatih Talu, Nesne Takip Yöntemlerinin Sınıflandırılması, Fırat Üniversitesi
- [10] Miodrag Bolic, Theory and Implementation of Particle Filters, School of Information Technology and Engineering University of Ottawa, 2004
- [11] Barış Kavcar, Simülasyon Yöntemi Kullanılarak Yapılan Satış Tahminleriyle Satış Bütçesi Hazırlanması, 2004
- [12] Greg Welch and Gary Bishop, An Introduction to the Kalman Filter, UNC-Chapel Hill, TR 95-041, July 24, 2006
- [13] Paul Sundvall, An Introduction to Particle Filtering, Kasım 2004
- [14] B. Ristic, M. S. Arulampalam, N. Gordon, Beyond the Kalman Filter, Particle Filters For Tracking Applications, Artech House, 2004
- [15] Miodrag Bolic, Architectures For Efficient Implementation of Particle Filters, Stony Brook University, Ağustos 2004

- [16] Gökhan SOYSAL, Multistatik Hedef Takibi Başarım Analizinde Gözlenen Bilgi Matrisi Kullanımı, Ankara Üniversitesi, 2012
- [17] Jeroen D. Hol, Resampling In Particle Filters, LiTH-ISY-EX-ET-0283-2004, 2004
- [18] Kevin Smith, Bayesian Methods for Visual Multi-Object Tracking with Applications to Human Activity Recognition, IDIAP Research Institute Swiss Federal Institute of Technology Lausanne, Switzerland, Şubat 2007
- [19] Muhammad-Kassiem Jacobs, Assimilation Of Lagrangian Data Into Idealized Models Of The Ocean Mesoscale Using Ensemble-Based Methods, Dalhousie University, Ağustos 2010
- [20] Roman Heimhuber, Efficient History Matching for Reduced Reservoir Models with PCE-based Bootstrap Filters, University of Stuttgart, Ağustos 2012
- [21] Lisa Turner, Christopher Sherlock, An Introduction to Particle Filtering, Mayıs 2013
- [22] Ahmed S.A. Alostaz, Optimized Fuzzy Tracking of A Moving Object With A Robotic Eye System, 2014
- [23] J. V. Candy, Bayesian Signal Processing, Classical, Modern And Particle Filtering Methods vol. 54, John Wiley&Sons, 2011
- [24] X. RongLi and Vesselin P. Jilkov, A Survey of Maneuvering Target Tracking: Dynamic Models, Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, Orlando, FL, USA, April 2000 (4048-22)
- [25] Joana Barbosa, Bastos Gomes, An Overview on Target Tracking Using Multiple Model Methods, Instituto Superior Tecnico Universidade Tecnica de Lisboa, Eylül 2008
- [26] Robert A. Singer, Estimating Optimal Tracking Filter Performance For Manned Maneuvering Targets, IEEE Transactions on Aerospace and Electronic Systems, 6(4):473-483, July 1970
- [27] Alper PASHA, Hava Trafik Kontrolü Benzetiminde Etkileşimli Çoklu Model (Interacting Multiple Model-IMM) Kestirim Performansı Ve Kalman Filtresi İle Karşılaştırılması, Havacılık Ve Uzay Teknolojileri Dergisi Temmuz 2008 Cilt 3 Sayı 4 (25-36), 2008
- [28] Seham Mouawad Aly, Raafat El Fouly, Hoda Baraka, Extended Kalman Filtering and Interacting Multiple Model for Tracking Maneuvering Targets in Sensor Networks, 13th International Conference on Aerospace Sciences & Aviation Technology, ASAT- 13, May 26 – 28, 2009

- [29] Anthony F. Genovese, The Interacting Multiple Model Algorithm for Accurate State Estimation of Maneuvering Targets, Johns Hopkins Apl Technical Digest, Volume 22, Number 4, 2001
- [30] Burak Manavoğlu, İki Boyutta Hedef İzleme Simülasyonu ve Modellenmesi, Hacettepe Üniversitesi, 2014
- [31] Recep Serdar Acar, Tracking Short-Range Ballistic Targets, METU, Eylül 2011
- [32] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, Per-Johan Nordlund, Particle Filters for Positioning, Navigation and Tracking, IEEE Transactions On Signal Processing, Vol. 50, No. 2 (425-437), February 2002
- [33] Xiaojun Yang and Keyi Xing, Joint State and Parameter Estimation in Particle Filtering and Stochastic Optimization, Stochastic Optimization—Seeing the Optimal for the Uncertain, Dr. Ioannis Dritsas (Ed.), ISBN: 978-953-307-829-8, 2011
- [34] Ghasem Saeidi, M. R. Moniri, Bearings-Only Tracking of Manoeuvring Targets Using Multiple Model Variable Rate Particle Filter with Differential Evolution, Asia Pacific Journal of Energy and Environment, Volume 1, No 3, 2014
- [35] Ihor Smal, Katharina Draegestein, Niels Galjart, Wiro Niessen and Erik Meijering, Particle Filtering for Multiple Object Tracking in Dynamic Fluorescence Microscopy Images: Application to Microtubule Growth Analysis, IEEE Transactions On Medical Imaging, Vol. 27, No. 6, June 2008