



EGE ÜNİVERSİTESİ

DOKTORA TEZİ

TELSİZ DUYARGA AĞLARINDA HEDEF TAKİBİ

Ayşegül ALAYBEYOĞLU

Tez Danışmanı : Doç. Dr. Aylin KANTARCI

İkinci Danışmanı : Prof. Dr. Kayhan ERCİYEŞ

Bilgisayar Mühendisliği Anabilim Dalı

Bilim Dalı Kodu : 619.01.00

Sunuş Tarihi : 16 Aralık 2009

Bornova-İZMİR

2009

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(DOKTORA TEZİ)

TELSİZ DUYARGA AĞLARINDA HEDEF TAKİBİ

Ayşegül ALAYBEYOĞLU

Tez Danışmanı : Doç. Dr. Aylin KANTARCI

İkinci Danışmanı : Prof. Dr. Kayhan ERCİYEŞ

Bilgisayar Mühendisliği Anabilim Dalı

Bilim Dalı Kodu : 619.01.00

Sunuş Tarihi : 16 Aralık 2009

Bornova-İZMİR

2009

Ayşegül ALAYBEYOĞLU tarafından **DOKTORA** tezi olarak sunulan “**TELSİZ DUYARGA AĞLARINDA HEDEF TAKİBİ**” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve **16 Aralık 2009** tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:

İmza

Jüri Başkanı	: Doç Dr. Aylin KANTARCI
Raportör Üye	: Prof. Dr. Kayhan ERCİYEŞ
Üye	: Prof. Dr. Aydın ÖZTÜRK
Üye	: Yrd. Doç. Dr. Aybars UĞUR
Üye	: Yrd. Doç. Dr. Şebnem BORA

ÖZET

TELSİZ DUYARGA AĞLARINDA HEDEF TAKİBİ

ALAYBEYOĞLU, Ayşegül

Doktora Tezi, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Aylin Kantarcı

Tez Eş Yöneticisi: Prof. Dr. Kayhan Erciyeş

16 Aralık 2009, 110 Sayfa

Hedef takibi uygulamaları, kablosuz duyarga ağlarının en yaygın uygulama alanlarından birisidir. Literatürde birçok hedef takibi algoritmaları mevcuttur ve bu çalışmalarda, t anındaki lider/kök düğüm, hedefin sadece $t+1$ anındaki konumunu tahminler. Bu yaklaşımlarda, hedefin hızlı gitmesi durumunda hedef, düğümler tarafından algılanmadan hızlı bir şekilde ilerleyebilir. Sonuç olarak bu çalışmalarda, hedef hızını arttırdıkça, hedefin kaybedilmesi olasılığı da artmaktadır. Tez kapsamında önermiş olduğumuz yaklaşım ile, hedefi kaybetme riskini minimuma indirerek, hedefin kaybedilmesi durumunda yeniden yakalamak için kullanılan düzeltme algoritmalarına olan ihtiyacı azaltmak hedeflenmiştir. Hedefi kaybetme riskini minimuma indirmek için de hedefin tahminlenen yönü doğrultusundaki düğümler uyandırılarak, bu yönde belirlenen sayıda küme, kapsama ağacı gibi yapıların hedef henüz gelmeden oluşturulması önerilmiştir. Ayrıca düzeltme mekanizmalarının kullanım sıklığının azalmasıyla, enerji tüketiminde de azalma olmakta ve bu da ağın genel yaşam süresini arttırmaktadır. Bu tezin katkıları; önermiş olduğumuz yaklaşımların literatürdeki genel yaklaşımlar ile karşılaştırılması, şu ana kadar literatürde karşılaşmadığımız hızlarda hareket eden hedefin başarıyla takip edilmesi, çok hızlı hareket eden hedefin takibi için yeni dört farklı dağıtık hedef takip algoritmasının önerilmesi, sadece doğrusal hareket eden değil, doğrusal olmayan hareketler yapan hedefin takibi de yapılarak, Parçacık Filtresi ve Genişletilmiş Kalman Filtresi gibi iki yaygın tahminleme algoritmalarının karşılaştırılması ve son olarak da düzeltme mekanizmasına olan ihtiyacın azaltılmasıyla enerji tüketiminden kazanç sağlanması olarak özetlenebilir.

Anahtar sözcükler: Hedef Takibi, Kablosuz Duyarga Ağları, Kümeleme, Parçacık Filtresi, Genişletilmiş Kalman Filtresi.

ABSTRACT**TARGET TRACKING IN WIRELESS SENSOR NETWORKS**

ALAYBEYOĞLU, Ayşegül

PHD in Computer Engineering Department

Supervisor: Assoc. Prof. Dr. Aylin KANTARCI

Co-Advisor: Prof. Dr. Kayhan ERCİYES

16'th December 2009, 110 pages

Target tracking applications are among the most common applications of wireless sensor networks. There are many tracking algorithms in the literature in which the current leader/root node at time t predicts the location only for time $t+1$. In case the target moves at high speed, it can pass by a group of nodes very fast without being detected. Therefore, in these studies, as the target increases its speed, the probability of missing that target also increases. In this thesis, instead of using recovery mechanisms frequently to detect the target again, we aim to find a way to decrease the target misses. In order to do this, we propose to wake up nodes in the predicted trajectory and form a number of clusters or spanning trees before the target comes. As the ratio of target missing decreases, the usage of recovery mechanisms also decreases leading to less energy consumption and higher network life time. The contributions of this thesis can be summarized as providing a comparison of existing target tracking approaches with the proposed approaches, tracking the targets moving at very high speed that we haven't met in the literature yet, proposing and implementing four new different distributed tracking algorithms, tracking not only the target with linear movement but also tracking the target with nonlinear movement, comparing the prediction accuracy of two favorite prediction algorithms Particle Filtering and Extended Kalman Filtering and lastly prolonging the network lifetime by decreasing the usage of recovery mechanism.

Keywords: Target Tracking, Wireless Sensor Networks, Clustering, Particle Filters, Extended Kalman Filters.

TEŞEKKÜR

Tez çalışması süresince değerli fikirlerini hiçbir zaman esirgemeyen, sabır ve hoşgörülerini ile beni destekleyen danışmanlarım Sayın Doç. Dr. Aylin KANTARCI'ya ve Sayın Prof. Dr. Kayhan ERCİYEŞ'e çok teşekkür ederim.

Tez izleme toplantılarında jüri olarak görev alan Sayın Prof. Dr. Aydın ÖZTÜRK ve Sayın Yrd. Doç. Dr. Aybars UĞUR'a değerli önerilerinden dolayı teşekkürlerimi sunarım.

Bilgisayar Mühendisliği bölümünü, hafta tatilleri ve bayram tatilleri dahil 24 saat açık tutarak istediğimiz kadar çalışma ortamı sağlayan önceki ve yeni bölüm başkanlarımız Sayın Prof. Dr. Halil ŞENGONCA ve Sayın Prof. Dr. Ata ÖNAL'a teşvik ve desteklerinden dolayı çok teşekkür ederim.

Doktora süresi boyunca bir çok imkan sağlayan ÖYP programına vermiş oldukları destekten dolayı Ege Üniversitesi Rektörü Sayın Prof. Dr. Candeğer YILMAZ'a ve Celal Bayar Üniversitesi Rektörü Sayın Prof. Dr. Semra ÖNCÜ'ye teşekkürü bir borç bilirim.

Doktora tezimin tahminleme mekanizmaları bölümünde yardımcı olan Ege Üniversitesi Elektrik Elektronik Mühendisliği Bölümü Araştırma Görevlisi Kemal BÜYÜKKABASAKAL'a ve simülasyon ortamına alışmamda yardımcı olan İzmir Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümü Araştırma Görevlisi Orhan DAĞDEVİREN'e çok teşekkür ederim.

Doktora süresi boyunca vermiş oldukları Doktora Bursu ile sağladıkları maddi destekten dolayı TÜBİTAK'a teşekkürlerimi sunarım.

Son olarak, desteğini hiçbir zaman esirgemeyen ve her zaman sevgiyle yanımda olan aileme minnettar olduğumu belirtmek isterim.

İÇİNDEKİLER

Sayfa

ÖZET	v
ABSTRACT	vii
TEŞEKKÜR	ix
ŞEKİLLER DİZİNİ	xv
ÇİZELGELER DİZİNİ	xx
1.GİRİŞ	1
2.LİTERATÜRDE İLGİLİ ÇALIŞMALAR	8
2.1. Hedefin Konumunun Belirlenmesi	8
2.2. Hedefin İzlenmesi	9
2.3. Verinin Ana Düğüme İletilmesi	12
3.DAĞITIK AKIŞ TABANLI KAPSAMA AĞACI (DATKA) VE ROTA BELİRLEME	15
3.1 Dağıtık Akış Tabanlı Kapsama Ağacı	16
3.2 Veriyi Tek Bir Merkeze Toplama	18
3.3 Polinom Aradeğerleme Yöntemi	19
3.3.1 Doğrusal Aradeğerleme Yöntemi	21
3.3.2 Polinom Aradeğerleme	22

İÇİNDEKİLER (devam)

Sayfa

4. DOĞRUSAL HAREKETLER İÇİN DİNAMİK ÇOKLU KÜME TABANLI HEDEF İZLEME ALGORİTMASI (DÇKİA).....	25
4.1 DÇKİA'nın Tanımı.....	27
4.2 DÇKİA'nın Sonlu Durum Makinesi.....	29
4.3 DÇKİA Örnek Bir İşlem.....	36
4.4 Düzeltme Algoritması.....	38
4.5 DÇKİA'nın Mesaj Karmaşıklık Analizi.....	39
4.6 DÇKİA Simülasyon Sonuçları	40
5.DOĞRUSAL HAREKETLER İÇİN DİNAMİK ÇOK KÜMELİ KAPSAMA AĞACI TABANLI HEDEF İZLEME ALGORİTMASI (DÇKAİA).....	45
5.1 DÇKAİA'nın Tanımı.....	47
5.2 DÇKAİA'nın Sonlu Durum Makinesi.....	50
5.3 Örnek Bir İşlem	55
5.4 DÇKAİA'nın Mesaj Karmaşıklık Analizi	59
5.5 DÇKAİA Simülasyon Sonuçları	61
6.DOĞRUSAL HAREKETLER İÇİN DİNAMİK KONİ TABANLI HEDEF İZLEME ALGORİTMASI (DKOİA)	66
6.1 DKOİA'nın Tanımı	66
6.2 DKOİA'nın Sonlu Durum Makinesi	71

İÇİNDEKİLER (devam)

6.3 DKOİA Örnek Bir İşlem	76
6.4 DKOİA'nın Mesaj Karmaşıklık Analizi.....	79
6.5 DKOİA Simülasyon Sonuçları	80
7.DOĞRUSAL OLMAYAN HAREKETLER İÇİN DİNAMİK KONİ TABANLI HEDEF İZLEME ALGORİTMASI	85
7.1 Sistem Modeli.....	86
7.2 Genel Tahminleme ve Güncelleme Çerçevesi	88
7.3 Genişletilmiş Kalman Filtreleri	91
7.4 Parçacık Filtreleri	94
7.5 Simülasyon Sonuçları	96
7.6 Yan Çalışmalar	102
7.6.1 Dağıtık Yapay Sinir Ağı Modeli	102
7.6.2 Bezier Spline Yöntemi.....	104
8. SONUÇ VE TARTIŞMA	106
KAYNAKLAR.....	111
EKLER	118
EK 1 SİMÜLASYON ORTAMI.....	118
EK 2 DENEYLER HAKKINDA DETAYLI BİLGİ	120
ÖZGEÇMİŞ.....	122

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1. Literatürdeki Hedef İzleme Algoritmalarının Kategorize Edilmesi	13
3.1 Lider Düğümlerin Ana Düğüme Veri İletimi	15
3.2 DATKA için Kullanılan Sonlu Durum Makinesi.....	16
3.3 DATKA'nın Örnek Bir İşlem Üzerinde Gösterimi	18
3.4 Ana Düğüme Veri İletimi	19
3.5 Polinom Aradeğerleme Fonksiyonları.....	20
3.6 Doğrusal Aradeğerleme.....	21
3.7 Polinom Aradeğerleme	24
4.1 Genel Dinamik Küme Tabanlı Hedef İzleme Yaklaşımı (GDKIA).....	26
4.2 DÇKIA'nın Genel İşleyişi	28
4.3 DÇKIA'nın Sonlu Durum Makinesi	30
4.4 Hedefin İzleme Alanına Girmesi ve Düğümlerin Hedefi Algılaması ...	36
4.5 Önceden Kümelerin Oluşturulması	36
4.6 Hedefin Yön Değiştirmesi	37
4.7 Düzeltme Algoritması	38

ŞEKİLLER DİZİNİ (devam)

4.8	DÇKİA ile Hedef Takibi	42
4.9	DÇKİA ve GDKİA Algoritmalarının Hedefi Kaybetme Olasılığı için Karşılaştırılması.....	43
4.10	DÇKİA ve GDKİA Algoritmalarının Hedefi Kaybetme Olasılık Farkının Düğüm Derecesi Bazlı Karşılaştırılması.....	43
4.11	DÇKİA ve GDKİA Algoritmalarının Enerji Tüketimi için Karşılaştırılması.....	44
5.1	Genel Dinamik Kapsama Ağacı Tabanlı Hedef İzleme Yaklaşımı (GDKAIA).....	45
5.2	DÇKAİA'nın Genel İşleyişi	47
5.3	Dinamik Kök Değişimi ve Ağacın Yeniden Yapılandırılması.....	49
5.4	DÇKAİA'nın Sonlu Durum Makinesi.....	51
5.5	DÇKAİA'da Kök Düğümün Hedefin Yönünü Belirlemesi	55
5.6	DÇKAİA'da Önceden Ağaçların Oluşturulması.....	56
5.7	DÇKAİA'da Önceden Ağaçlar Hazırlanmış Durumda	56
5.8	DÇKAİA'da Dinamik Kök Değişimi	57
5.9	DÇKAİA'da Hedefin Yeni Yönünün Belirlenmesi	58
5.10	DÇKAİA'da Hedefin Yeni Yönü Doğrultusunda Ağaçların Oluşturulması	59

ŞEKİLLER DİZİNİ (devam)

5.11 DÇKAİA ile Hedef Takibi	63
5.12 GDKAİA, GDKİA, DÇKİA ve DÇKAİA Algoritmalarının Hedefi Kaybetme Olasılığı için Karşılaştırılması	64
5.13 GDKAİA, GDKİA, DÇKİA ve DÇKAİA Algoritmalarının Enerji Tüketimleri için Karşılaştırılması.....	64
6.1 İvme ve Açılı İlişkisi	67
6.2 DKOİA'nın Genel İşleyişi.....	68
6.3 Koni Çizme ve Koni İçinde Kalan Dğümleri Belirleme	69
6.4 DKOİA'nın Sonlu Durum Makinesi	72
6.5 Kök Dğümün Koni İçinde Kalan Dğümlerle Kapsama Ağaçları Oluşturması.....	76
6.6 DKOİA'nın Çalışması ile Elde Edilen Yapı	77
6.7 DKOİA'da Kök Dğümün Dinamik Değişimi.....	77
6.8 DKOİA'da Yeni Kök Dğümün Koni Oluşturması.....	78
6.9 DKOİA'da Yeni Kök Dğümün Koni içinde Kapsama Ağaçları Oluşturması.....	79
6.10 DKOİA ile Hedef Takibi	82
6.11 DKOİA , DÇKAİA,DÇKİA, GDKAİA ve GDKİA Algoritmalarının Hedefi Kaybetme Olasılığı için Karşılaştırılması.....	83
6.12 DKOİA , DÇKAİA,DÇKİA, GDKAİA ve GDKİA Algoritmalarının Enerji Tüketimleri için Karşılaştırılması	84

ŞEKİLLER DİZİNİ (devam)

7.1	Genişletilmiş Kalman Filtresi Algoritmasının Genel İşleyişi	94
7.2	Farklı Hareket Rotalarının PF_ DKOİA ile Takip Edilmesi	99
7.3	PF ile GKF'nin Hedef Takibinde Farklı Hareketler için Tahminleme Hataları	101
7.4	PF ile GKF'nin Hedef Takibinde Farklı Hareket Tipleri için Tahminleme Hataları	102
7.5	Hedef Takibinde Tahminleme için Tasarlanan YSA Modeli	103
7.6	Dağıtık YSA'nın Hedef Takibinde Kullanımı	103
7.7	Bezier Spline Yöntemi ile Noktalardan Geçen Eğrinin Bulunması	104
8.1	Doğrusal Olmayan Hareketler için Önerilen Bütün Hedef Takip Algoritmalarının Hedefi Kaybetme Oranlarının Karşılaştırılması	106

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
3.1 Ana Düğümde Bulunan Tablo	19
8.1 Doğrusal Hareketler için Algoritmaların Farklı Hız Değerleri için Karşılaştırılması.....	108
8.2 Doğrusal Olmayan Hareketler için Algoritmaların Karşılaştırılması.....	109
8.3 Tahminleme Mekanizmalarının Farklı Hareket Türleri için Karşılaştırılması	109

1. GİRİŞ

Telsiz iletişim ortamındaki gelişmeler düşük maliyetli, düşük güce sahip, çok fonksiyonlu ve kısa mesafede iletişim kurabilen duyarga (Ing. sensor) düğümlerinin gelişmesine imkan sağlamıştır. Bu küçük duyarga düğümleri algılama, veri işleme ve iletişim kurma yeteneklerine sahiptir. Birçok düğüm bir araya gelerek duyarga ağlarını (Ing. Sensor Networks) oluştururlar. Duyarga ağlarında bulunan duyarga düğümleri birbirleri ile işbirliği halindedirler ve yetenekleri doğrultusunda veri üzerinde işlem yapabilirler.

Duyarga ağlarının, ortamdaki sıcaklık, nem, basınç, ses, hareketlilik ve ışık gibi ortam koşullarındaki değişiklikleri takip edebilen sismik, termik, manyetik ve görsel gibi bir çok farklı tipte duyargalar içerebilmeleri, bu duyarga ağlarının bir çok farklı uygulama alanında kullanılabilmesini sağlar. Bu uygulama alanları askeri, çevre, sağlık, ev ve diğer ticari alanlar olmak üzere sınıflandırılabilir. Duyarga ağları, askeri alanda, özellikle savaş alanlarında mevcut donanım bilgisine ulaşmak, düşman askerinin hareketlerini izlemek ve savaş hasarı ile ilgili bilgi toplamak için, çevresel uygulamalarda hayvanların hareketlerini izlemek, kimyasal ve biyolojik tespitlerde bulunmak, orman yangınlarını ve sel felaketlerini tespit etmek gibi bir çok amaç için, sağlık uygulamalarında hasta takibi için kullanılabilir. Ev uygulamalarında elektrik süpürgesi, mikrodalga fırın gibi cihazların içine yerleştirilirken, ticari uygulamalarda binaların havalandırma ve ısıtma sistemlerinde veya araba hırsızlıklarının tespiti gibi uygulamalarda kullanılabilmektedir (Simon et al., 2004; Thorstensen et al., 2004; Zhang et al., 2004). Bu tez kapsamında telsiz duyarga ağlarında hedef takibi uygulama alanı üzerine çalışılmıştır.

Telsiz duyarga ağlarında düğümler arasındaki yönlendirme işlemi için en çok kullanılan tasarım yöntemlerinden birisi de Çizge teorisidir (Graph Theory). Çözüm aranan problem, çizge teorisi kullanılarak tanımlanabilirse, çizge teorisine ait olan özellikler ve çözüm yöntemleri problemin çözümü için faydalı olacaktır. Bir çizge, noktalar ve bu noktaları birleştiren kenarlardan oluşan çizgiler topluluğudur. Tez kapsamında, duyarga ağını modellemek için çizge teorisinden yararlanılmıştır. Duyarga ağındaki duyarga düğümleri, çizge teorisindeki noktalar ile, düğümler arasındaki iletim bağı da çizge teorisindeki çizgiler ile tanımlanmıştır. Bir dizi çizgi kümesi (E) ve nokta kümesi (V) kullanılarak, duyarga ağı $G(V,E)$ çizge modeli şeklinde belirtilmiştir.

Telsiz duyurga ağlarının merkezi olmayan, dağıtık yapısı, bu ortamda kullanılacak olan algoritmaların da dağıtık olmasını gerektirir. Buna göre bir düğüm mesaj aldığı anda, bir takım işlemleri gerçekleştirir ve elde ettiği sonuca göre komşu düğümlerine mesaj gönderir. Bir düğüm sadece kendi durumuyla ilgili bilgiye sahiptir. Eğer komşu düğümlerinin durumları hakkında bilgi edinmek istiyorsa onlara mesaj göndererek iletişim kurması gerekir. Düğümler arasındaki iletişim ile hedeflenen genel işlem gerçekleştirilebilir. Gerçekleştirilmek istenen işlem, ortamdaki bütün düğümlere dağıtıldığı için, duyurga ağlarında kullanılan algoritmalar dağıtık algoritmalar olarak tanımlanır (Wattenhofer, 2006).

Duyurgaların, düşük maliyetli, çok fonksiyonlu ve kısa mesafede iletişim kurabilme yeteneği, telsiz duyurga ağlarında bir çok uygulama için kullanılabilmesine imkan sağlamıştır. Özellikle de hedef takip uygulamaları son yıllarda bir çok araştırmacının dikkatini çekmiş ve bu konu üzerinde çalışmalar yapılmıştır. Telsiz duyurga ağlarında hedef takibi için bir çok farklı çalışma olmasına rağmen, bu çalışmaların hedef takibi için gerçekleştirmiş oldukları ortak aşamalar vardır. Bu aşamalar, hedefi algılama, hedefin konumunu hesaplama ve hedefi izleme şeklinde özetlenebilir.

Algılama aşamasında, hedef izleme alanına girdiğinde, hedefe yakın ve uyanık durumda olan düğümler tarafından hedef algılanır. Bir duyurganın hedefi algılaması demek, duyurganın hedeften elde ettiği sinyal gücü değerinin belli bir eşik değerinin üzerinde olması demektir (Li et al., 2002). Hedefi algılamak için bazı çalışmalarda (Ren et al., 2005) bütün duyurga düğümleri sürekli uyanık durumda beklerken, bazı çalışmalarda ise (Gui and Mohapatra, 2004; Ren et al., 2005), enerji korunumu için düğümler periyodik olarak uyuma ve uyku durumlarına geçerler.

Konumlandırma aşamasında, hedefin varlığı tespit edildikten sonra, hedefi algılayan düğümler birbirleri ile işbirliğine geçerek hedefin konumunu bulmaya çalışırlar. Hedefin konumunu bulmak için üçaçılama (Ing. Triangulation), üçkenarlama (Ing. Trilateration) ve hücreleme (Ing. Voronoi) (Hightower and Borriello, 2003) hücreleri gibi bir çok konumlandırma yöntemi mevcuttur.

İzleme aşaması için, literatürdeki var olan teknikler küme tabanlı, kapsama ağacı tabanlı ve tahminleme tabanlı olmak üzere sınıflandırılabilirler. Özetle, küme tabanlı yaklaşımda, küme lideri, üyelerinden aldığı bilgiler ile hedefin konumunu hesaplar ve hedefi izler. Kapsama ağacı tabanlı yaklaşımda, bir kök düğüm ve bu

kök düğümünden n sıçrama mesafesindeki düğümlerle oluşturmuş olduğu ağaç bulunmaktadır. Kök düğüm ağacındaki düğümlerden elde ettiği veriler ile hedefi izler. Son olarak tahminleme tabanlı yaklaşımda, elde edilen verilerden hedefin bir sonraki konumu tahminlenir ve ilgili alandaki düğümler uyandırılarak hedefi algılamaları sağlanır.

Literatürdeki hedef izleme çalışmalarında genel olarak, t anındaki lider/kök düğüm hedefin sadece $t+1$ anındaki konumunu tahminlemektedir. Bu yaklaşımda, hedefin hızlı gitmesi durumunda, hedef, düğümler tarafından algılanmadan hızlı bir şekilde ilerleyebilir. Sonuç olarak bu çalışmalarda, hedef hızını artırdıkça, hedefi kaybetme olasılığı da artmaktadır. Hedefi kaybetme olasılığının yüksek olmasından dolayı, var olan bu çalışmalar, hedefin kaybedilmesi durumunda devreye girecek olan karmaşık düzeltme algoritmaları içerir. Tez kapsamında önermiş olduğumuz yaklaşım ile, hedefi yeniden yakalamak için karmaşık düzeltme algoritmaları kullanmak yerine, hedefi kaybetme riskini minimuma düşürmek hedeflenmiştir. Bunun için de hedefin tahminlenen yönü doğrultusundaki düğümler uyandırılarak, bu yönde küme, kapsama ağacı gibi yapılandırmaların gerçekleştirilmesi önerilmiştir.

Tezin literatüre yapmış olduğu katkılar özetlenecek olursa:

- Literatürde şu ana kadarki yapılan çalışmalarda 20 m/s civarındaki hızlarda ölçüm yapılmıştır. Bu çalışma ile 30m/s'den 100m/s'ye kadar henüz test edilmemiş hızlarda hareket eden hedefin takibi başarıyla gerçekleştirilmiştir.
- Çok hızlı hareket eden hedefin takibi için dört yeni dağıtık hedef takip algoritması önerilmiştir. Önermiş olduğumuz izleme algoritmalarında vurgulanan ve literatürde bulunmayan yeni fikir, hedefin belirlenen yönünde önceden çoklu küme ve kümelenendirilmiş kapsama ağaçlarını oluşturarak, hedefi kaybetme riskini minimuma indirmektir.
- Literatürdeki hedef takip algoritmaları küme tabanlı, kapsama ağacı tabanlı ve tahminleme tabanlı olmak üzere üç grupta sınıflandırılmıştı. Tez kapsamında her bir sınıfa girecek şekilde izleme algoritmaları önerilmiştir. Böylece küme, kapsama ve tahminleme tabanlı yaklaşımların karşılaştırılması gerçekleştirilebilmiştir.

- Önermiş olduğumuz her bir algoritma ile literatürdeki yaklaşımın, çok hızlı hareket eden hedef takibinde, hedefi kaybetme oranları için karşılaştırmaları yapılmıştır. Simülasyon sonuçlarından, önermiş olduğumuz yaklaşımın çok hızlı hareket eden hedefin izlenmesinde, literatürdeki yaklaşımlara göre çok daha iyi sonuç verdiği ve hedefi kaybetme riskini oldukça düşürdüğü görülmüştür.
- Önermiş olduğumuz algoritmalar ile hedefi kaybetme oranını düşürerek, hedefi tekrar bulmak için gerekli olan düzeltme (Ing. Recovery) mekanizmasına olan ihtiyacı da azaltmaktayız. Düzeltme mekanizması ne kadar az kullanılırsa, enerji tüketimi de azalmakta ve bu da ağır yaşam süresini arttırmaktadır.
- Tez kapsamında sadece doğrusal hareket eden hedefin değil, doğrusal olmayan hareketlerde bulunan hedefin takibi de gerçekleştirilmiştir. Doğrusal olmayan hareketlerde bulunan hedefin takibi için, tahminlemede çok yaygın iki yöntem olan Parçacık Filtreleri ve Genişletilmiş Kalman Filtreleri kullanılmıştır.
- Rasgele birçok doğrusal olmayan hareket modeli oluşturularak, Parçacık Filtrelerinin ve Genişletilmiş Kalman Filtrelerinin tahminlemedeki hata oranları karşılaştırılmıştır.
- Son olarak, bütün önermiş olduğumuz algoritmaların, doğrusal olmayan hareketler için, hedefi kaybetme oranları karşılaştırılmış ve genel bir sonuca varılmıştır.

Tez kapsamında, duyarga ağlarında hedef takibi için, kullanılmak üzere önermiş olduğumuz bütün algoritmalar dağıtık bir yapıya sahiptir ve algoritmaların tasarlanmasında çizge teorisinden yararlanılmıştır. Tezde, hedef takip sistemi için önermiş ve kullanmış olduğumuz algoritmalar ve genel kapsamaları:

- **Dağıtık Akış Tabanlı Kapsama Ağacı Algoritması (DATKA):**

Küme liderlerinin/köklerinin ana düğüme veri iletebilmeleri amacıyla oluşturulan kapsama ağacı algoritmasıdır.

- **Polinom Aradeğerleme (Ing. Polynomial Interpolation) Algoritması:**

Ana düğümün, lider düğümlerin kendisine göndermiş olduğu konum bilgilerinden, hedefin rotasını belirleyebilmesi için kullanılan algoritmadır.

- **Hedef Takibi Algoritmaları :**

Hedef takibi için her bir senaryoda küme lideri seçim ve konum bulma algoritmaları ortak olarak kullanılmıştır. Küme lideri seçimi için iki aşamalı lider seçim algoritması, hedefin konumunu hesaplamak için de Üçkenarlama (Ing. Trilateration) algoritması kullanılmıştır. Önermiş olduğumuz bütün algoritmalarda temel hedefimiz hedefi kaybetme riskini minimuma indirmek ve enerji kaybını düşürmektir. Hedef takibi için önermiş olduğumuz algoritmalar:

- Doğrusal Hareketler için Dinamik Çoklu Küme Tabanlı Hedef İzleme Algoritması (DÇKİA)**

Önermiş olduğumuz DÇKİA ile hedefin tahminlenen yönünde, belirlenen sayıda kümeler oluşturularak, hedef takibinin gerçekleştirilmesi hedeflenmiştir.

- Doğrusal Hareketler için Dinamik Çok Kümeli Kapsama Ağacı Tabanlı Hedef İzleme Algoritması (DÇKAİA)**

Önermiş olduğumuz DÇKAİA ile hedefin tahminlenen yönünde, belirlenen sayıda kümelendirilmiş kapsama ağaçları oluşturularak, hedef takibinin gerçekleştirilmesi hedeflenmiştir.

- Doğrusal Hareketler için Dinamik Koni Tabanlı Hedef İzleme Algoritması (DKOİA)**

Önermiş olduğumuz DÇKAİA ile hedefin tahminlenen yönü ve hızı ile ilişkili olarak bir koni oluşturulup, bu koni içinde belirlenen sayıda kümelendirilmiş kapsama ağaçları yapılandırılarak, hedef takibinin gerçekleştirilmesi sağlanmaktadır.

d. Doğrusal Olmayan Hareketler için Dinamik Koni Tabanlı Hedef İzleme Algoritması

Bu bölümde hedef takibi, doğrusal olmayan hareketler için gerçekleştirilmiştir. DÇKAİA algoritmasına, Parçacık Filtreleri(PF) ve Genişletilmiş Kalman Filtreleri (GKF) entegre edilerek tahminlemedeki hata oranları karşılaştırılmıştır.

i. PF_DKOİA

Parçacık filtreleri, hedefin bir sonraki konumunu tahminlemek için parçacıkların olasılıksal dağılımından yararlanan, olasılık tabanlı bir tahminleme algoritmasıdır. Parçacık filtrelerinin DKOİA algoritmasına entegre edilmesiyle, manevra yapan hedefin takibi hedeflenmiştir.

ii. GKF_DKOİA

GKF, hedefin hareket dinamiklerini ve ortamdan elde edilen ölçüm verilerini dikkate alarak hedefin bir sonraki durumunu belirleyen, Kalman filtreleri tabanlı bir tahminleme algoritmasıdır. Hedefin bir sonraki durumu ile ilgili tek bir kestirim yapar. GKF'nin DKOİA algoritmasına entegre edilmesiyle, manevra yapan hedefin takibi hedeflenmiştir.

• Düzenleme (Ing. Recovery) Algoritması

Bu algoritma hedef kaybedildiğinde, hedefin tekrar bulunabilmesi amacıyla kullanılmaktadır. Her bir hedef takip algoritmasında ortak olarak kullanılan bir mekanizmadır.

Tezin ilerleyen bölümleri şu şekilde devam etmektedir: 2. Bölümde, literatürde var olan hedef izleme algoritmaları incelenmiş ve hedef izlemenin her bir aşamasında, literatürdeki hedef izleme algoritmalarında kullanılan yaklaşımlar detaylı olarak açıklanmıştır. 3. Bölümde, kullanmış olduğumuz Akış Tabanlı Dağıtık Kapsama Ağacı Algoritması ile Polinom Aradeğerleme Algoritması detaylandırılmıştır. 4., 5. ve 6. Bölümlerde önermiş olduğumuz Dinamik Çoklu Küme Tabanlı, Dinamik Çok Kümeli Kapsama Ağacı Tabanlı ve Dinamik Koni Tabanlı Hedef İzleme Algoritmalarının tanımları, sahte kodları, sonlu durum makineleri, açık dağıtık kaynak kodları ve karmaşıklık analizleri verilmiş, düzeltme algoritmasından bahsedilmiştir. Her bir algoritmanın bir örnek üzerinde

alışması detaylı olarak açıklanmış ve ns-2 simülasyon ortamında uygulanarak, elde edilen sonuçlar grafikler ile yorumlanmıştır. 7. Bölümde, doğrusal olmayan hareket modelleri için Parçacık Filtreleri ve Genişletilmiş Kalman Filtreleri olmak üzere iki tahminleme yöntemi detaylı olarak açıklanmış ve Dinamik Koni Tabanlı Hedef İzleme Algoritmasına entegre edilmiş, ns-2 simülasyon ortamında uygulanarak, elde edilen sonuçlar grafiğe dökülmüş ve yorumlanmıştır. Son olarak, 8. Bölümde elde edilen bütün sonuçlar yorumlanarak karşılaştırılmıştır.

2. LİTERATÜRDE İLGİLİ ÇALIŞMALAR

Literatürdeki, telsiz duyarga ağlarında hedef izleme üzerine yapılan çalışmalar incelendiğinde, hepsinde ortak olarak izleme alanına giren hedef tespit edilir, konumu hesaplanır ve izleme işlemi gerçekleştirilir. Duyarga düğümünün elde ettiği sinyal gücü değeri belli bir eşik değerinin üzerine çıktığında, hedef algılanmış demektir. Hedefin algılanmasını takiben, hedefin konumunun bulunması ve hedefin izlenmesi için literatürde bir çok farklı yöntem kullanılmıştır. Ayrıca hedefin konumu hesaplandıktan sonra, konum bilgisi, lider düğümler tarafından ana düğüme iletilebilmektedir. Ana düğüme en etkin veri iletimi için de literatürde bir çok farklı yöntem kullanılmıştır. Aşağıdaki alt bölümlerde bu yöntemler detaylı olarak anlatılmıştır.

2.1 Hedefin Konumunun Belirlenmesi

Hedef izleme uygulamalarında düğüm konumlarının, GPS ya da diğer konumlandırma teknikleri ile bilindiği varsayılmaktadır. Hedefi algılayan düğümler seçmiş oldukları bir lider düğüme elde ettikleri veriyi gönderirler ve lider düğüm de en yüksek sinyal gücü değerine sahip üç duyarga düğümünün konumlarını kullanarak üçkenarlama ya da üçaçılama (Hightower and Borriello, 2001) gibi yöntemlerle hedefin konumunu tespit eder. üçkenarlama yönteminde en az üç düğümün hedefe olan uzaklık ve konum bilgisi kullanılarak hedefin konumu hesaplanırken, üçaçılama yönteminde düğümlerin hedefe olan yön bilgileri de dikkate alınmaktadır. Bu yöntemlerin yanında hedefin konumunun belirlenmesi için kullanılabilecek başka bir teknik de Voronoi diyagramlarının kullanılmasıdır. Voronoi tabanlı yaklaşımlarda, duyarga ağ alanı, duyargaların konumlarına göre Voronoi hücrelerine bölünecektir. Hedef hangi duyarganın Voronoi hücrelerine düşüyorsa, o duyarga hedefe en yakın duyarga olarak tespit edilmektedir. Hedefin konumunun daha net belirlenebilmesi için üç duyarga düğümünün konumunun kullanılması daha iyi sonuç verecektir. Bunun için (Tseng et al., 2003)'de ağ ortamı üç kere dinamik olarak Voronoi hücrelerine bölünmüştür. İlk adımda hedef hangi duyarganın hücrelerine düşerse o duyarga hedefe en yakın duyarga olarak tespit edilir. İkinci adımda seçilen duyarga dikkate alınmadan, alan Voronoi hücrelerine bölünür. Hedefin bulunduğu hücre hangi duyarga düğümüne aitse o düğüm hedefe en yakın ikinci düğüm olarak belirlenir. Benzer şekilde üçüncü adımda ilk iki adımda seçilen duyargalar dikkate alınmadan ağ ortamı Voronoi hücrelerine bölünür. Hedefin bulunduğu hücre hangi duyarga düğümüne aitse o düğüm hedefe en yakın üçüncü düğüm olarak

belirlenir ve bu üç duyarga düğümü arasında üçaçılama yöntemi uygulanarak hedefin net konumu tespit edilebilmektedir. (Wang et al., 2008; Zhang and Cao., 2004; Mechitov et al., 2003) yaklaşımlarında ise düğümlerin ikili algılama yaptıkları düşünülmüş ve hedefi algılayan düğümlerin konumlarının ortalaması alınarak hedefin konumu belirlenmiştir. (Ray et al., 2003)'te en yüksek sinyal gücüne sahip düğümün konumu hedefin konumu olarak kabul edilmiştir.

2.2 Hedefin İzlenmesi

Hedefin konumunun belirlenmesinden sonraki adım hedef hareketinin izlenmesidir. Hedef izleme algoritmaları genel olarak küme tabanlı, kapsama ağacı tabanlı ve tahminleme tabanlı olmak üzere üç sınıfa ayrılabilir. (Tseng et al., 2003; Yang et al., 2007; Yang and Sikdar, 2003; Chang and Hou, 2004; Olule et al., 2007; Brooks et al., 2003; Bulusu et al., 2001; Brooks et al., 2002; Jeong et al., 2007; Younis and Fahmy, 2004; Xu and Lee, 2007; Suganya, 2008; Goshorn et al, 2007; Wang et al., 2007)'de küme tabanlı yaklaşımlar önerilmiştir. Küme tabanlı yaklaşımların bazılarında kümeler ağın kurulumu aşamasında statik olarak yerleştirilirken, bazılarında ise hedef hareket ettikçe dinamik olarak oluşturulmaktadır. Statik olarak kümelerin yerleştirildiği yaklaşımlarda hedef hangi kümenin izleme alanına girdiyse o küme aktif hale gelmekte ve aktif küme lideri, kümesindeki düğümlerden elde ettiği bilgileri işleyerek hedefin gideceği yönü belirlemektedir. Hedefin tahmini gideceği yöndeki küme lideri uyandırılarak hedefin mevcut konum, hız ve yön bilgileri yeni küme liderine iletilir (Yang and Sikdar, 2003). Kümelerin, ağın kurulumu aşamasında statik olarak oluşturulmasının bir takım dezavantajları bulunmaktadır. Bu dezavantajlardan bir tanesi, herhangi bir küme liderinin çeşitli nedenlerden dolayı görevini yerine getirememesi durumunda kümesindeki düğümlerin kullanışsız hale gelmesidir. Benzer şekilde küme içindeki düğümlerin kullanışsız hale gelmesi durumunda küme liderinin görevini yerine getiremez. Bir başka önemli dezavantaj da farklı kümelerde bulunan düğümlerin birbirleri ile doğrudan iletişime geçip bilgi paylaşımında bulunamayışlarıdır. Hedefin hareket etmesiyle kümelerin dinamik olarak oluşturulduğu yaklaşımlarda ise küme liderleri önceden belirlenmediği için, küme lideri belirleme mekanizmasına gereksinim vardır. Bazı yaklaşımlarda en yüksek sinyal gücü değerine sahip olan düğüm küme lideri seçilirken (Chang and Hou, 2004; Olule et al., 2007; Goshorn et al, 2007), bazı yaklaşımlarda da iki aşamalı lider düğüm seçme mekanizması kullanılmaktadır (Yang et al., 2007). Bu mekanizmaya göre ilk aşamada küme içerisindeki komşu düğümler birbirlerine hedefe olan uzaklık ve kimlik bilgilerini iletirler. Eğer düğüme, hedefe

kendisinden daha yakın olan bir komşu düğümünden mesaj gelmezse, kendisini aday lider düğüm olarak belirler. Aksi halde hedefe en yakın olan komşu düğümünü ata düğüm (parent) olarak belirler. Kümedeki bütün düğümler birbirlerinden tek sıçramalık mesafede olmayacağı için ilk aşama sonucunda küme lideri olmak için bir kaç aday düğüm çıkacaktır. İkinci aşamada ise bu aday düğümler küme içindeki bütün düğümlere hedefe olan uzaklık ve kimlik bilgisini yayar. Herhangi bir aday düğüm, hedefe kendisinden daha yakın bir aday düğümden mesaj alırsa, aday düğüm olmaktan vazgeçer ve hedefe en yakın olan aday düğüm küme lideri olarak seçilir. (Chang and Hou, 2004; Brooks et al., 2003)'da lider düğümü seçmek için Voronoi hücreleri tabanlı bir yaklaşım kullanılmaktadır. Bu yaklaşımda Voronoi hücreleri, önceden belirlenmiş güçlü düğümler arasında oluşturulmakta ve hedef hangi güçlü düğümün hücresine düşüyorsa o düğüm küme lideri olarak belirlenmektedir. Bu yaklaşımlarda kümeleme işlemi Voronoi hücreleri kullanılarak gerçekleştirilmektedir. (Xu and Lee, 2007)'de ise küme liderleri, periyodik olarak, kalan enerji miktarlarına ve düğümün komşu düğümlerine olan yakınlığına göre seçilir.

Bazı yaklaşımlarda (Bulusu et al., 2001), küme, kullanıcının gönderdiği istek mesajı ile eşleşen türde veri algılayan düğümler ile oluşturulur. Eşleşmeyen düğümler ise bu isteği not alırlar ve bu istekle ilgili algılama yaptıklarında cevap mesajı gönderirler. (Jeong et al., 2007)'de iki aşamalı hedef takibi gerçekleştirilir. İlk aşamada, hedefi algılayan düğümler küme oluşturarak hedefle ilgili bilgi toplarlar. İkinci aşamada ise hedefin küme içindeki hareketi takip edilir. (Younis and Fahmy, 2004)'de hedefin takip edileceği alan belirlenirken merkezi hedefin tahminlenen konumu olan bir daire çizilip bu daire içindeki düğümlerin algılama yapması yerine, hedefin hız, yön, ivme gibi kinematik özellikleri dikkate alınarak uyandırılacak düğümler belirlenir. Bunun yapılmasının nedeni klasik yöntem ile daireler çizilerek oluşturulan kümede, hedef bazı alanlara belirli bir zaman dilimi içinde hiç uğramamaktadır. Bu nedenle, klasik yöntemlerde olduğu gibi daire içindeki bütün düğümlerin uyandırılması gereksiz görülmektedir. (Younis and Fahmy, 2004)'teki yöntemle benzer şekilde (Yang et al., 2006)'da da hedefin hareket yönünün ve hızının dinamik olarak değişeceği göz önünde bulundurularak hedef izleme algoritmaları önerilmiştir.

Küme tabanlı yaklaşımların önemli avantajları vardır. Bunların başında da gereksiz mesaj iletimindeki azalma gelir. Gereksiz iletimin azalmasıyla enerji tüketimi azalacak ve ağın yaşam süresi artacaktır. Düğümlerin mesajlarını sadece

küme liderlerine iletmeleri ile uzak mesafelere iletimin neden olacağı büyük enerji kayıpları engellenmiş olacaktır.

(Zhang et al., 2004)'de ağaç tabanlı hareket izleme üzerine bir çalışma önerilmiştir. Bu çalışmaya göre hedefi algılayan düğümler birbirleri ile iletişime geçerek kendilerine bir kök düğüm seçerler. Bu yaklaşımda kök düğüm seçme mekanizmasına gereksinim vardır. Kök düğüm seçmek için genellikle bir önceki bölümde bahsedilen iki aşamalı yöntem kullanılmaktadır. Kök düğüm, kapsama ağacındaki bütün düğümlerden veri almakta ve bu veriler üzerinde bir takım işlemler yaparak hedef ile ilgili bilgi sağlamaktadır. Kapsayan ağaç tabanlı yaklaşımlarda hedef hareket ettikçe ağaçtan çıkarılacak eski düğümler ve ağaca eklenecek yeni düğümler olacaktır. Kök düğüm ile hedef arasındaki uzaklık belirli bir değeri aştığından yeni kök düğüm belirlenerek ağaç yeniden yapılandırılacaktır. Bu yaklaşım, kök düğüme ağaçtaki bir çok düğümden veri gönderimi yapıldığı için hedefle ilgili daha doğru bilgi elde etmek açısından avantajlıdır.

Küme tabanlı ve ağaç tabanlı hareket izleme yaklaşımlarının yanında, (Wang et al., 2008; Mechitov et al., 2003) uygulamalarında duyarga düğümlerinin ikili algılama yaptığı düşünülmüştür. Hedefin konum tespiti, hedefi algılayan düğümlerin konumlarının ortalaması alınarak hesaplanmış ve belirli zaman aralıkları ile hedefin konumları hesaplanarak elde edilen koordinatlar doğrultusunda bir doğru çizilip hedefin gideceği yön tespit edilmiştir ve hedefi, sadece hedefin gideceği yönün etrafındaki düğümlerin izlemesi sağlanmıştır. (Nasipuri and Li, 2002)'de dikdörtgen bir yapıya sahip ağ ortamı düşünülmüş ve dikdörtgenin köşelerine konumu bilinen düğümler yerleştirilmiştir. Ortamdaki diğer düğümler bu dört konumu bilinen düğümden periyodik olarak sinyaller alırlar. Hedefin konumu, duyarga düğümlerinin, farklı konumu bilinen düğümlerden almış olduğu sinyal zamanına ve bu düğümlere olan açısız konumlarına dayalı olarak hesaplanır

Hedefin izlenmesi sırasında önemli bir işlem, hedefin yakın gelecekte ulaşacağı konumun belirlenmesidir. Genel olarak şimdiye kadar incelediğimiz (Yang and Sikdar, 2003; Olule et al., 2007; Wang et al., 2008; Mechitov et al., 2003) yaklaşımlarında, hedefin iki veya daha fazla zaman dilimlerinde elde edilen konumları ve bu konumlara ulaşmak için geçen süre bilgileri kullanılarak hedefin hız ve yön bilgilerine ulaşılmıştır. Bu bilgiler kullanılarak da herhangi bir zaman içinde hedefin hangi konumda bulunacağı tahmini bilgisine ulaşılmıştır.

Literatürdeki hedef izleme uygulamalarının çoğu, hedefin sabit hızla hareket ettiğini varsaymaktadır. Bu çalışmalar, hedef aniden durduğunda, dönüş ya da ivmeli bir hareket yaptığında hatalı tahminleme yapabilmektedir. (Gupta and Das, 2003)'te tahminleme başarımını arttırmak için Kalman Filtreleri, (Kuang et al., 2008)'de Parçacık Filtreleri kullanılmaktadır. (Xu and Li, 2003; Xu et al., 2004)'te hedefin bir sonraki konumunun belirlenmesi için hedefin geçmiş hareket bilgileri kullanılmıştır.

2.3 Verinin Ana Düğüme (Sink) İletilmesi

Literatürde, ortamdaki düğümler ile ana düğüm arasında veri iletimini gerçekleştirmek için geliştirilmiş bir çok protokol bulunmaktadır. Bu protokoller genel olarak veri merkezli, hiyerarşik ve lokasyon tabanlı duyarlı olmak üzere sınıflandırılabilir (Akkaya and Younis, 2005).

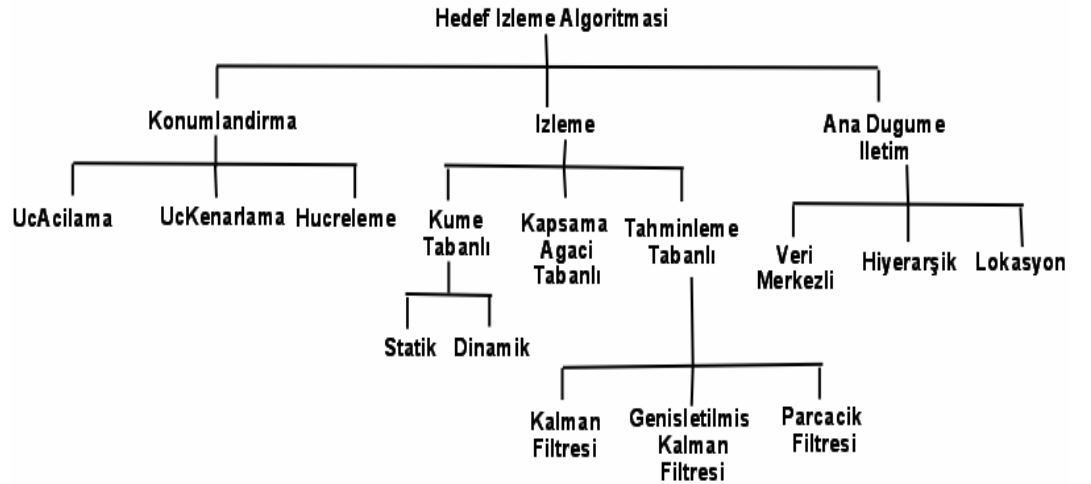
Veri merkezli yönlendirme protokollerine örnek olarak (Intanagonwiwat et al., 2000; Schurgers and Srivastava, 2001; Heinzelman et al., 1999) verilebilir. (Intanagonwiwat et al., 2000)'da hedefle ilgili bilgi almak isteyen ana düğüm ağa bir ilgi mesajı yayar. İlgi mesajını alan düğümler de, eğer algıladıkları veri bu ilgi mesajının içeriği ile eşleşiyorsa, ilgi mesajının geldiği yolun tersi yönünde cevap mesajı gönderir. Böylece ana düğüm ile duyarga düğümleri arasında yol oluşturulmuş olur. (Schurgers and Srivastava, 2001)'de ise ilgi mesajı ağa yayıldığında zıplama sayısı da tutulur. Böylece ağdaki her bir düğüm, ana düğüme olan minimum zıplama sayısını bulabilir ve verisini bu yol üzerinden ana düğüme iletir. (Heinzelman et al., 1999)'daki temel düşünce verinin üst veriler ile isimlendirilmesidir. Ana düğüm, verinin kendisini iletmeden önce ağa verinin içeriğini tanımlayan üst verileri yayar. Üst veri ile ilgilenen düğümler ana düğüme istek mesajını gönderir. İstek mesajını alan ana düğüm, ilgili düğümlere gerçek veriyi ulaştırır.

Hiyerarşik yönlendirme protokollerinde ise düğümler küme içlerine alınarak, ana düğüme gönderilecek mesaj sayısı azaltılmaya çalışılmaktadır. (Heinzelman et al., 2000; Lindsey and Raghavendra, 2002) yaklaşımları, hiyerarşik yönlendirme protokolleri için örnek olarak verilebilecek çalışmalardır. (Heinzelman et al., 2000)'daki temel düşünce, duyarga düğümlerinin elde ettikleri sinyal gücü değerine göre kümelenmesi ve her kümeye bir küme lideri atanarak, ana düğüme veriyi sadece bu küme liderlerinin göndermesini sağlamaktır. (Lindsey and Raghavendra, 2002)'de bir çok küme oluşturmak yerine düğümler zinciri

oluşturulması önerilmiştir. Bu yaklaşıma göre her bir düğüm, komşu düğümünden aldığı veriye kendi verilerini de ekleyerek bir sonraki komşusuna iletir. Zincirden sadece bir düğüm ana düğüme mesaj atmak için seçilir.

Lokasyon Tabanlı yönlendirme protokollerine (Rodoplu and Ming, 1999; Xu et al., 2001; Yu et al., 2001) çalışmaları örnek verilebilir. (Rodoplu and Ming, 1999)'da her bir düğüm için bir yayım alanı oluşturulur. Bu yayım alanında bir çok düğüm bulunmaktadır ve veriyi bu düğümler üzerinden iletmek, doğrudan iletmeye göre enerji açısından daha etkin bir çözümdür. Bu yaklaşımdaki temel düşünce, iki düğüm arasındaki iletim için az güce ihtiyaç duyan ve az düğüm barındıran bir alt ağı bulmaktır. Böylece ağdaki bütün düğümleri dikkate almaya gerek kalmadan, ana düğüme ulaşmak için minimum güce sahip yol bulunur. (Xu et al., 2001)'de ağ, ızgaralara bölünmüştür ve her bir ızgara için lider düğüm bulunur. Bu düğüm, diğer düğümlere veri iletimi için liderlik yapar. Fakat buradaki lider düğüm diğer hiyerarşik protokollerde olduğu gibi veri birleştirme işlemi yapmaz. (Yu et al., 2001)'de ana düğümün bütün ağa ilgi mesajı yayması yerine sadece belirli bir alana göndermesini sağlayarak enerjiden kazanç sağlanmaktadır.

Yukarıda bahsedilen yöntemleri, hedef izleme senaryoları için Şekil2.1'deki gibi kategorize edebiliriz.



Şekil2.1 Literatürdeki Hedef İzleme Algoritmalarının Kategorize Edilmesi

Yukarıda bahsetmiş olduğumuz çalışmalar, yavaş ya da normal hızlarda giden hedefin takibi için gerçekleştirilmiştir. Çok hızlı hareket eden hedefler dikkate alınmamıştır. Ayrıca var olan bu çalışmaların hiç birinde amaç, hedefi kaybetme riskini azaltmak olmamıştır. Bunun yerine hedefin kaybedilmesi durumunda, hedefin yeniden bulunması için düzeltme mekanizmaları geliştirilmiştir. Biz ise çalışmamızda, hedefi kaybetme riskini minimuma düşürerek bu mekanizmalara olan ihtiyacı azaltmayı, böylece enerji tüketiminden kazanç sağlayarak ağın yaşam süresini uzatmayı amaçlamaktayız. Bunun için de literatürde olmayan, çoklu kümeleme ve çoklu kümelenendirilmiş kapsama ağacı yaklaşımlarını önermekteyiz.

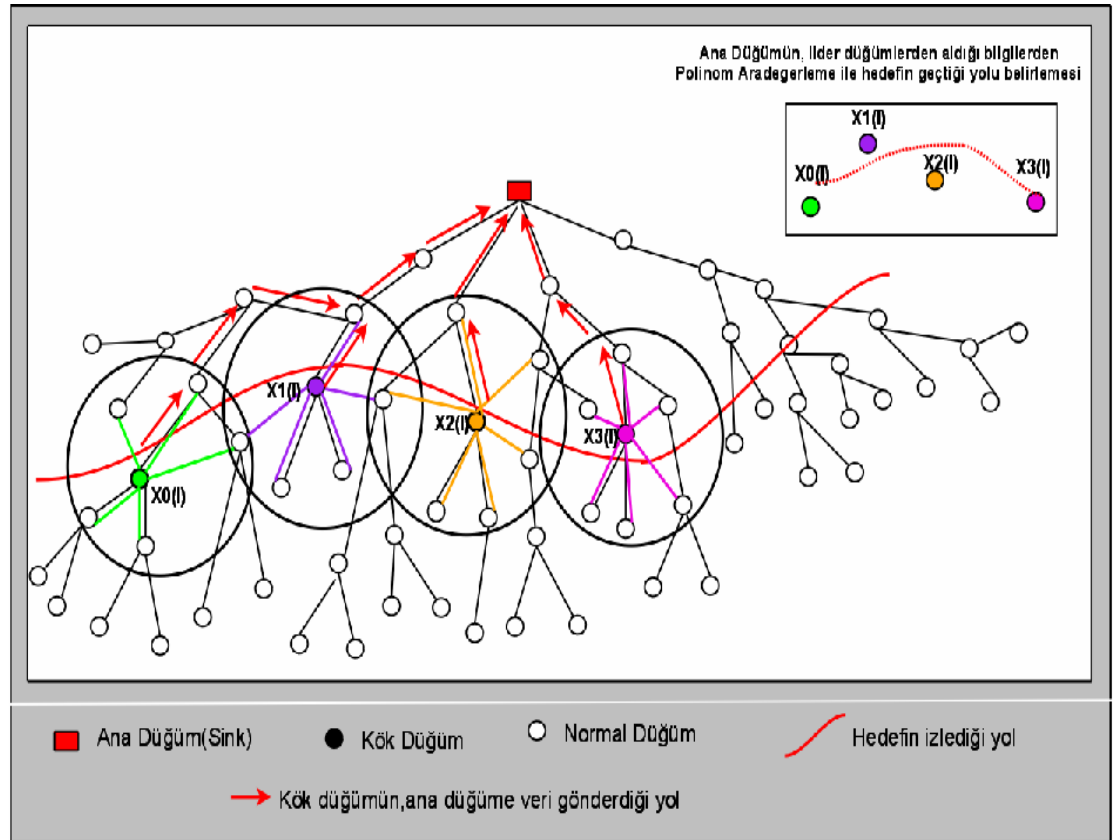
Literatürdeki hedef takip sistemleri incelendiğinde, hedefin t anındaki durumuna göre sadece $t+1$ anında bulunacağı konumunun tahminlendiğini ve bu tahmini konumda küme veya kapsama ağacı gibi yapılandırmalar gerçekleştirilerek hedef takibi sağlandığı görülmektedir. Ayrıca, literatürde birçok çalışmada hedef hızının 20-30 m/sn'yi aşmadığı varsayılmıştır. Çalışmalarımız, mevcut yaklaşımlarda hedefin hızı arttıkça düğümler tarafından algılanmadan ilerleme durumunun ortaya çıkabildiğini ve bunun hedefin konumunun hesaplanamamasına yol açtığını göstermektedir.

Bu tez ile yüksek hızla ya da manevralar yaparak ilerleyen hedefin takibini minimum hedef kaybetme riski ile gerçekleştiren dinamik bir hedef takip sisteminin tasarlanması hedeflenmiştir. Bu amaçla, dağıtık algoritmalar kullanarak yeni hedef takibi sistemleri geliştirilmiştir. Geliştirdiğimiz sistemlerdeki ortak fikir, hedefin ilerlediği yön ve hız ile ilişkili olarak önceden düğüm kümeleri oluşturmak ve bu düğümleri uyanık tutarak hedefin tespit edilip, konumunun hesaplanmasını sağlamaktır. Ayrıca, ani dönüş ve ivmelenme gibi hareketlerde hedefin sonraki konumunu tahminleyebilmek için Parçacık Filtreleri ve Genişletilmiş Kalman Filtreleri tahminleme teknikleri kullanılmıştır.

Geliştirdiğimiz sistemlerin hedefi kaybetme riski hakkında bilgi sahibi olabilmek için mevcut yaklaşımlar ile ve kendi aralarına performans analiz karşılaştırmaları yapılmıştır. Mevcut hedef takip yaklaşımı ile karşılaştırıldığında, geliştirdiğimiz sistemlerin hepsinde çok yüksek hızlarda bile hedefi kaybetme riskinin çok daha düşük olduğu görülmüştür.

3. DAĞITIK AKIŞ TABANLI KAPSAMA AĞACI ve ROTA BELİRLEME

Ağ ortamında hedef ilerledikçe, duyarga düğümleri hedefi algılar ve lider / kök düğümlerine algıladıkları veriyi iletirler. Lider düğüm, üye düğümlerinden topladığı veriler ile hedefin konumunu hesaplar ve bu konum bilgisini Dağıtık Akış Tabanlı Kapsama Ağacı (DATKA) üzerinden Ana Düğüme (Sink) iletirler. Şekil 3.1’de, gerçekleştirilen işlem genel olarak gösterilmektedir.



Şekil 3.1. Lider Düğümlerinin Ana Düğüme Veri İletimi

Şekil 3.1’de de görüldüğü gibi, hedef izleme algoritması çalışmadan önce, ana düğümden ortamdaki bütün düğümlere ulaşabileceğimiz bir kapsama ağacı oluşturulur. Kapsama ağacı oluşturulduktan sonra, hedef izleme alanına girdiğinde X0 kümesindeki düğümler tarafından algılanır ve X0(l) lider düğümü, kümesindeki düğümlerden elde ettiği veriler ile hedefin konumunu hesaplar. Konum hesaplama işlemi tamamlandıktan sonra veriyi tek bir düğüme toplama (Ing. Convergecast) yöntemi ile kapsama ağacı üzerinden ana düğüme konum bilgisini iletir. Hedef ilerledikçe X1, X2 ve X3 kümelerinin liderleri, hesaplamış oldukları konum bilgilerini ana düğüme iletirler. Şekil 3.1’in sağ üst köşesindeki

küçük resimde, ana düğümde toplanmış olan her bir lider düğümün göndermiş olduğu konum bilgileri gösterilmektedir. Ana düğüm bu konumlardan geçen rotayı Polinom Aradeğerleme yöntemi ile netleştirerek, hedefin şu ana kadar geçtiği yolu belirler.

3.1 Dağıtık Akış Tabanlı Kapsama Ağacı (DATKA)

Bu algoritmanın gerçekleştiriminde akış tabanlı bir yaklaşım kullanılmıştır. DATKA'yı oluştururken kullandığımız sonlu durum makinesi Şekil 3.2'de belirtilmiştir. Senaryolarımızda, veriler küme liderlerinden ana düğüme doğru, kapsama ağacı üzerinden tek yönlü olarak iletileceği için, bir düğümün sadece ata düğümünü bilmesi yeterlidir yani çocuk düğümlerini bilmesi gerekli değildir. Kullanmış olduğumuz sonlu durum makinesi de bu düşünceye dayalı olarak tasarlanmıştır.

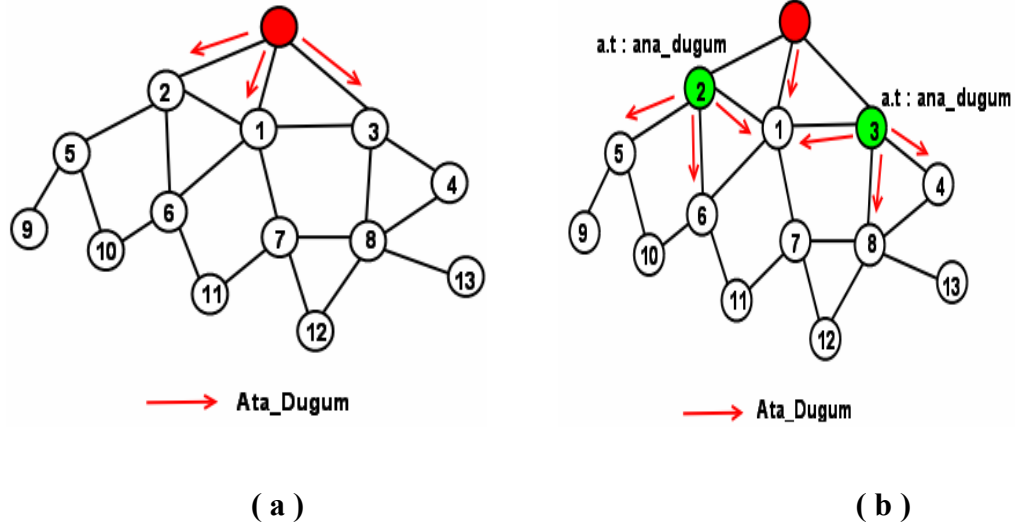


Şekil 3.2. DATKA İçin Kullanılan Sonlu Durum Makinesi

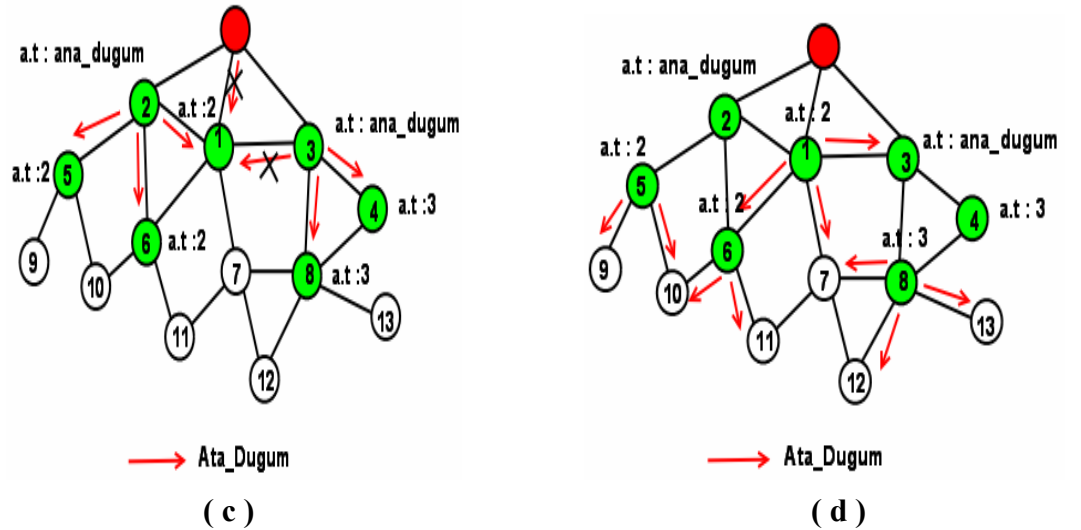
Şekil 3.2'de de görüldüğü gibi başlangıçta bütün düğümler İLK_DURUM durumundadır. Kendisine BAŞLA mesajı gelen düğüm komşu düğümlerine ATA_DÜĞÜM mesajı göndererek ANA_DÜĞÜM durumuna geçer. ANA_DÜĞÜM durumundaki düğüm, oluşturulacak dağıtık kapsama ağacının kök düğümüdür. İLK_DURUM durumundaki bir düğüme, başka bir düğümden ATA_DÜĞÜM mesajı geldiğinde, kendisine ilk ATA_DÜĞÜM mesajını gönderen düğüm, ata düğümü olur ve UYANIK durumuna geçer. Böylece bu düğüme başka düğümlerden ATA_DÜĞÜM mesajı gelse de düğüm UYANIK

duruma geçtiği için değişen bir şey olmaz. İLK_DURUM durumundaki düğüm başka bir düğümden ATA_DÜĞÜM mesajını aldığı anda komşu düğümlerine de almış olduğu ATA_DÜĞÜM mesajını iletir.

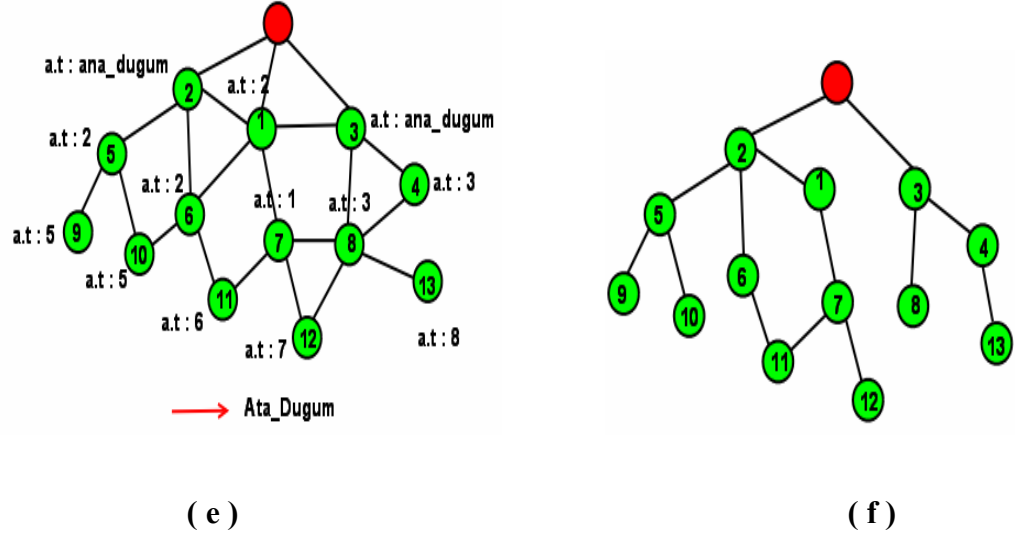
Şekil 3.3, Dağıtık Akış Tabanlı Kapsama Ağacı Algoritmasının bir örnek üzerinden çalışmasını açıklamaktadır.



Şekil 3.3 (a)'da kırmızı renk ile belirtilen ana düğümün kapsama ağacını oluşturma işlemi görülmektedir. Ana düğüm, komşu düğümleri olan 1, 2 ve 3 numaralı düğümlere ATA_DÜĞÜM mesajı gönderir. (b)'de mesajı alan 2 ve 3 numaralı düğümler yeşil renk ile gösterilmiştir. Mesajı alan 2 ve 3 numaralı düğümler ata düğüm olarak ana düğümü belirlemişler ve İLK_DURUM durumundan UYANIK duruma geçiş yaparak komşu düğümlerine ATA_DÜĞÜM mesajı göndermişlerdir.



Şeklin (c) bölümünde 1 numaralı düğüme üç farklı düğümden ATA_DÜĞÜM mesajının geldiği görülmektedir. 1 numaralı düğümün ilk aldığı ATA_DÜĞÜM mesajı 2 numaralı düğümün gönderdiği mesaj olduğu için ata düğüm olarak 2 numaralı düğümü belirler ve UYANIK durumuna geçer. Ana düğümün ve 3 numaralı düğümlerin göndermiş olduğu ATA_DÜĞÜM mesajlarını dikkate almaz. Şeklin (d) bölümünde 1,5,6 ve 8 numaralı düğümlerin mesaj gönderme işlemleri gösterilmektedir.

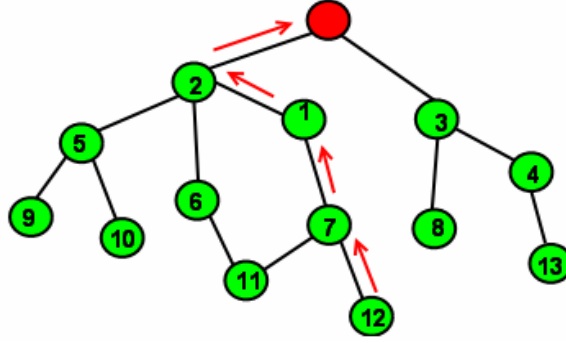


Şekil 3.3. DATKA'nın Örnek Bir İşlem Üzerinde Gösterimi

Şekil 5.5 (e)'de her bir düğümün ata düğümü, (f) bölümünde de en son oluşturulan kapsama ağacı gösterilmektedir.

3.2 Veriyi Tek Bir Merkeze Toplama (Ing.Convergecast)

Veriyi tek bir merkeze toplama işlemi, yaprak düğümlerden ana düğüme doğru verinin kapsama ağacı üzerinden yukarıya doğru iletilmesi işlemidir. Şekil 3.4'de örnek bir veriyi tek bir merkeze toplama işlemi gösterilmektedir. 12 numaralı düğüm, kimlik (id) numarasını, elde etmiş olduğu veriyi ve veriyi elde ettiği zaman bilgisini, MERKEZE_TOPLAMA mesajı içerisine yerleştirerek, kendi ata düğümü olan 7'ye, 7 numaralı düğüm ata düğümü 1'e, 1 numaralı düğüm ata düğümü olan 2'ye, 2 numaralı düğüm ata düğümü olan ana düğüme veriyi iletir. Bu işlemde ana düğüm dışındaki bütün düğümler veriyi aldıkları an, ata düğümlerine iletirler.



Şekil 3.4. Ana Düğüme Veri İletimi

Ana düğüm kendisine gelen MERKEZE_TOPLAMA mesajı içerisindeki bilgileri aşağıdaki gibi bir tabloda tutar.

Çizelge 3.1. Ana Düğümde Bulunan Tablo

Veriyi Gönderen Düğüm	Hedef Konum Bilgisi	Zaman
12	X:123 Y: 239	10.s
12	X: 187 Y: 257	11.s

3.3 Polinom Aradeğerleme Yöntemi (Ing. Polynomial Interpolation)

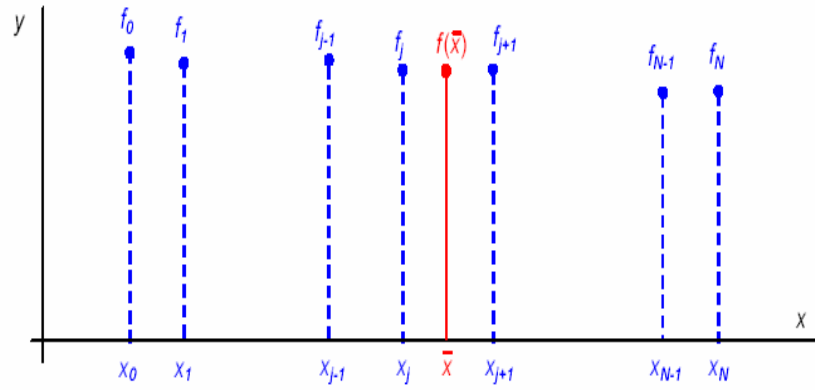
Ana düğüm, kapsama ağacı üzerinden kendisine gelen verileri, Çizelge 3.1’de belirtildiği gibi bir tabloda tutar. Daha sonra bu tablodaki konum ve zaman bilgilerini dikkate alarak, hedefin rotasını belirler. Hedefin, geçmiş olduğu, zamana göre sıralı konumlardan yaklaşık olarak geçen rotayı, Polinom Aradeğerleme yöntemi kullanarak belirler.

Aradeğerleme, bilinen en az iki değeri kullanarak, bu iki değer arasındaki aradaki bilinmeyen bir üçüncü değeri hesaplama işlemidir. Aynı zamanda, verilen bir fonksiyon sınıfından, grafiği verilen sınırlı sayıdaki veri noktasından geçecek şekilde bir $y=p(x)$ fonksiyonu seçme işlemi olarak da tanımlanabilir. Aradeğerleme teorisinin bir çok önemli uygulaması vardır. İki boyutlu görüntülerin ölçeklenmesi, yaklaştırma teorisi, sayısal entegrasyon ve diferansiyel

denklemlerin sayısal çözümlemesi gibi çok önemli çalışma alanlarında kullanılmaktadır (Öztürk, 2008).

Bir sistemin çalışmasının izah etmenin etkili yollarından birisi de matematiksel modelini oluşturmaktır. Literatürde matematiksel modeli oluşturmak için birçok algoritma mevcuttur. Tez kapsamında, hareketli nesnenin konumundaki değişimi belirleyebilmek ve rotasını gösterebilmek için Polinom Aradeğerleme yöntemi kullanılmıştır.

Polinom Aradeğerleme yöntemi, en basit tanımı ile "varolan sayısal değerleri kullanarak, boş noktalardaki değerlerin tahmin edilmesi" olarak açıklanmaktadır. Aradeğerleme genelde, toplanan (x_i, f_i) veri noktalarının bir fonksiyon eğrisine uydurulması amacıyla kullanılmaktadır. Bu amaçla kullanılan fonksiyonlara "aradeğerleme fonksiyonları" denir (Yükselen, 2008).



Şekil 3.5 Polinom Aradeğerleme Fonksiyonları

Şekil 3.5'ten de görülebileceği gibi aradeğerleme teknikleri ile, bir fonksiyonun $x_i (i = 0, 1, \dots, N)$ noktalarında, bilinen $f_i (i = 0, 1, \dots, N)$ değerlerinden hareketle herhangi bir $\bar{x}$ ara noktasında, bilinmeyen $f(\bar{x})$ ara değerinin bulunması istenmektedir (Yükselen, 2008).

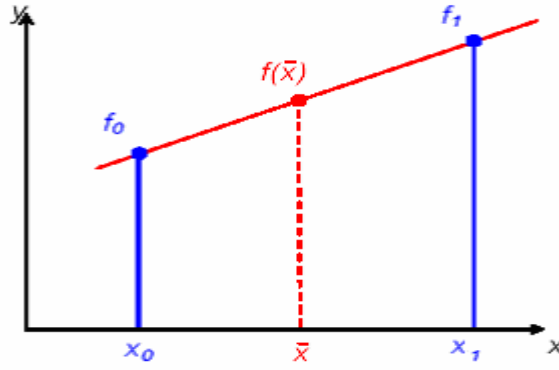
Aradeğerleme fonksiyonu olarak çoğu zaman çeşitli mertebeden polinomlar kullanılır. Ancak bazı hallerde logaritmik, üstel, hiperbolik gibi daha özel fonksiyonlar, periyodik veri değerleri için trigonometrik fonksiyonlar kullanılabilir. Veri noktaları eşit aralıklı olarak dağılmışsa sonlu fark esaslı aradeğerleme yöntemleri, eşit aralıklı değilse doğrusal aradeğerleme, Lagrange aradeğerleme vb yöntemler daha uygun olur (Yükselen, 2008).

Tez kapsamında aradeğerleme için kullanılan koordinat bilgileri eşit aralıkta bir dağılıma sahip olmadığı için doğrusal veya polinom aradeğerleme teknikleri kullanılmıştır.

3.3.1 Doğrusal Aradeğerleme (1.dereceden Polinom İnterpolasyonu)

Elimizdeki veri noktaları eşit aralıklı bir dağılıma sahip değilse, ara bir noktanın karşılığı olan değeri bulmak için doğrusal aradeğerleme yöntemi kullanılır.

Doğrusal aradeğerleme için örnek bir grafik Şekil 3.6’da verilmiştir (Yükselen, 2008).



Şekil 3.6 Doğrusal Aradeğerleme

(x_0, y_0) ve (x_1, y_1) gibi iki noktadan,

$$f(x) = a_0 + a_1x \quad (1)$$

şeklinde bir doğru geçirerek aradeğerleme yapılmak istenirse, nokta koordinatları doğru denklemini sağlayacağı için

$$f_0 = a_0 + a_1x_0 \quad \text{ve} \quad f_1 = a_0 + a_1x_1 \quad (2)$$

gibi iki denklem elde edilir (Yükselen, 2008). Bu denklem sistemi çözülerek katsayılar için

$$\begin{aligned}
a_0 &= \frac{f_0 x_1 - f_1 x_0}{x_1 - x_0} \\
a_1 &= \frac{f_1 - f_0}{x_1 - x_0}
\end{aligned} \tag{3}$$

elde edilir. Böylece doğru denklemi:

$$f(x) = \frac{f_0 x_1 - f_1 x_0}{x_1 - x_0} + \frac{f_1 - f_0}{x_1 - x_0} x \tag{4}$$

şekline gelir. Bu denklem f_0 ve f_1 için düzenlenerek

$$f(x) = L_0 f_0 + L_1 f_1 \tag{5}$$

$$L_0 = \frac{x - x_1}{x_0 - x_1} \quad L_1 = \frac{x - x_0}{x_1 - x_0} \tag{6}$$

şeklinde de yazılabilir (Yükselen, 2008).

3.3.2 n.dereceden Polinom Aradeğerleme (Lagrange Polinomları)

$(x_0, f_0), (x_1, f_1), (x_2, f_2)$ gibi üç noktadan,

$$f(x) = a_0 + a_1 x + a_2 x^2 \tag{7}$$

şeklinde bir parabol geçirilerek aradeğerleme yapılmak istenirse, bu defa nokta koordinatları yardımıyla

$$\begin{aligned}
f_0 &= a_0 + a_1 x_0 + a_2 x_0^2 \\
f_1 &= a_0 + a_1 x_1 + a_2 x_1^2 \\
f_2 &= a_0 + a_1 x_2 + a_2 x_2^2
\end{aligned} \tag{8}$$

denklem sistemi elde edilir ki bu denklem sisteminin çözümü sonucunda katsayılar

$$\begin{aligned}
a_0 &= \frac{f_0(x_1x_2^2 - x_2x_1^2) + f_1(x_2x_0^2 - x_0x_2^2) + f_2(x_0x_1^2 - x_1x_0^2)}{(x_0 - x_1)(x_0 - x_2)(x_2 - x_1)} \\
a_1 &= \frac{f_0(x_1^2 - x_2^2) + f_1(x_2^2 - x_0^2) + f_2(x_0^2 - x_1^2)}{(x_0 - x_1)(x_0 - x_2)(x_2 - x_1)} \\
a_0 &= \frac{f_0(x_2 - x_0) + f_1(x_0 - x_2) + f_2(x_1 - x_0)}{(x_0 - x_1)(x_0 - x_2)(x_2 - x_1)}
\end{aligned} \tag{9}$$

olarak bulunur (Yükselen, 2008). Bu ifadede, parabol denklemi f_0, f_1, f_2 için düzenlenerek yine

$$f(x) = L_0 f_0 + L_1 f_1 + L_2 f_2 \tag{10}$$

şeklinde daha basit bir biçime sokulabilir. Buradaki L_i büyüklükleri

$$\begin{aligned}
L_0 &= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \\
L_1 &= \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \\
L_2 &= \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}
\end{aligned} \tag{11}$$

şeklinde olup bu büyüklüklere “Lagrange polinomları” adı verilmektedir (Yükselen, 2008).

Bu ifadelerden hareketle n. dereceden eğriler için,

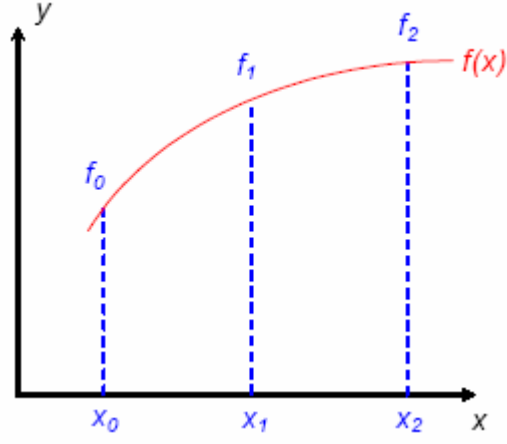
$$f(x) = \sum_{k=0}^N L_k f_k \tag{12}$$

şeklinde bir genelleştirme yapmak mümkündür. Buradaki Lagrange polinomu da ,

$$L_k = \frac{(x - x_0)(x - x_1) \dots (x - x_{k-1})(x - x_{k+1}) \dots (x - x_N)}{(x_k - x_0)(x_k - x_1) \dots (x_k - x_{k-1})(x_k - x_{k+1}) \dots (x_k - x_N)} = \prod_{\substack{j=0 \\ j \neq k}}^N \frac{(x - x_j)}{(x_k - x_j)} \tag{13}$$

şeklinde yazılabilir (Yükselen, 2008).

Polinom aradeğerleme için örnek bir grafik Şekil 3.7’de verilmiştir (Yükselen, 2008).



Şekil 3.7 Polinom Aradeğerleme

Tez kapsamında, akış tabanlı dağıtık kapsama ağacı üzerinden ana düğüme gönderilen $n+1$ adet konum bilgisinden geçecek eğri n . dereceden olacağı için, ana düğümde biriken konum bilgilerinden geçen eğriyi bulabilmek için n . dereceden polinom aradeğerleme yöntemi kullanılmaktadır.

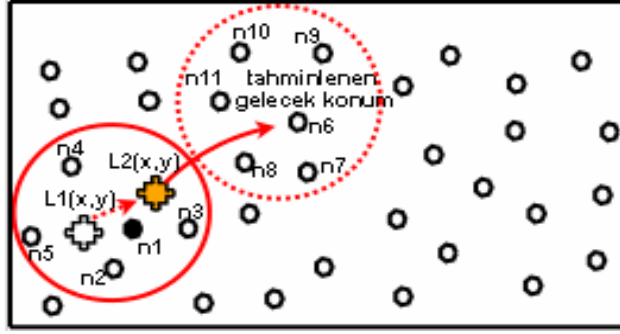
4. DOĞRUSAL HAREKETLER İÇİN DİNAMİK ÇOKLU KÜME TABANLI HEDEF İZLEME ALGORİTMASI (DÇKİA)

Kümeleme yöntemi, telsiz duyurga ağlarında kısıtlı kaynak tüketimini yönetmek ve yönlendirme işlemlerini kolaylaştırmak amacıyla çok yaygın kullanılan bir tekniktir. Kümeleme yönteminde, düğümler mesajlarını sadece küme liderlerine ileterek uzak mesafelere iletimin neden olacağı büyük enerji kayıplarını engellerler ve bu da ağın yaşam süresi arttırır. Kümelenendirilmiş ağlarda, düğümler küme lideri ve küme üyesi olmak üzere iki durumda bulunabilirler. Hedef izleme algoritmalarında küme liderleri, küme içi ve kümeler arası işlemleri yönetmekten sorumlu iken, küme üyeleri de hedefle ilgili algıladıkları veriyi küme liderlerine iletmekten sorumludurlar.

Telsiz duyurga ağlarında hedef takibi uygulamalarında, hareketli hedef, izleme alanına girdiğinde, hedefe yakın olan bir grup düğüm tarafından algılanır ve bu düğümler birbirleri ile işbirliğine geçerek hedefin konumunu hesaplarlar. Üçaçılama ve üçkenarlama (Hightower and Borriello, 2001) teknikleri konum hesaplama için kullanılan en yaygın tekniklerdir. Hedefin konumu hesaplandıktan sonra, küme lideri, başlangıçta oluşturulan dağıtık kapsama ağacı üzerinden konum bilgisini ana düğüme iletir.

Literatürde var olan çalışmalar incelendiğinde, küme tabanlı hedef izleme algoritmaları statik ve dinamik olmak üzere ikiye ayrılır. Statik küme tabanlı yaklaşımlarda küme ve omurga yapısı hedef izleme algoritması çalışmaya başlamadan oluşturulur. Diğer bir deyişle, kümeler, küme üyeleri ve liderleri ağın kurulumu aşamasında belirlenmiş olur. Dinamik yaklaşımda ise, kümeler, hedefin tespit edilmesiyle dinamik olarak oluşturulur. (Alaybeyoglu et al., 2009)'da yapmış olduğumuz çalışmada, dinamik küme tabanlı yaklaşım ile statik küme tabanlı yaklaşımı karşılaştırılmıştır. Elde ettiğimiz sonuçlardan, dinamik küme tabanlı yaklaşımın, hedefi kaybetme olasılığının ve hedefin konumunu hesaplamadaki hata oranının düşmesi açısından daha iyi sonuç verdiği görülmüştür.

Şekil 4.1. literatürdeki dinamik küme tabanlı yaklaşımı genel olarak açıklamaktadır.



Şekil 4.1. Genel Dinamik Küme Tabanlı Hedef İzleme Yaklaşımı (GDKIA)

Şekil 4.1.'den de görüldüğü üzere, hedef izleme alanına girdiğinde $n1, n2, n3$ ve $n4$ düğümleri tarafından algılanır ve bu düğümler birbirleri ile işbirliğine geçerek $n1$ düğümünü lider düğüm olarak seçerler. $n1$ lider düğümü, kümesindeki düğümlerden elde ettiği veriler ile hedefin konumunu, yönünü ve hızını hesaplayarak, hedefin $t+1$ anındaki konumunu tahminler ve bu konuma en yakın olan düğüme UYARI mesajı gönderir. Bu mesajı alan düğüm, hedefin kendisine yaklaştığını anlayarak, kendisinden tek sıçrama mesafesindeki düğümler ile kümesini oluşturur. Yeni oluşturulan küme, artık hedefi algılamaya hazır bulunmaktadır. Algoritma 4.1, Genel Dinamik Küme Tabanlı Hedef İzleme Algoritmasının (GDKIA) sözde kodunu (Ing. pseudo code) göstermektedir.

Algoritma 4.1 i düğümü için GDKIA Algoritmasının Sözde Kodu

- 1: Eger Ben bir *normal* düğüm isem
 - 2: Eger Hedef algılanmış ise
 - 3: Bir lider_j düğüm seç
 - 4: Eger Ben.Liderim mesajı alınmış ise
 - 5: uye düğüm ol
 - 6: Eger Ben bir *uye* düğüm isem
 - 7: Eger Hedef algılanmış ise
 - 8: lider_j düğümüne *Hedef_Bilgisi* mesajı gönder
 - 9: Eger *Uyari* mesajı alınmış ise
 - 10: Hedefin tahminlenen konumuna en yakın düğümüne *Uyari* mesajı gönder
 - 11: Eger Ben bir *lider* düğüm isem
 - 12: Eger *Hedef_Bilgisi* mesajı alınmış ise
 - 13: Hedefin konumunu, hızını ve yönünü hesapla
 - 14: Hedefin tahminlenen konumuna en yakın düğümüne *Uyari* mesajı gönder
-

Algoritma 4.1. GDKIA Algoritması Sözde Kodu

Tez kapsamında, dinamik kümeleme algoritmasının daha iyi sonuç vermesinden yola çıkarak, literatürde var olan bu dinamik kümeleme yaklaşımının, çok hızlı hareket eden hedefler için Çoklu Küme (Ing. MultiClustered) sürümünü önerdik.

Tasarlamış ve uygulamasını gerçekleştirmiş olduğumuz Dinamik Çoklu Küme Tabanlı Hedef İzleme Algoritması'nın (DÇKIA) detayları aşağıdaki bölümlerde verilmiştir.

4.1 DÇKIA Algoritmasının Tanımı

DÇKIA ile, çok hızlı hareket eden bir hedefin, kaybolma riskini minimuma düşürerek takip edilmesini sağlayan bir algoritmanın gerçekleştirilmesi planlanmıştır. Bu amaçla, hareketli hedefin tahminlenen yönü üzerinde, dinamik olarak çoklu kümeler oluşturulmaktadır. Böylece hedef, tahminlenen konumuna daha ulaşmadan, kümeler hazır olacak ve düğümler de uyanık duruma geçerek hedefin gelmesini bekliyor olacaktır. Algoritmanın işleyişi temel olarak iki adımdan oluşmaktadır. İlk adımda, hedef algılanır ve başlangıç kümesi oluşturulur. Bu adımda küme lideri seçimi için literatürdeki yaklaşımlardan (Abbasi et al., 2007) iki aşamalı lider düğüm seçme yöntemi kullanılmıştır. Bu mekanizmaya göre ilk aşamada hedefi algılayan düğümler, komşu düğümlerine hedefe olan uzaklık ve kimlik bilgilerini iletirler. Eğer düğüme, hedefe kendisinden daha yakın olan bir komşu düğümünden mesaj gelmezse, kendisini aday lider düğüm olarak ilan eder. Hedefi algılayan bütün düğümler birbirlerinden tek sıçramalık mesafede olmayabileceği için ilk aşama sonucunda küme lideri olmak için bir kaç aday düğüm çıkabilir. İkinci aşamada ise sadece aday düğümler, diğer düğümlere hedefe olan uzaklık ve kimlik bilgisini sel (Ing. Flooding) yöntemi ile yayar. Herhangi bir aday düğüm, hedefe kendisinden daha yakın bir aday düğümünden mesaj alırsa, liderliği kaybeder. Hedefe en yakın olan aday düğüm küme lideri olarak seçilir.

Algoritmanın ikinci adımı ise hedefin takibinin gerçekleştirilmesidir. Bu adımda, hedefi izlemek için hedefin yönü doğrultusunda kümelerin oluşturulması önerilmiştir. Hedefin tahminlenen yönünde, önceden kaç adet küme oluşturulacağı, hedefin hızına bağlı olarak, (1)'deki formül ile belirlenmektedir.

$$ks_{t+1} = ks_t * e^{-a} \quad (1)$$

(1)'de belirtilen formülde, ks_t , başlangıç küme sayısını, ks_{t+1} , hedefin hızlanmasıyla belirlenen küme sayısı ve a da ivmeyi belirtmektedir. Bu formüldeki ks_t başlangıç küme sayısı hız değerleri 30-40 m/s için 2, 50-60 m/s için 3, 70-80 m/s için 4 ve 90-100 m/s için 5 olarak belirlenmiştir. Formüle göre hız sabit kaldığı sürece başlangıçta belirtilen küme sayısı kadar önceden küme

tahmini yönü doğrultusunda önceden oluşturduğu kümeler mor kesik çizgiler ile gösterilmiştir. Hedef, önceden oluşturulmuş kümelere her girişinde, ilgili kümedeki uyanık durumda bekleyen düğümler, hedef ile ilgili topladıkları bilgileri lider düğümlerine iletirler. Her küme lideri de, elde ettiği bilgiler doğrultusunda hedefin yeni konumu ve yönünü hesaplayarak, hedefin yeni yönü doğrultusunda kümeler oluşturur. Şekil 4.2.'de $X_0(l)$ ve $X_1(l)$ lider düğümlerinin hedefin yön tahminleri aynıdır. Hedef, X_2 kümesine geldiğinde ise, $X_2(l)$ lider düğümü hedefin yönünün değiştiğini fark etmiş ve hedefin yeni yönünü belirlemiştir. Bu doğru yönünde, turuncu renk ile belirtilen kesikli kümeler oluşturulmuştur. Siyah, kesiksiz çizilen daireler ise hedefin mevcut bulunduğu kümeyi belirtmektedir.

Algoritma 4.2., DÇKIA Algoritması'nın sözde kodunu göstermektedir.

Algoritma 4.2 i düğümü için DÇKIA Algoritmasının Sözde Kodu

```

1: Eger Ben bir normal düğüm isem
2:   Eger Hedef algılanmış ise
3:     Bir liderj düğümü seç
4:   Eger Ben_Liderim mesajı alınmış ise
5:     uye düğüm ol
6: Eger Ben bir uye düğüm isem
7:   Eger hedef algılanmış ise
8:     liderj düğümüne Hedef_Bilgisi mesajını gönder
9:   Eger Sen_Lidersin mesajı alınmış ise
10:    Eger Onceden_Olusturulacak_Kume_Sayisi sifirdan büyük ise
11:      Sen_Lidersin mesajını, hedefin yönüne doğrultusundaki komşu düğüme gönder
12:    Eger Onceden_Olusturulacak_Kume_Sayisi sifira eşit ise
13:      lider düğüm ol
14: Eger Ben bir lider düğüm isem
15:   Eger Hedef_Bilgisi mesajı alınmış ise
16:     Hedefin konumunu, hızını, yönünü hesapla
17:     Onceden_Olusturulacak_Kume_Sayisi hesapla
18:     Sen_Lidersin mesajını, hedefin yönü doğrultusundaki düğüme gönder

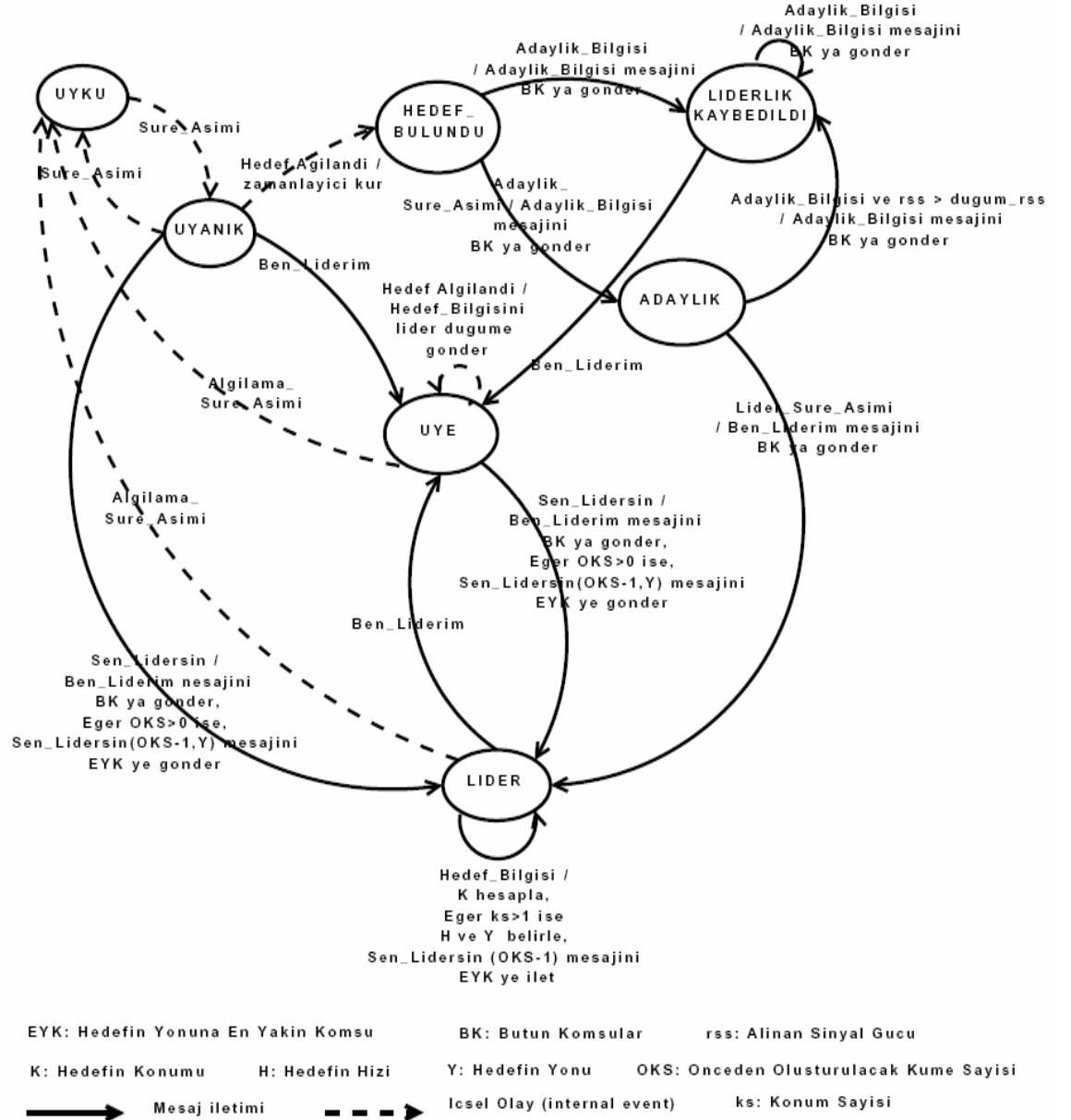
```

Algoritma 4.2. DÇKIA Algoritmasının Sözde Kodu

4.2 DÇKIA Algoritması Sonlu Durum Makinesi

Şekil 4.3. , DÇKIA Algoritmasının sonlu durum makinesini göstermektedir. Buna göre, bir düğüm, UYKU, UYANIK, HEDEF_BULUNDU, LİDERLİK_KAYBEDİLDİ, ADAY, ÜYE, LİDER durumlarından birinde olabilir ve ADAYLIK_BİLGİSİ, BEN_LİDERİM, SEN_LİDERSİN, HEDEF_BİLGİSİ mesajlarından birini alıp gönderebilir. Bunların yanında belirli bir süre beklenildiğini ve bu bekleme süresi dolduğunda zaman aşımı olduğunu belirten içsel olaylar da SÜRE_AŞIMI, ADAYLIK_SÜRE_AŞIMI, LİDERLİK_SÜRE_AŞIMI, ALGILAMA_SÜRE_AŞIMI olarak ifade edilmiştir.

Bu mesaj tiplerinden birini alması ya da içsel olaylardan birinin gerçekleşmesi halinde, bir düğümün durumunda olacak değişiklik, Şekil 4.3. 'teki sonlu durum makinesi üzerinde gösterilmiş ve devamında sonlu durum makinesinde kullanılan durum ve mesajların açıklamaları yapılmıştır. Sonlu durum makinesinde, süre aşırımları gibi içsel olaylar kesikli çizgi ile ifade edilirken, mesaj alımları kesiksiz çizgi ile belirtilmiştir.



Şekil 4.3. DÇKIA'nın Sonlu Durum Makinesi

Şekil 4.3.'te gösterilen sonlu durum makinesinde kullanılan durum tanımlamaları:

- **UYKU:** UYKU durumunda bulunan bir düğüm algılama, mesaj gönderim ve alımında bulunmaz. Dolayısıyla en fazla enerji kazancı, düğüm UYKU durumdayken sağlanır. Enerji tüketimini azaltmak için, düğümler periyodik olarak UYKU ve UYANIK durumları arasında geçiş yaparlar.
- **UYANIK:** Düğüm UYANIK durumdayken, algılama, mesaj gönderim ve alımında bulunulabilir. Enerji tüketimini azaltmak için periyodik olarak UYKU durumuna geçiş olur. Düğüm UYANIK durumdayken, hareketli hedeften sinyal alması durumunda HEDEF_BULUNDU durumuna geçer ve elde ettiği sinyal gücü değeri ile ters orantılı olacak şekilde bir zamanlayıcı kurar.
- **HEDEF_BULUNDU:** Düğümün hedefi algıladığını belirten bir durumdur. Düğüm, HEDEF_BULUNDU durumundayken zamanlayıcısı dolarsa, liderliğe adaylık anlamına gelen ADAY durumuna geçer ve komşularına elde ettiği sinyal gücü değerini içeren ADAYLIK_BİLGİSİ mesajını iletir. Eğer düğüm HEDEF_BULUNDU durumundayken ve henüz zamanlayıcısı dolmamışken, ADAYLIK_BİLGİSİ mesajı alırsa, liderlik yarışını kaybetmiş demektir ve LİDERLİK_KAYBEDİLDİ durumuna geçerek, kendisine gelen ADAYLIK_BİLGİSİ mesajını komşu düğümlerine iletir.
- **LİDERLİK_KAYBEDİLDİ:** Düğümün liderlik yarışını kaybettiğini belirten bir durumdur ve bu durumda hedefe kendisinden daha yakın bir düğüm bulunmaktadır. Düğüm LİDERLİK_KAYBEDİLDİ durumundayken ADAYLIK_BİLGİSİ mesajı gelirse, bu mesajı komşularına iletir.
- **ADAYLIK:** Düğümün liderliğe aday olduğunu belirten bir durumdur. Eğer düğüm bu durumda iken, kendisinin algılamış olduğu sinyal gücü değerinden daha büyük bir değere sahip bir düğümün ADAYLIK_BİLGİSİ mesajını alırsa, liderliği kaybeder ve LİDERLİK_KAYBEDİLDİ durumuna geçer. Ayrıca kendisine gelen ADAYLIK_BİLGİSİ mesajını komşu düğümlerine iletir.
- **ÜYE:** Düğümün, bir kümenin üyesi olduğunu belirten bir durumdur ve hedefi algıladıkça lider düğüme elde ettiği sinyal gücü değerini iletir. Düğüm ÜYE durumundayken BEN_LİDERİM mesajı gelirse, başka bir

lider düğümün oluşturacağı kümeye de üyelik yapacağı anlamına gelir ve elde ettiği sinyal gücü bilgisini yeni lider düğüme de iletir. Düğüm, ÜYE durumundayken SEN_LİDERSİN mesajı gelirse, LİDER durumuna geçer ve komşu düğümlerine BEN_LİDERİM mesajını göndererek kendi kümesini oluşturur. ÜYE durumunda olan bir düğüm belirli bir süre hedeften sinyal almamışsa, hedef bu düğümden uzaktadır demektir ve bu düğümün daha fazla uyanık kalarak enerji kaybetmemesi için UYKU durumuna geçer.

- **LİDER:** Düğümün, bir kümenin lideri olduğu anlamına gelen bir durumdur ve üye düğümlerinden periyodik olarak almış olduğu bilgileri kullanarak, hedefin konumunu, hızını ve yönünü tespit eder. Hedefin hızına bağlı olarak, önceden oluşturulacak küme sayısını belirler ve SEN_LİDERSİN mesajı ile önceden küme oluşturma işlemini başlatır.

Şekil 4.3’te gösterilen sonlu durum makinesinde kullanılan mesaj tanımlamaları:

- **ADAYLIK_BİLGİSİ:** Düğümün, liderliğe aday olduğunda komşu düğümlere ilettiği ve içinde kimlik ile hedeften alınan sinyal gücü bilgisini barındıran mesaj tipidir.

Mesajın içerdiği alanlar:

- o **Gönderen:** Adaylığını ilan eden düğümün kimlik bilgisidir.
- o **Rss:** Düğümün, hedeften elde ettiği sinyal gücü değeridir.
- o **Mesaj Tipi:** Düğümün adaylığını ilan ettiği anlamına gelen “Adaylık_Bilgisi” bulunmaktadır.
- **BEN_LİDERİM:** Lider düğümün, kümesini oluşturmak için komşu düğümlerine göndermiş olduğu, içinde kendi kimlik bilgisi bulunan mesajdır. Bu mesajı alan düğümler, ilgili liderin oluşturmuş olduğu kümenin üyesi olurlar.

Mesajın içerdiği alanlar:

- o **Gönderen:** Liderliğini ilan eden düğümün kimlik bilgisidir.
- o **Mesaj Tipi:** Düğümün liderliğini ilan ettiği anlamına gelen “Ben_Liderim” bulunmaktadır.
- **SEN_LİDERSİN:** Lider düğümün, hedefin ilerlediği yönde önceden kümeleri oluşturmak amacıyla, hedefin yönüne en yakın komşu düğümüne göndermiş olduğu mesajdır. Bu mesajı alan düğüm LİDER durumuna geçer ve BEN_LİDERİM mesajını komşu düğümlerine ileterek kümesini oluşturur. Eğer bu mesajın içindeki önceden oluşturulacak küme sayısı değeri sıfırdan büyükse, mesajın içinde belirtilen yöne en yakın komşu düğümüne bu mesaj iletilir ve önceden oluşturulacak küme sayısı değeri bir azaltılır.

Mesajın içerdiği alanlar:

- o **Gönderen:** Mesajı gönderen düğümün kimlik bilgisi bulunur.
- o **Mesaj Tipi:** Düğümün önceden küme oluşturmak için komşu düğümlerine göndermiş olduğu “Sen_Lidersin” tipindeki mesaj.
- o **Sayaç:** Önceden oluşturulacak kümelerin çok fazla üst üste çakışmasını engellemek için kullanılır. Lider düğümler, birer düğüm arayla oluşturulmuş olur.
- o **Önceden_Uyandırılacak_Küme_Sayısı:** Sen_Lidersin mesajını alan her düğüm önce sayaç alanını kontrol eder. Eğer bu değer sıfıra eşitse, düğüm lider olur ve Ben_Liderim mesajı göndererek kendi kümesini oluşturur. Daha sonra, Önceden Uyandırılacak_Küme_Sayısı alanındaki değerin birden büyük olup olmadığına bakar. Eğer büyükse, bu değeri bir azaltarak, Sen_Lidersin mesajını göndermeye devam eder.
- o **Yön:** Hedefin hesaplanan yön bilgisidir. Hedefin yönü doğrultusundaki düğüme Sen_Lidersin mesajı gönderilir.

- **HEDEF_BİLGİSİ:** ÜYE durumundaki düğümün, hedeften elde ettiği sinyal gücü değerini, lider düğümüne iletmek için kullandığı mesaj tipidir.

Mesajın içerdiği alanlar:

- o **Gönderen:** Mesajı gönderen düğümün kimlik bilgisi bulunur.
- o **Mesaj Tipi:** Üye düğümün algıladığı veriyi göndermesi anlamına gelen “Hedef_Bilgisi” tipindeki mesajdır.
- o **Alınan_Sinyal_Gücü:** Üye düğümün, hedeften algıladığı sinyal gücü değeridir.

Yukarıda belirtilen durum ve mesaj bilgilerine ek olarak aşağıda belirtilen içsel olaylar (Ing. Internal event) da gerçekleşmektedir.

- **SÜRE_AŞIMI:** Düğümün, enerji tüketimini azaltmak amacıyla periyodik olarak UYKU ve UYANIK durumlarına geçişini sağlayan zamanlayıcının süresinin dolduğunu belirten içsel bir olaydır.
- **ADAYLIK_SÜRE_AŞIMI:** HEDEF_BULUNDU durumundaki düğümün ADAY durumuna geçişini sağlayan zamanlayıcının süresinin dolduğunu belirten içsel bir olaydır.
- **LİDERLİK_SÜRE_AŞIMI:** ADAY durumundaki düğümün LİDER durumuna geçişini sağlayan zamanlayıcının süresinin dolduğunu belirten içsel bir olaydır.
- **ALGILAMA_SÜRE_AŞIMI:** ÜYE veya LİDER durumundaki düğümlerin belirli bir süre hedefi algılamadıklarında, bu sürenin dolduğunu belirten içsel bir olaydır. Bu süre dolduğunda düğümler UYKU durumuna geçerler.

Önermiş olduğumuz Dinamik Çoklu Küme Tabanlı Hedef İzleme Algoritmasının dağıtık açık kodu Algoritma 4.3.’te verilmiştir.

Algoritma 4.3 dugum j için DCKIA Algoritması Dağıtık Kodu

```

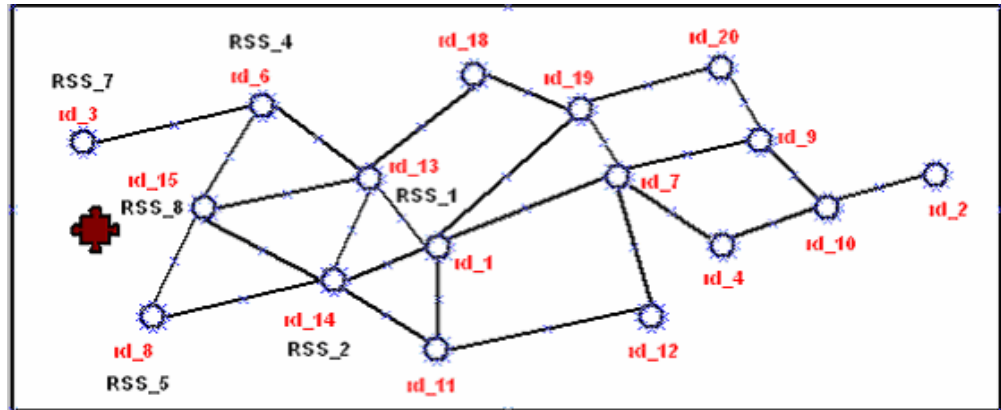
1: baslangic  $mevcut\_durum_j$  = UYANIK
2:    $Lejant : \square DURUM \wedge gelen\_mesaj \rightarrow eylem$ 
3:    $S$ : Hedefin Hizi,  $K_t$ : Hedefin  $t$  anındaki konumu,  $Y$ : Hedefin Yonu
4:    $k_Y$ :  $Y$ 'ye en yakın komşu,  $\Gamma_j$ :  $j$ 'nin komşuları
5:    $OKS$ : Önceden oluşturulacak kume sayısı
6:
7:    $\square UYANIK \wedge Ben\_Liderim(i) \rightarrow lider_j \leftarrow id_i, mevcut\_durum_j \leftarrow UYE, Zamanlayici \leftarrow ata$ 
8:        $\wedge Hedef\_Sinyal \rightarrow mevcut\_durum_j \leftarrow HEDEF\_BULUNDU, Zamanlayici \leftarrow ata$ 
9:        $\wedge Sure\_Asimi \rightarrow mevcut\_durum_j \leftarrow UYKU, Zamanlayici \leftarrow ata$ 
10:       $\wedge Sen\_Lidersin(OKS, Y) \rightarrow Ben\_Liderim$  mesajını  $\Gamma_j$ 'a gönder,
11:           $Sen\_Lidersin(OKS - 1, Y)$  mesajını  $k_Y$ 'a gönder
12:           $mevcut\_durum_j \leftarrow LIDER, Zamanlayici \leftarrow ata$ 
13:    $\square UYKU \wedge Sure\_Asimi \rightarrow lider_j \leftarrow id_i, mevcut\_durum_j \leftarrow UYANIK, Zamanlayici \leftarrow ata$ 
14:    $\square UYE \wedge Ben\_Liderim(i) \rightarrow lider_j \leftarrow id_i, Zamanlayici \leftarrow ata$ 
15:        $\wedge Hedef\_Sinyal \rightarrow HEDEF\_BILGISI$ 'ni  $lider_j$  dugumüne gönder,  $Zamanlayici \leftarrow ata$ 
16:        $\wedge Sure\_Asimi \rightarrow mevcut\_durum_j \leftarrow UYANIK, Zamanlayici \leftarrow ata$ 
17:        $\wedge Sen\_Lidersin(OKS, Y) \rightarrow Ben\_Liderim$  mesajını  $\Gamma_j$ 'a gönder,
18:           $Sen\_Lidersin(OKS - 1, Y)$  mesajını  $k_Y$ 'a gönder
19:           $mevcut\_durum_j \leftarrow LIDER, Zamanlayici \leftarrow ata$ 
20:    $\square LIDER \wedge Hedef\_Bilgisi \rightarrow Hedefin\ t\ anındaki\ K_t$ 'yi hesapla
21:        $K_t$  ve  $K_{t-1}$ 'den  $Y$  ve  $S$ 'yi hesapla,
22:        $Sen\_Lidersin(OKS, Y)$  mesajını  $k_Y$  gönder
23:        $\wedge Sure\_Asimi \rightarrow mevcut\_durum_j \leftarrow UYANIK, Zamanlayici \leftarrow ata$ 
24:        $\wedge Ben\_Liderim(i) \rightarrow mevcut\_durum_j \leftarrow UYE, Zamanlayici \leftarrow ata, lider_j \leftarrow id_i$ 
25:    $\square HEDEF\_BULUNDU \wedge Adaylik\_Bilgisi(i) \rightarrow Adaylik\_Bilgisi(i)$ 'ni  $\Gamma_j$ 'a ilet,
26:        $mevcut\_durum_j \leftarrow LIDERLIK\_KAYBEDILDI$ 
27:        $\wedge Sure\_Asimi \rightarrow \Gamma_j$ 'ye  $Adaylik\_Bilgisi$  gönder,  $Zamanlayici \leftarrow ata, mevcut\_durum_j \leftarrow ADAY$ 
28:    $\square LIDERLIK\_KAYBEDILDI \wedge Adaylik\_Bilgisi(i) \rightarrow Adaylik\_Bilgisi(i)$ 'ni  $\Gamma_j$ 'ya ilet
29:        $\wedge Ben\_Liderim(i) \rightarrow mevcut\_durum_j \leftarrow UYE, lider_j \leftarrow id_i,$ 
30:    $\square ADAY \wedge Adaylik\_Bilgisi(i, RSS) \rightarrow eger\ RSS > kendi\_RSS\ ise$ 
31:        $Adaylik\_Bilgisi(i, RSS)$ 'ni  $\Gamma_j$ 'e ilet,
32:        $mevcut\_durum_j \leftarrow LIDERLIK\_KAYBEDILDI$ 
33:

```

Algoritma 4.3. DCKIA Algoritmasının Dağıtık Açık Kodu

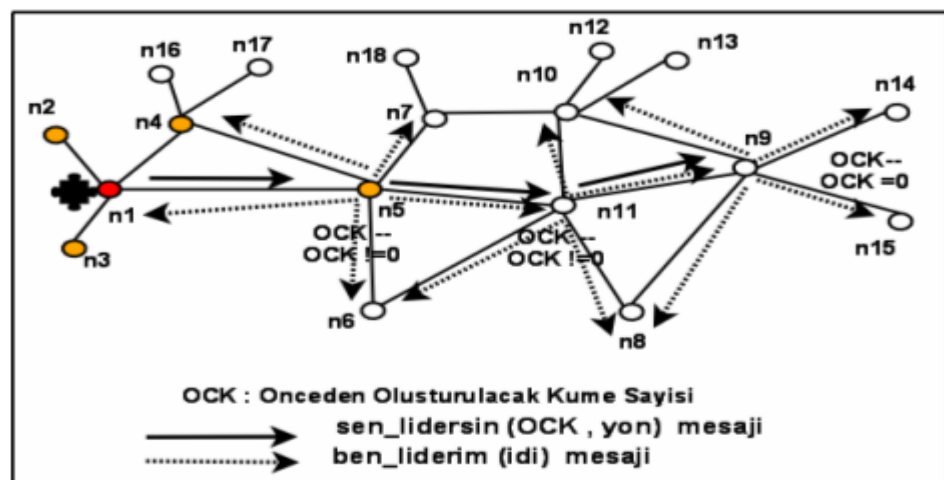
4.3 Örnek Bir İşlem

Bu bölümde, DÇKIA Algoritması'nın bir örnek üzerinde çalışması açıklanmıştır. Şekil 4.4'te hedefin izleme alanına girmesi ve hedefi hangi düğümlerin algıladığı gösterilmiştir.



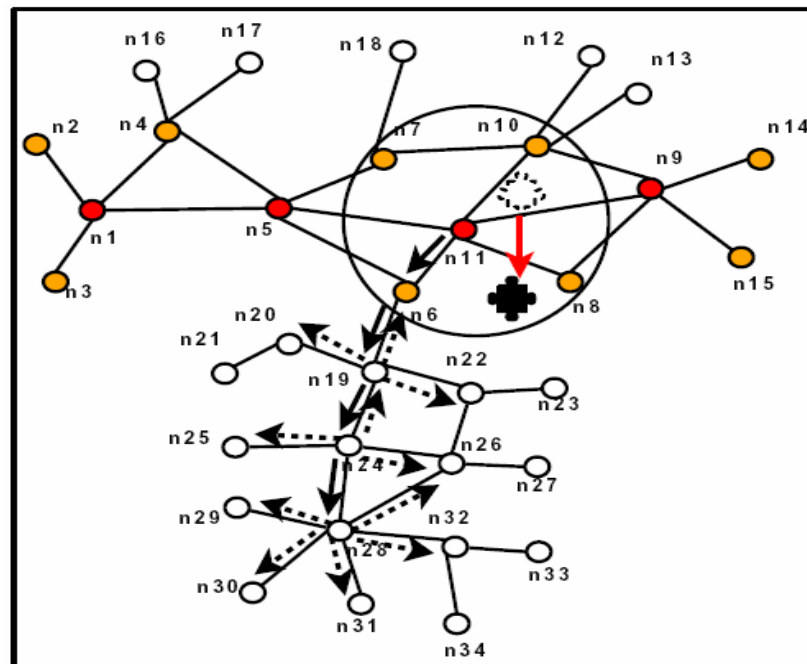
Şekil 4.4. Hedefin İzleme Alanına Girmesi ve Düğümlerin Hedefi Algılaması

Şekil 4.4.'te de görüldüğü gibi 15, 3, 8, 6, 14 ve 13 numaralı düğümler hedefi algılayan düğümlerdir. Düğüm hedefe ne kadar yakın ise elde ettiği sinyal gücü (Ing. Received Signal Strength (rss)) değeri de o kadar yüksektir, yani alınan sinyal gücü değeri hedefe olan uzaklıkla ters orantılıdır. Hedefi algılayan düğümler arasından, hedefle ilgili bir takım hesaplamaları yapması için iki aşamalı lider belirleme algoritması ile bir lider düğüm seçilir. 15 numaralı düğüm, lider düğüm olarak seçilmiş ve kendisinden tek sıçrama mesafesindeki düğümler ile başlangıç kümesini oluşturmuştur.



Şekil 4.5. Önceden Kümelerin Oluşturulması

Şekil 4.5, önceden küme oluşturma işlemini göstermektedir. n1 numaralı mevcut lider düğüm hedefin konumunu, hızını ve yönünü tespit ettikten sonra hedefin yönü doğrultusunda kümeler oluşturmaya başlar. Önceden kaç adet küme oluşturulacağı hedefin hızına göre belirlenir. Bunun için de kümesindeki düğümlerin içinden hedefin yönüne en yakın komşu düğüme SEN_LİDERSİN (önceden_oluşturulacak_küme_sayısı, hedefin_yönü) mesajını iletir. Mesajın içindeki önceden_oluşturulacak_küme_sayısı parametresi kaç adet küme oluşturulacağını belirtirken, hedefin_yönü parametresi de hedefin yönünü ifade etmektedir. n5 numaralı düğüm SEN_LİDERSİN mesajını aldığı anda LİDER durumuna geçerek kendisinden tek sıçrama mesafesinde olan düğümlere BEN_LİDERİM mesajını gönderir ve kümesini oluşturur. Daha sonra SEN_LİDERSİN mesajının içinde gelen önceden_oluşturulacak_küme_sayısı değerini bir azaltarak bu değerın sıfıra eşit olup olmadığını kontrol eder. Eğer bu değer sıfırdan büyükse küme oluşturma işlemine devam edilir.



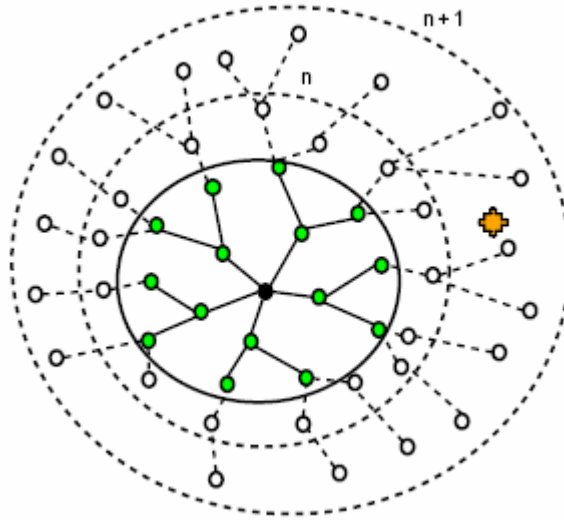
Şekil 4.6. Hedefin Yön Değiştirmesi

Şekil 4.6.'da hedefin yön değiştirmesi durumu gösterilmektedir. n11 numaralı lider düğüm, hedefin yön değiştirdiğini fark eder ve kümeleri hedefin yeni yönü doğrultusunda oluşturur.

4.4 Düzeltme Algoritması

Hedefin çok hızlı ilerlemesi durumlarında, hedef düğümler tarafından algılanamamakta ve konumu hesaplanamamaktadır. Lider düğümün hedefin konumunu hesaplayabilmesi için en az üç üye düğümünden hedefle ilgili bilgi alması gerekir. Eğer yeterli sayıda düğümden bilgi alamazsa hedefin konumunu hesaplayamaz ve bu da hedefin kaybedildiği anlamına gelir. Hedefin kaybedilmesi durumunda, hedefin tekrar bulunabilmesi için düzeltme mekanizmalarına ihtiyaç vardır. Hedefin tekrar bulunabilmesi için kullanılabilecek en basit çözüm ortamdaki bütün düğümlerin uyandırılmasıdır. Fakat bu yöntemde ortamdaki bütün düğümler uyandırıldığı için enerji tüketim miktarı yüksek olacaktır.

Tez kapsamında kullanmış olduğumuz düzeltme algoritmasına göre, lider düğüm hedefin kaybedildiğini fark ettiğinde kümesindeki düğüm sayısını artırır. Bunun için mevcut kümesini n sıçrama mesafesindeki düğümler ile oluşturmuş ise, $n+1$ sıçrama mesafesindeki düğümlerine ALARM mesajı gönderir. ALARM mesajını alan düğümler, hedeften sinyal almışlar ise elde ettikleri sinyal gücü değerini ve kimlik bilgisini CEVAP mesajı içerisine yerleştirerek lider düğüme iletirler. Lider düğüm, hedefin konumunu hesaplamak için yeterli sayıda CEVAP mesajı alıncaya kadar kümesini genişletmeye devam eder. Tez kapsamında kullanılan düzeltme algoritmasının genel işleyişi Şekil 4.7’de gösterilmektedir.



Şekil 4.7. Düzeltme Algoritması

Mevcut yaklaşımlar ile çok hızlı hareket eden hedeflerin takibinde hedefin kaybedilme oranı daha yüksektir. Dolayısıyla mevcut yaklaşımlarda hedefin tekrar bulunması için düzeltme mekanizması daha sık kullanılmakta, bu da enerji tüketiminin daha yüksek olmasına sebep olmaktadır.

4.5 DÇKIA Algoritmasının Mesaj Karmaşıklık Analizi

Dağıtık algoritmaların en önemli tasarım kriterlerinden biri algoritmanın kullandığı mesaj sayısıdır. Bu bölümde, DÇKIA algoritmasının her bir aşamasındaki kullandığı mesaj sayısı adım adım hesaplanarak, mesaj karmaşıklığı belirlenmiştir.

Teorem 4.4.1: DÇKIA algoritmasının mesaj karmaşıklığı $O(n)$ 'dir.

İspat:

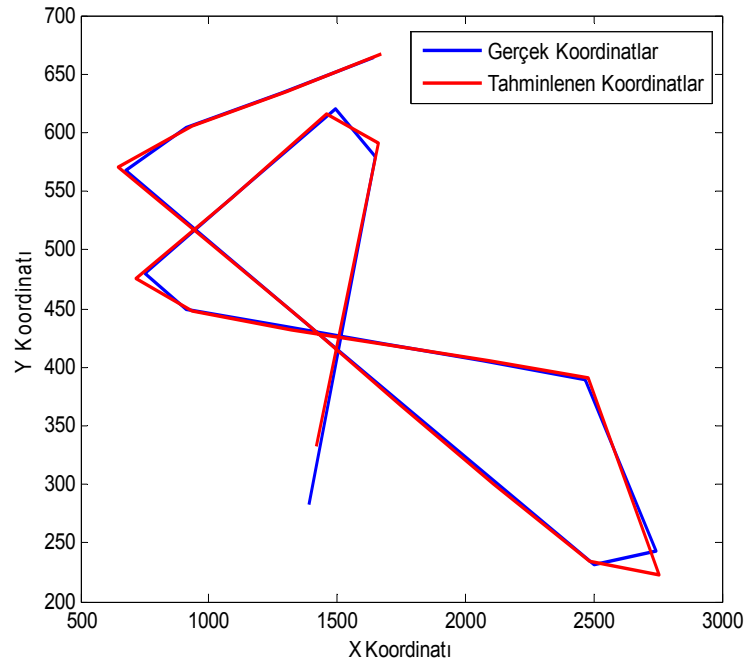
- **Lider düğümü belirleme aşamasında**, hedefi algılayan bir düğüm komşu düğümüne ADAYLIK_BİLGİSİ mesajı gönderir. Bir düğümün ortalama k adet komşusunun olduğunu ve a adet düğümün hedefi algıladığını düşünürsek, bu aşamada kullanılan mesaj sayısı: $k*a$ 'dır.
- **İlk kümenin oluşturulması aşamasında**, seçilen lider düğüm k adet komşu düğümüne BEN_LİDERİM mesajı gönderir. Bu aşamadaki mesaj sayısı: k 'dir.
- **Üye düğümlerin elde ettikleri verileri lider düğüme iletmesi aşamasında**, küme içerisindeki k adet üye düğüm iki farklı zamanda elde ettikleri veriyi lider düğüme iletir. Bu aşamadaki mesaj sayısı: $2*k$ 'dir.
- **Lider düğümün hedefin yönü doğrultusunda küme oluşturma mesajı göndermesi aşamasında**, SEN_LİDERSİN mesajı, hedefin yönüne en yakın düğüme sıçrama sayısı kadar (h) iletilir. Bu durumda tek bir mesaj h kadar iletilmiş olur. Mesaj sayısı: h 'dir.
- **Önceden kümelerin oluşturulması aşamasında**, her bir önceden oluşturulan lider düğüm kümesini oluşturabilmek için k adet komşu düğümüne BEN_LİDERİM mesajı gönderir ve sıçrama sayısı (h) kadar küme oluşturulacağı için bu aşamadaki mesaj sayısı: $k*h$ 'dir.

- Ortamda n adet düğümün olduğunu düşünürsek, simülasyon süresi boyunca ortamda $n/(k+1)$ adet küme oluşturulabilir. $k+1$, bir kümede bulunabilecek düğüm sayısını belirtmektedir.

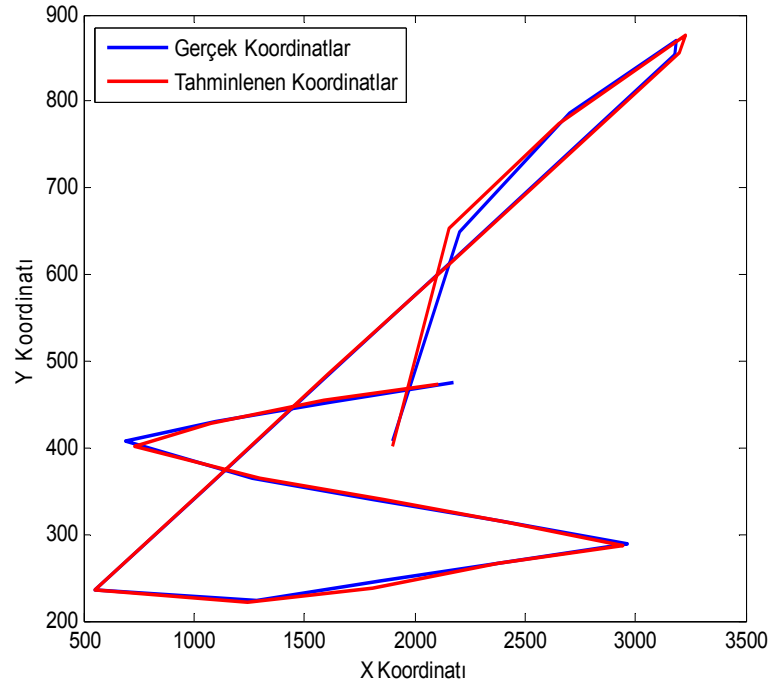
Sonuç olarak, yukarıda belirtilen mesaj sayıları toplandığında, DÇKIA'nın hedefi takip edebilmesi için kullandığı toplam mesaj sayısı $(k*a+3*k+h+k*h)* (n/(k+1))$ olarak hesaplanır ki bu da mesaj karmaşıklığının $O(n)$ olduğu anlamına gelir.

4.6 Simülasyon Sonuçları

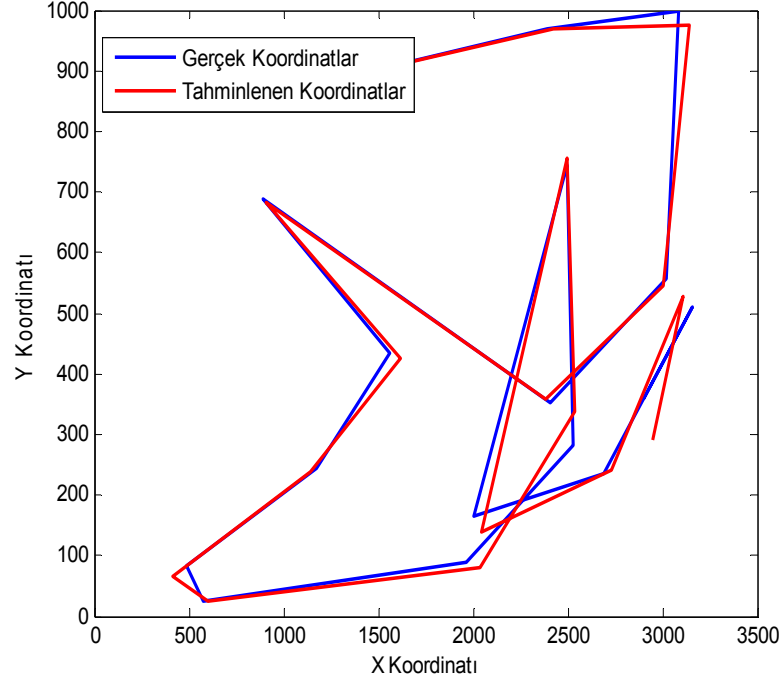
Algoritmalar ns 2.31 simülasyon ortamında gerçekleştirilmiştir. Ölçümler 70'er adet deneylerin ortalaması alınarak belirlendi. Ölçümlerde çok hızlı hareket eden hedefleri incelediğimiz için, hız değerleri olarak literatürde şu ana kadar incelenmemiş değerler olan 30m/s'den 100 m/s'ye kadar hedef takibi gerçekleştirilmiştir. Önerilen algoritma ile her hız değeri için hedefin takip edilebildiğini Şekil 4.8.'den görebilmekteyiz. Hız değerleri arttıkça gerçek rota ile hesaplanan rota arasındaki farkta artış olsa da, hedefin çok hızlı hareket ettiği durumlarda rotada çok büyük bir sapma olmamıştır.



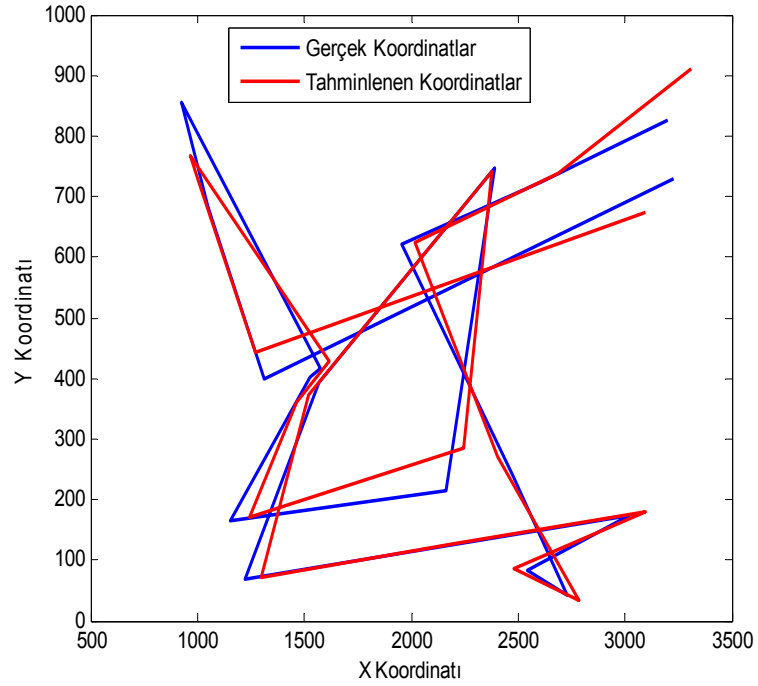
(a) 30-40 m/s



(b) 50-60 m/s



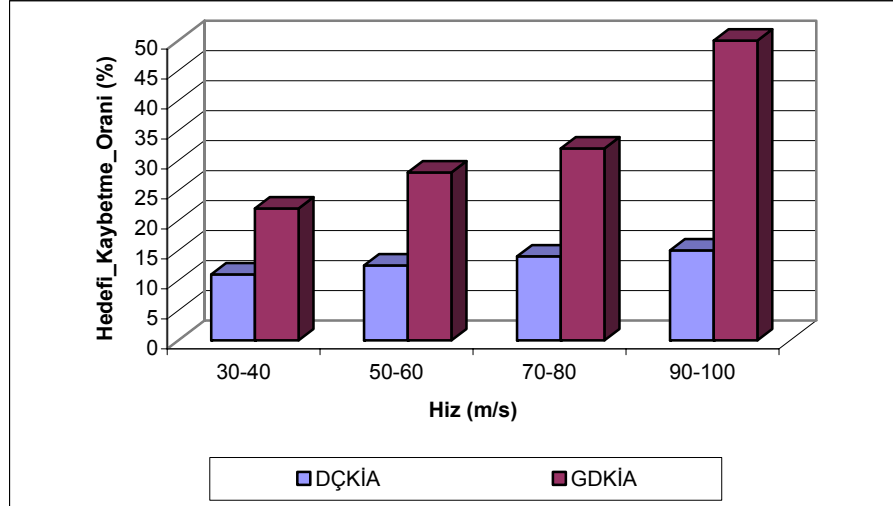
(c) 70-80 m/s



(d) 90-100 m/s

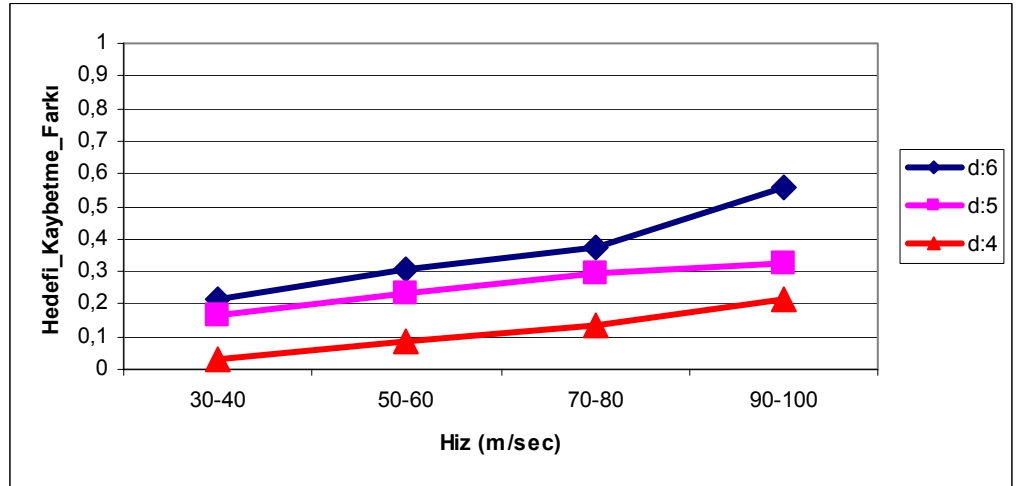
Şekil 4.8. DÇKIA ile Hedef Takibi

Önermiş olduğumuz algoritma ile hedefi kaybetme olasılığını azaltmayı hedeflemiş bulunmaktayız. Şekil 4.9’da, önerilen hedef izleme yaklaşımı ile mevcut yaklaşım hedefi kaybetme oranları için karşılaştırılmıştır. Lider düğüm, kümesindeki düğümlerden elde ettiği veriler ile periyodik olarak hedefin konumunu hesaplamaktadır. Lider düğümün hedefin konumunu hesaplayabilmesi için en az üç üye düğümünden veri alması gerekir. Eğer yeterli sayıda veri elde edemezse hedefin konumunu hesaplayamaz ve bu durum hedefin kaybedilmesi olarak tanımlanır. Buna göre, her bir hız değeri için önerilen yaklaşımın mevcut yaklaşıma göre hedefi kaybetme olasılığı açısından daha iyi sonuç verdiğini görmekteyiz. Bunun yanında hız değeri arttıkça, iki yaklaşım arasındaki hedefi kaybetme oranları arasındaki farkın daha da arttığı görülebilmektedir. Hedefi kaybetme oranı, simülasyon süresi boyunca hedefin konumunun hesaplanamadığı durumların yüzdesi alınarak belirlenmiştir.



Şekil 4.9. DÇKİA ve GDKİA Algoritmalarının Hedefi Kaybetme Olasılığı için Karşılaştırılması

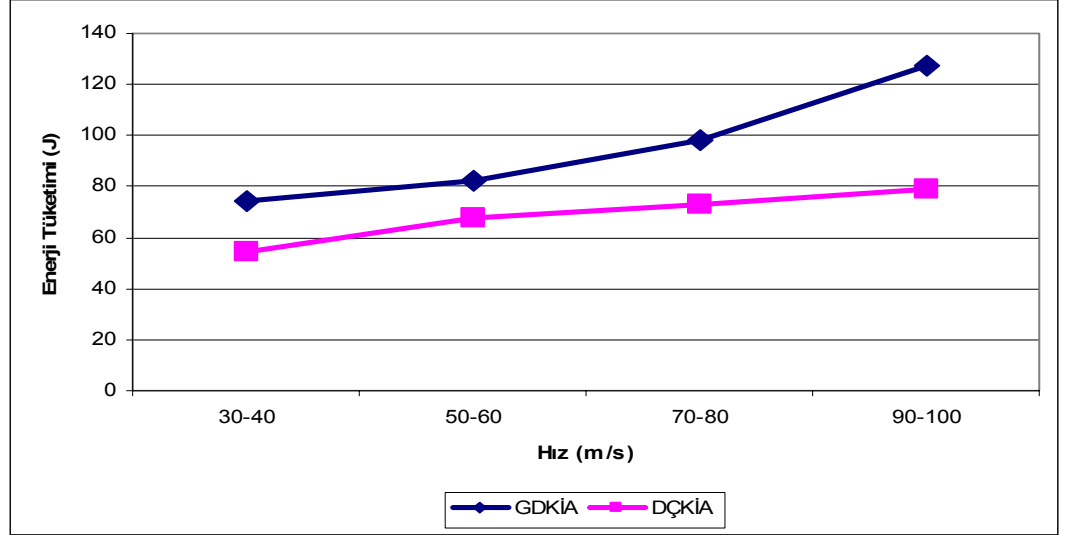
Şekil 4.9.'da önermiş olduğumuz yaklaşım ile mevcut yaklaşım arasındaki hedefi kaybetme olasılık farkları, düğüm derecesi (d) temel alınarak karşılaştırılmıştır. Bir düğümün derecesi, o düğümün sahip olduğu ortalama komşu sayısı olarak tanımlanır.



Şekil 4.10. DÇKİA ve GDKİA Algoritmalarının Hedefi Kaybetme Olasılık Farkının Düğüm Derecesi (d) için Karşılaştırılması

Şekil 4.10'dan da görüldüğü üzere düğüm derecesi arttıkça hedefi kaybetme oranları arasındaki fark da artmaktadır. Bunun nedeni, düğüm derecesi arttıkça, oluşturulacak olan kümeler içindeki düğüm sayısı da artacaktır. Küme içerisindeki

düğüm sayısının artması hedefi kaybetme olasılığını düşürecek, bu da mevcut yaklaşım ile önerilen yaklaşım arasındaki hedefi kaybetme olasılık farkını arttıracaktır.



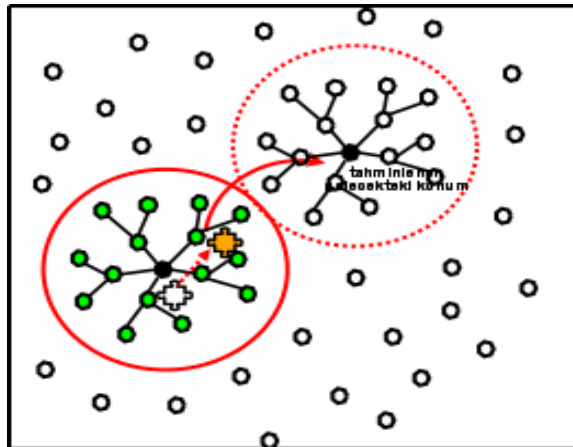
Şekil 4.11. DÇKIA ve GDKIA Algoritmalarının Enerji Tüketimi için Karşılaştırılması

Son olarak Şekil 4.11’de, mevcut yaklaşım ile önermiş olduğumuz yaklaşım enerji tüketimi için karşılaştırılmıştır. Mevcut yaklaşımda hedefi kaybetme oranı yüksek olduğu için düzeltme mekanizmalarının kullanım sıklığı da yüksektir. Düzeltme mekanizması yüksek olması, mevcut yaklaşımlardaki enerji tüketimini de arttırmaktadır. Grafikten de görüldüğü gibi hız arttıkça hedefi kaybetme oranı da arttığı için enerji tüketimi de paralel olarak artmaktadır.

5. DOĞRUSAL HAREKETLER İÇİN DİNAMİK ÇOK KÜMELİ KAPSAMA AĞACI TABANLI HEDEF İZLEME ALGORİTMASI (DÇKAİA)

Bir önceki bölümde önermiş olduğumuz DÇKİA algoritmasında kümeler, lider düğümden tek sıçrama mesafesindeki düğümler ile oluşturulmuştur. Hedefin konumunu hesaplayabilmemiz için en az üç duyarganın hedefi algılamış olması gerekir. Eğer küme içerisinde bulunan düğümlerden hedefi algılayan en az üç düğüm bulunamaz ise hedefin konumu hesaplanamaz. Bu da hedef takip edilebilirliğini olumsuz yönde etkiler. Böyle bir riski azaltmak için kümeler daha çok sayıda düğüm içerecek şekilde oluşturulabilir. Bu çözüm, hedefi algılayan düğüm sayısını arttırarak hem hedefi kaybetme riskini azaltacak hem de hedefle ilgili daha doğru bilgi edinmemizi sağlayacaktır.

Kapsama ağacı tabanlı hedef izleme yaklaşımına göre, kapsama ağaçları izleme alanında hedefin tespit edilmesiyle dinamik olarak oluşturulur. Hedefi algılayan düğümlerden biri kök düğüm olarak seçilir ve kök düğüm, oluşturmuş olduğu kapsama ağacı üzerinden, düğümlerin algılamış olduğu verileri toplayarak hedefin konumunu hesaplar ve ana düğüme konum bilgisini iletir. Daha sonra hedefin hızını ve yönünü hesaplayarak hedefin bir sonraki konumunu tahminler ve hedefin tahmini konumuna en yakın düğüme UYARI mesajı gönderir. Bu mesajı alan düğüm yeni kök düğüm olur ve kendi kapsama ağacını oluşturarak hedefi algılamaya hazır hale gelir. Şekil 5.1 literatürdeki dinamik kapsama ağacı tabanlı hedef izleme yaklaşımını genel olarak göstermektedir (Zhang and Cao, 2004).



Şekil 5.1. Genel Dinamik Kapsama Ağacı Tabanlı Hedef İzleme Yaklaşımı (GDCAIA)

Algoritma 5.1 Genel Dinamik Kapsama Ağacı Tabanlı Hedef İzleme Algoritmasının sözde kodunu göstermektedir.

Algoritma 5.1 i dugumu için *GDKAIA* Algoritmasının Sözde Kodu

```

1: Eger Ben bir normal dugum isem
2:   Eger Hedef algılanmış ise
3:     Bir  $kok_j$  dugumu sec
4:   Eger Ben_Liderim mesajı alınmış ise
5:     Uye dugum ol
6:   Eger Uyari mesajı alınmış ise
7:     Eger sayac sifirdan büyük ise
8:       Hedefin yonune en yakn komsu dugume Uyari mesajini gonder
9:     Eger sayac sifira esit ise
10:      kok dugum ol
11: Eger Ben bir uye dugum isem
12:   Eger Hedef algılanmış ise
13:      $kok_j$  dugumune Hedef_Bilgisi mesajini gonder
14:   Eger Uyari mesajı alınmış ise
15:     Eger sayac sifirdan büyük ise
16:       Hedefin yonune en yakn komsu dugume Uyari mesajı gonder
17: Eger Ben bir kok dugum isem
18:   Eger Hedef_Bilgisi mesajı alınmış ise
19:     Hedefin konumunu, hizini, yonunu hesapla
20:     Hedefin yonune en yakn dugume Uyari mesajı gonder

```

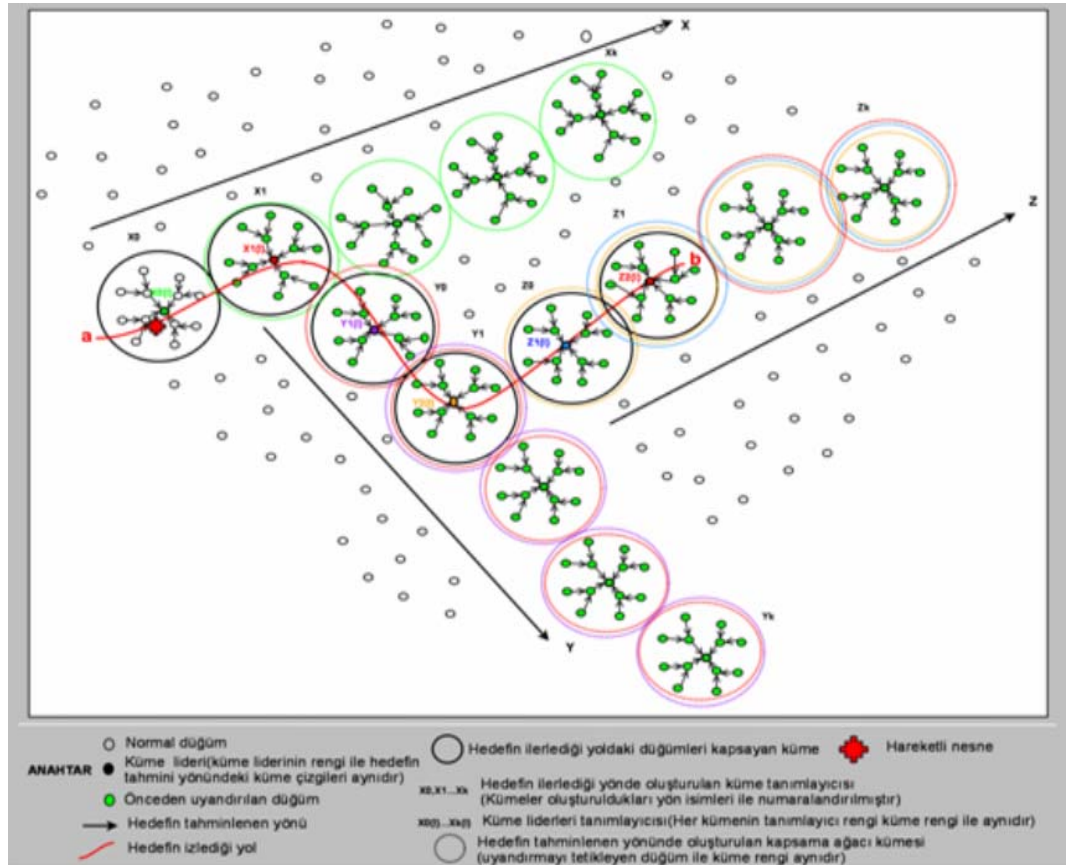
Algoritma 5.1. GDKAIA Algoritmasının Sözde Kodu

Tez kapsamında, literatürde var olan bu dinamik kapsama ağacı tabanlı hedef izleme yaklaşımından esinlenerek çok hızlı hareket eden hedefler için Çok Kümeli yeni bir versiyonunu önerdik. Önermiş olduğumuz DÇKAİA algoritmasında literatürdeki genel yaklaşımdan farklı olarak, hedefin yönü doğrultusunda, hedefin hızıyla ilişkili olarak önceden kapsama ağaçları oluşturulur ve bu önceden yapılandırılmış ağaçlar sayesinde, hedef algılandığı an konumu tespit edilebilmektedir. 4. bölümde önermiş olduğumuz DCKİA algoritmasından farklı olarak DÇKAİA algoritmasında her bir küme içerisindeki kapsama ağacında dinamik kök değişim işlemi gerçekleştirilmektedir. Dinamik kök değişimi ile amacımız, her zaman hedefe en yakın olan düğümün kök düğüm olmasını sağlamaktır. Böylece, kök düğüm hedefin yönünü daha doğru hesaplar ve hedefin kaybedilme riski daha da azalmış olur. Bir başka farklılık ise küme içerisinde bulunan düğüm sayısının DÇKAİA'da daha fazla olmasıdır ki bu da hedefi kaybetme olasılığını oldukça düşürmektedir. Bu iki önermiş olduğumuz algoritma arasındaki benzerlikler ise, önceden oluşturulacak kapsama ağacı sayısını

belirlemede kullanılan formül, kök düğüm seçme ve hedef konumlandırma algoritmaları olarak özetlenebilir. Tasarlamış ve uygulamasını gerçekleştirmiş olduğumuz algoritmanın detayları aşağıdaki alt bölümlerde verilmiştir.

5.1 Algoritmanın Tanımı

Bu bölümdeki amacımız, hedefin çok hızlı hareket ettiğini varsayarak ve tahminlenen yönü üzerinde, dinamik olarak, çok sayıda kapsama ağacı kümeleri oluşturmaktır. Böylece hedef, tahminlenen konumuna daha ulaşmadan, kapsama ağacı içeren kümeler hazır olacak ve düğümler de uyanık duruma geçerek hedefin gelmesini bekliyor olacaktır. Hedefin tahminlenen yönünde, önceden kaç adet kapsama ağacı kümesi oluşturulacağına, hedefin hızına bağlı olarak karar vermekteyiz. Hedefin hızı arttıkça önceden oluşturacağımız kapsama ağacı sayısı artacak, hız azaldıkça da bu sayı azalacaktır. Şekil 5.2.'de gerçekleştirilmek istenen algoritmanın genel işleyişi belirtilmiştir.



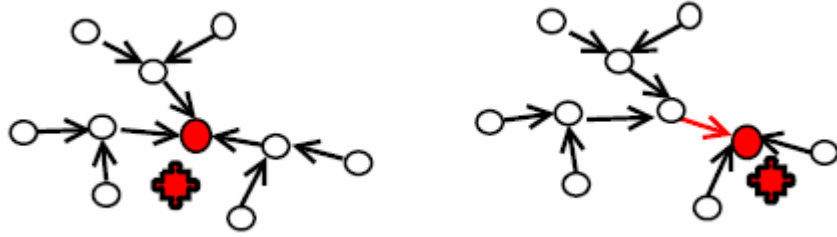
Şekil 5.2. DÇKAİA'nın Genel İşleyişi

Şekil 5.2.'den de görüldüğü gibi, hedef a noktasından alana girmekte ve b noktasına gitmeyi hedeflemektedir. Hedef a noktasından alana girdiğinde, hedefi algılayan düğümler birbirleri ile işbirliğine geçerek, $X_0(l)$ 'yi kök düğüm olarak seçerler. $X_0(l)$ kök düğümü, kendisinden n sıçrama mesafesinde olan düğümlerle kapsama ağacını oluşturur ve bu ağaç üzerinden düğümlerin hedefle ilgili algıladıkları veriyi toplayarak, hedefin konum, hız ve yön bilgisini tespit eder. Hedefin yönü doğrultusunda ve hızına bağlı olarak belirlemiş olduğu sayı kadar önceden kapsama ağaçlarını oluşturur. Şekil 5.2'de, her bir kök düğüm, ayrı bir renk ile belirtilmiştir ve bu düğümlerin önceden oluşturmuş oldukları kapsama ağacı kümeleri de aynı renkte çizilmiştir. Örneğin yeşil renkteki $X_0(l)$ kök düğümünün önceden oluşturmuş olduğu kapsama ağacı kümeleri de yeşil kesik çizgiler ile gösterilmiştir. Hedef, önceden oluşturulmuş kapsama ağacı kümesine her girişinde, ilgili kümedeki uyanık durumunda hedefi bekleyen düğümler, hedef ile ilgili topladıkları bilgileri kök düğümüne iletir. Her kök düğüm de, elde ettiği bilgiler doğrultusunda, hedefin yeni konumu ve yönünü hesaplayarak, hedefin yeni yönü doğrultusunda kapsama ağacı kümeleri oluşturur. Şekil 5.2.'de hedefin X_0 kümesindeki tahminlenen yönü ile X_1 kümesindeki tahminlenen yönü farklıdır. $X_1(l)$ kırmızı renkli kök düğümü, hedefin yeni yönü doğrultusunda bir doğru çizerek bu doğru üzerinde kırmızı kesik çizgili daireler ile belirtilen kapsama ağacı kümelerinin oluşturulmasını sağlar. $Y_1(l)$ kök düğümü $X_1(l)$ kök düğümü ile aynı hedef yönü tahmini yaptığı için, aynı yönde kapsama ağacı kümesi oluşturur ve oluşturduğu kapsama ağacı kümeleri de kendisi ile aynı renk olan mor renkte kesik çizgilerden oluşan daire ile belirtilmiştir. Bu işlemler hedef alanı terk edinceye kadar aynı şekilde devam eder. Böylece hedef aniden hızlansa bile önceden oluşturulmuş olan bu kapsama ağaçları sayesinde hedefin hareketi takip edilebilecektir.

Önermiş olduğumuz algoritmanın başka bir özelliği de kapsama ağacındaki kök düğümün dinamik olarak değişiyor olmasıdır. Hedef ilerledikçe, ağaç üzerinde hedefe yakın olan düğüm sürekli değişecektir. Önermiş olduğumuz yaklaşımda, her zaman, hedefe en yakın olan düğümün kök düğüm olmasını istediğimiz için, hedef hareket ettikçe kök düğüm de dinamik olarak değişmektedir. Böylece hedefin yönü daha doğru tahmin edilebilmektedir. Kapsama ağacındaki kök düğümün değişmesi, ağacın da yeniden yapılandırılmasını gerektirmektedir. Çalışmamızda, ağacın yeniden yapılandırma işlemini, jeton (token) tabanlı bir algoritma olan Raymond Algoritması'ndan[23] esinlenerek gerçekleştirmekteyiz.

Raymond Algoritmasına göre her bir düğümün bir ata düğümü vardır ve düğümler kendilerine gelen istek mesajlarını ilk giren ilk çıkar mantığı ile kuyruğa yerleştirirler. Düğümler bir anda kuyruktan sadece bir tek istek mesajını iletir. Algoritmada amaç jetonu almak isteyen düğüme jetonu iletmek ve jetonun iletilmesi esnasında ağağdaki yolların yeniden yapılandırılmasını sağlamaktır.

Şekil 5.3.'de, kök düğümün değişmesi ile üye düğümlerden köke giden yolun da dinamik olarak değişmesi görülmektedir.



Şekil 5.3. Dinamik Kök Değişimi ve Ağacın Yeniden Yapılandırılması

Algoritma 5.2. , Dinamik Çok Kümeli Kapsama Ağacı Tabanlı Hedef İzleme Algoritmasının sözde kodunu göstermektedir.

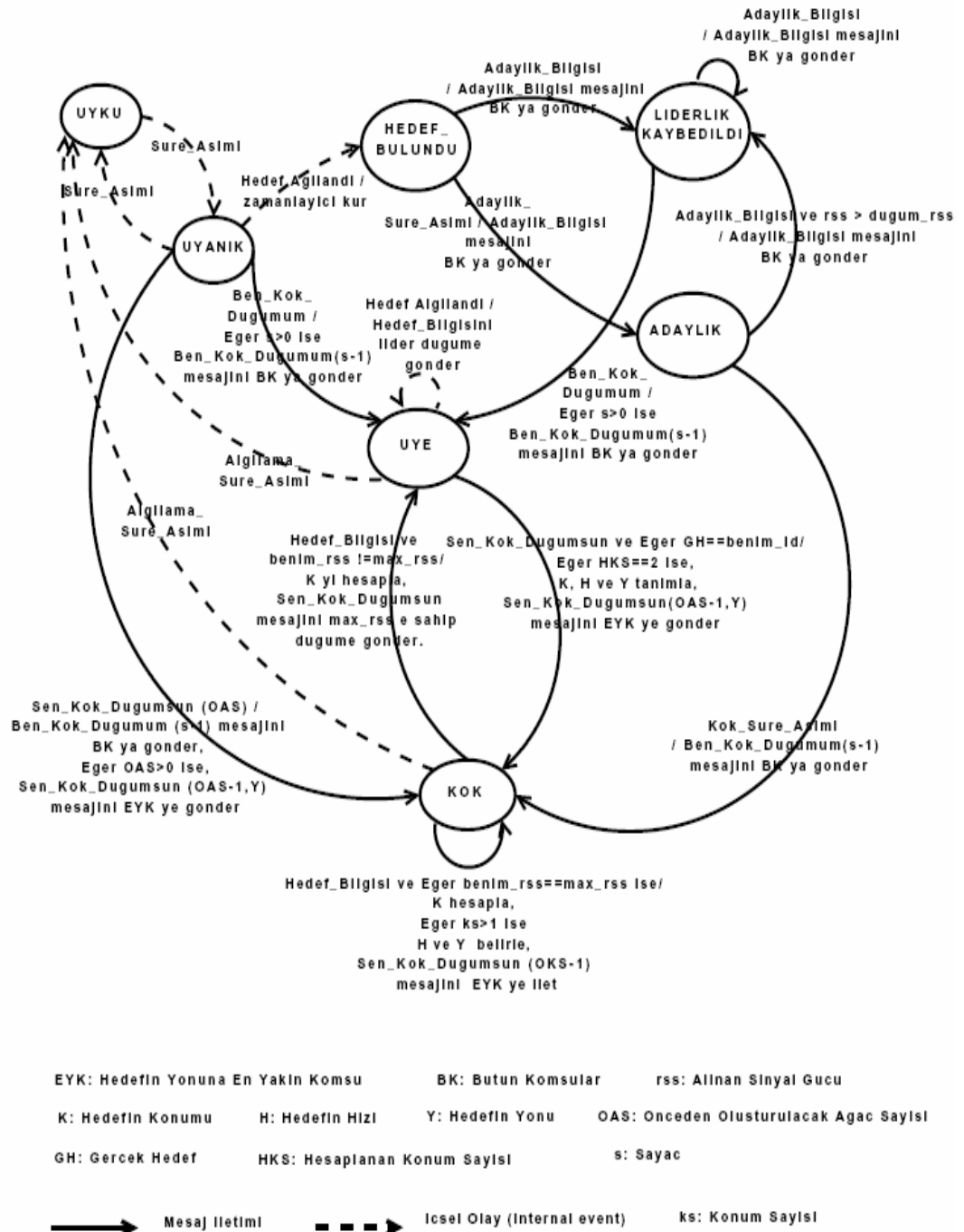
Algoritma 5.2 i düğümü için DCKAIA Algoritmasının Sözde Kodu

- 1: Eger Ben bir *normal* düğüm isem
 - 2: Eger Hedef algılanmış ise
 - 3: Bir kok_j düğümü seç
 - 4: Eger *Ben_Kok_Düğümüm* mesajı alınmış ise
 - 5: *uye* düğüm ol
 - 6: Eger *Sayac* sıfırdan büyük ise
 - 7: *Ben_Kok_Düğümüm* mesajını komşu düğümlere gönder
 - 8: Eger Ben bir *uye* düğüm isem
 - 9: Eger Hedef algılanmış ise
 - 10: kok_j düğümüne *Hedef_Bilgisi* mesajını gönder
 - 11: Eger *Sen_Kok_Düğümün* mesajı alınmış ise
 - 12: Eger *Sayac* sıfırdan büyük ise
 - 13: *Sen_Kok_Düğümün* mesajını hedefin yönüne en yakın düğüme gönder
 - 14: Eger *Sayac* sıfıra eşit ise
 - 15: *kok* düğüm ol
 - 16: Eger *Onceden_Olusturulacak_Kapsama_Agaci_Sayisi* sıfırdan büyük ise
 - 17: *Sen_Kok_Düğümün* mesajını hedefin yönüne en yakın komşu düğüme gönder
 - 18: Eger Ben bir *kok* düğüm isem
 - 19: Eger *Hedef_Bilgisi* mesajı alınmış ise
 - 20: Hedefin konumunu, yönünü, hızını hesapla
 - 21: Olusturulacak kapsama ağacı sayısını hesapla
 - 22: *Sen_Kok_Düğümün* mesajını hedefin yönüne en yakın düğüme gönder
-

Algoritma 5.2. DCKAIA Algoritmasının Sözde Kodu

5.2 Sonlu Durum Makinesi

Şekil 5.4., Dinamik Çok Kümeli Kapsama Ağacı Tabanlı Hedef İzleme Algoritmasının sonlu durum makinesini göstermektedir. Buna göre, bir düğüm, UYKU, UYANIK, HEDEF_BULUNDU, LİDERLİK_KAYBEDİLDİ, ADAY, ÜYE, KÖK durumlarından birinde olabilir ve ADAYLIK_BİLGİSİ, BEN_KÖK_DÜĞÜMÜM, SEN_KÖK_DÜĞÜMSÜN, HEDEF_BİLGİSİ mesajlarından birini alıp, gönderebilir. Bunların yanında belirli bir süre beklenildiğini ve bu bekleme süresi dolduğunda zaman aşımı olduğunu belirten içsel olaylar da SÜRE_AŞIMI, ADAYLIK_SÜRE_AŞIMI, LİDERLİK_SÜRE_AŞIMI, ALGILAMA_SÜRE_AŞIMI olarak ifade edilmiştir. Bu mesaj tiplerinden birini alması halinde, bir düğümün durumunda olacak değişiklik, Şekil 5.4.’teki sonlu durum makinesi üzerinde gösterilmiş ve devamında sonlu durum makinesinde kullanılan durum ve mesajların açıklamaları yapılmıştır.



Şekil 5.4. DÇKAİA'nın Sonlu Durum Makinesi

Şekil 5.4.'te gösterilen sonlu durum makinesinde belirtilen durumlardan özellikle ÜYE ve KÖK durumları arasında geçişlerin ve mesajlaşmaların yoğun olduğu gözlemlenmektedir. Bu geçişlerin en büyük nedeni kök düğümün dinamik olarak değişiyor olması ve hedefin yönü doğrultusunda önceden kapsama ağaçlarının oluşturuluyor olmasıdır.

ÜYE durumunda olan bir düğüm,

- Hedeften sinyal aldığı sürece kök düğümüne elde ettiği veriyi iletir.
- BEN_KÖK-DÜĞÜMÜM mesajı aldığı anda, eğer bu mesajı gönderen kök düğüm, kendi kök düğümü ise ve içindeki sayaç değeri sıfırdan büyükse, kök düğümünün ağaç oluşturma işlemi devam ediyor demektir ve sayaç değerini bir azaltarak mesajı komşu düğümlere iletmeye devam eder.
- SEN_KÖK_DÜĞÜMSÜN mesajı aldığı anda iki durum gerçekleşebilir. Eğer SEN_KÖK_DÜĞÜMSÜN mesajının içerisinde belirtilen gerçek hedef kendisi ise, bu dinamik kök değişiminin gerçekleştiği anlamına gelir ve düğüm ÜYE durumundan KÖK durumuna geçer. Bu geçiş esnasında ata düğüm (parent) tanımlamalarında da değişiklikler olur çünkü ağacın yeniden yapılandırılması gerekir. Ağacın yeniden yapılandırılması işlemi, Raymond Algoritmasından esinlenerek gerçekleştirilmektedir. SEN_KÖK_DÜĞÜMSÜN mesajı aynı zamanda önceden kapsama ağacı oluşturma işlemi için de kullanılmaktadır. Düğüm, kendisine bu mesaj geldiğinde hesaplanmış konum bilgisi sayısına ve önceden oluşturulacak kapsama ağacı sayısı bilgilerine bakar. Eğer hedefin hesaplanan konum sayısı ikiye eşitse ve önceden oluşturulacak kapsama ağacı sayısı sıfırdan büyük ise, düğüm, bu mesajın önceden kapsama ağacı oluşturulması amacıyla gönderildiğini anlar ve kapsama ağacı oluşturma işlemine geçer.
- ALGILAMA_SÜRE_AŞIMI içsel olayının gerçekleşmesi durumunda, düğümün hedeften belirli bir süre sinyal almadığını ve bu düğümün daha fazla uyanık kalmasına gerek olmadığını belirtir. Bu durumda düğüm, enerji korunumu için, ÜYE durumundan UYKU durumuna geçiş yapar.

KÖK durumunda olan bir düğüm,

- Kapsama ağacından topladığı sinyal gücü bilgilerinden, eğer kendisinden daha yüksek bir sinyal gücü değerine sahip düğüm varsa, kök düğümlüğü bırakır ve KÖK durumundan ÜYE durumuna geçiş yapar ve en yüksek sinyal gücü değerine sahip düğüme SEN_KÖK_DÜĞÜMSÜN mesajı gönderir.

- KÖK durumundaki bir düğüm, en az iki kere hedefin konumunu hesaplamışsa ve en yüksek sinyal gücü değerine sahip düğüm hala kendisi ise, hedefin konumunu, hızını ve yönünü hesaplayarak, hedefin yönüne en yakın komşu düğümüne SEN_KÖK_DÜĞÜMSÜN mesajı gönderir. Kök düğüm, hedefin hızına bağlı olarak önceden oluşturulacak kapsama ağacı sayısını belirler ve bu değeri SEN_KÖK_DÜĞÜMSÜN mesajının içerisine önceden_oluşturulacak_ağaç_sayısı parametresi olarak yerleştirir.
- Düğüm KÖK durumundayken, ALGILAMA_SÜRE_AŞIMI içsel olayının gerçekleşmesi durumunda, düğümün hedeften belirli bir süre sinyal almadığını ve bu düğümün daha fazla uyanık kalmasına gerek olmadığını belirtir. Bu durumda düğüm, enerji korunumu için, KÖK durumundan UYKU durumuna geçiş yapar.

Önermiş olduğumuz Dinamik Çok Kümeli Kapsama Ağacı Tabanlı Hedef İzleme Algoritmasının dağıtık açık kodu Algoritma 5.3. 'te verilmiştir.

Algoritma 5.3 $dugum_j$ için DCKAIA Algoritması Dağıtık Kodu

```

1: baslangic  $mevcut\_durum_j = UYANIK$ 
2:    $Lejant : \square DURUM \wedge gelen\_mesaj \rightarrow eylem$ 
3:    $S$ : Hedefin Hizi,  $K_t$ : Hedefin  $t$  anındaki konumu,  $Y$ : Hedefin Yonu
4:    $k_Y$ :  $Y$ 'ye en yakın komsu,  $\Gamma_j$ :  $j$ 'nin komşuları
5:    $OAS$ : Önceden oluşturulacak kapsama ağacı sayısı
6:    $s$ : sayac       $u$ : Ağaçlar arası uzaklık
7:    $gh$ : Gerçek hedef  $hks$ : Hesaplanan konum
8:
9:    $\square UYANIK \wedge Ben\_Kok\_Dugumum(i, s) \rightarrow kok_j \leftarrow id_i, mevcut\_durum_j \leftarrow UYE, Zamanlayici \text{ ata}$ 
10:      Eger  $s > 0$  ise  $Ben\_Kok\_Dugumum(i, s - 1)$ 
11:      mesajını  $\Gamma_j$  a ilet
12:       $\wedge Hedef\_Sinyal \rightarrow mevcut\_durum_j \leftarrow HEDEF\_BULUNDU, Zamanlayici \text{ ata}$ 
13:       $\wedge Sure\_Asimi \rightarrow mevcut\_durum_j \leftarrow UYKU, Zamanlayici \text{ ata}$ 
14:       $\wedge Sen\_Kok\_Dugumum(OAS, Y, gh, u) \rightarrow Eger gh <> benim\_id \wedge u > 0$  ise
15:           $Sen\_Kok\_Dugumum(OAS, Y, gh, u)$  mesajını  $k_{kid}$ 'e
16:          gonder,
17:          Eger  $gh <> benim\_id \wedge u > 0$  ise
18:           $Ben\_Kok\_Dugumum(i, s - 1)$  mesajını  $\Gamma_j$ 'a
19:          gonder,  $mevcut\_durum_j \leftarrow KOK, Zamanlayici \text{ ata}$ 
20:          Eger  $gh == benim\_id$  ise  $mevcut\_durum_j \leftarrow KOK,$ 
21:           $Zamanlayici \text{ ata}$ 
22:    $\square UYKU \wedge Sure\_Asimi \rightarrow kok_j \leftarrow id_i, mevcut\_durum_j \leftarrow UYANIK, Zamanlayici \text{ ata}$ 
23:    $\square UYE \wedge Ben\_Kok\_Dugumum(i, s) \rightarrow kok_j \leftarrow id_i, mevcut\_durum_j \leftarrow UYE, Zamanlayici \text{ ata}$ 
24:      Eger  $s > 0$  ise  $Ben\_Kok\_Dugumum(i, s - 1)$ 
25:      mesajını  $\Gamma_j$  a ilet
26:       $\wedge Hedef\_Sinyal \rightarrow HEDEF\_BILGISI$ 'ni  $kok_j$  dugumune gonder,  $Zamanlayici \text{ ata}$ 
27:       $\wedge Sure\_Asimi \rightarrow mevcut\_durum_j \leftarrow UYANIK, Zamanlayici \text{ ata}$ 
28:       $\wedge Sen\_Kok\_Dugumum(OAS, Y, gh, u) \rightarrow Eger gh <> benim\_id \wedge u > 0$  ise
29:           $Sen\_Kok\_Dugumum(OAS, Y, gh, u - 1)$  mesajını  $k_Y$ 'e
30:          gonder,
31:          Eger  $gh <> benim\_id \wedge u > 0$  ise
32:           $Ben\_Kok\_Dugumum(i, s - 1)$  mesajını  $\Gamma_j$ 'a
33:          gonder,  $mevcut\_durum_j \leftarrow KOK, Zamanlayici \text{ ata}$ 
34:          Eger  $gh == benim\_id$  ise  $mevcut\_durum_j \leftarrow KOK,$ 
35:           $Zamanlayici \text{ ata}$ 
36:    $\square KOK \wedge Hedef\_Bilgisi \rightarrow$  Hedefin  $t$  anındaki  $K_t$ 'yi hesapla
37:       $K_t$  ve  $K_{t-1}$  'den  $Y, S$  ve  $OAS$ 'yi hesapla,
38:       $Sen\_Kok\_Dugumum(OAS, Y, gh, u)$  mesajını  $k_Y$  gonder
39:       $\wedge Sure\_Asimi \rightarrow mevcut\_durum_j \leftarrow UYANIK, Zamanlayici \text{ ata}$ 
40:       $\wedge Ben\_Kok\_Dugumum(i, s) \rightarrow mevcut\_durum_j \leftarrow UYE, Zamanlayici \text{ ata}, kok_j \leftarrow id_i$ 
41:    $\square HEDEF\_BULUNDU \wedge Adaylik\_Bilgisi(i) \rightarrow Adaylik\_Bilgisi(i)$ 'ni  $\Gamma_j$  'a ilet,
42:       $mevcut\_durum_j \leftarrow LIDERLIK\_KAYBEDILDI$ 
43:       $\wedge Sure\_Asimi \rightarrow \Gamma_j$ 'ye  $Adaylik\_Bilgisi$  gonder,  $Zamanlayici \text{ ata}, mevcut\_durum_j \leftarrow ADAY$ 
44:    $\square LIDERLIK\_KAYBEDILDI \wedge Adaylik\_Bilgisi(i) \rightarrow Adaylik\_Bilgisi(i)$  'ni  $\Gamma_j$ 'ya ilet
45:       $\wedge Ben\_Kok\_Dugumum(i, s) \rightarrow mevcut\_durum_j \leftarrow UYE, kok_j \leftarrow id_i,$ 
46:    $\square ADAY \wedge Adaylik\_Bilgisi(i, RSS) \rightarrow Eger RSS > kendi\_RSS$  ise
47:       $Adaylik\_Bilgisi(i, RSS)$ 'ni  $\Gamma_j$ 'e ilet,
48:       $mevcut\_durum_j \leftarrow LIDERLIK\_KAYBEDILDI$ 
49:

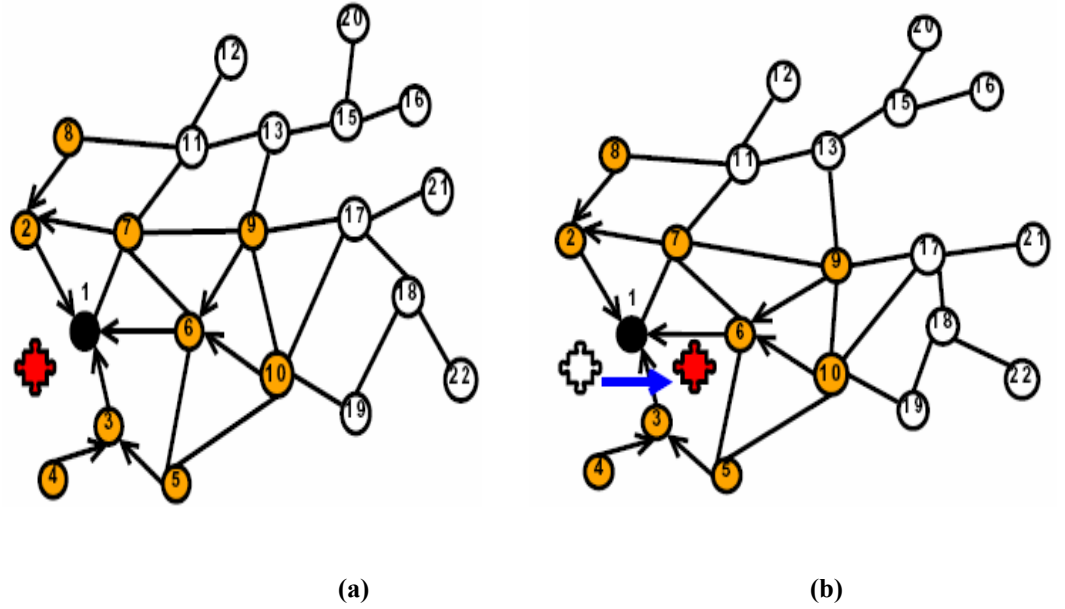
```

Algoritma 5.3. DCKAIA Algoritmasının Dağıtık Açık Kodu

5.3 Örnek Bir işlem

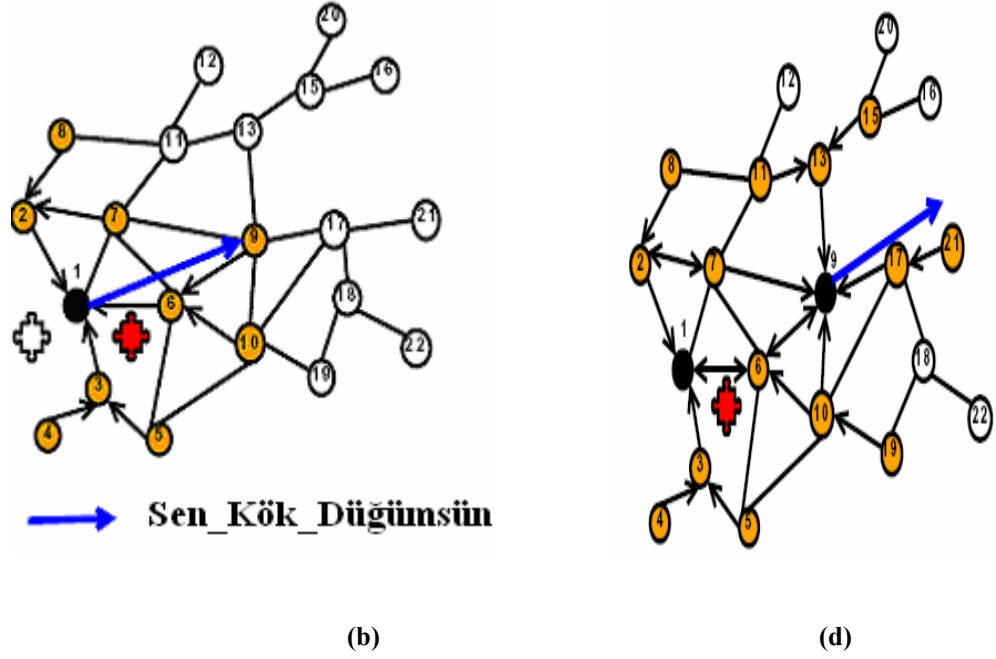
Şekil 5.5., Dinamik Çok Kümeli Kapsama Ağacı Tabanlı Hedef İzleme Algoritmasının bir örnek üzerinden çalışmasını gösterilmektedir.

Şekil 5.5 (a)'da ilk başlangıç ağacının oluşturulmuş hali bulunmaktadır. 1 numaralı düğüm kök düğüm ve sarı renk ile belirtilen düğümler de kapsama ağacı üzerinde bulunan düğümlerdir. 1 numaralı kök düğüm, kapsama ağacı üzerinden elde ettiği verilerle hedefin konumunu, hızını ve yönünü tespit etmektedir. (b)'de 1 numaralı düğümün iki konum bilgisinden hedefin yönünü tespit edişi gösterilmektedir.



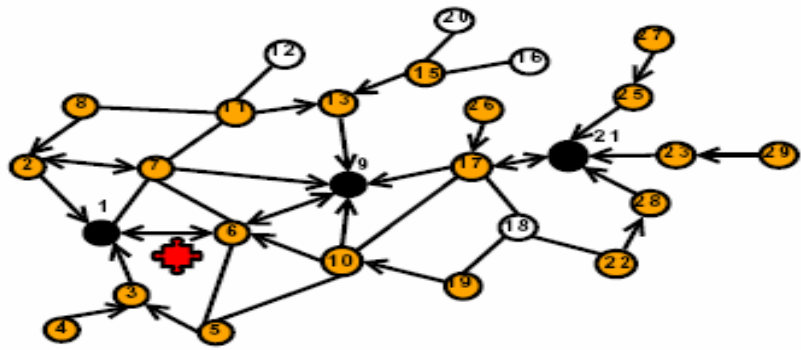
Şekil 5.5. DÇKAİA'da Kök Düğümün Hedefin Yönünü Belirlemesi

Hedefin yönü tespit edildikten sonra, kök düğüm kendisinden iki sıçrama mesafesindeki düğüme kapsama ağacı oluşturmasını sağlayacak şekilde, hedefin yönüne en yakın yönde bulunan komşusuna SEN_KÖK_DÜĞÜMSÜN (Hedefin_Yönü, Önceden_Oluşturulacak_Kapsama_Ağacı_Sayısı) mesajını iletir. Şekil 5.6 (a)'da 1 numaralı düğüm öncelikle hedefin yönüne en yakın düğüm olan 6 numaralı düğüme, 6 numaralı düğüm de hedefin yönüne en yakın düğüm olan 9 numaralı düğüme SEN_KÖK_DÜĞÜMSÜN mesajını iletir. 9 numaralı düğüm yeni kök düğüm olur ve kendisinden iki sıçrama mesafesindeki düğümlerle kapsama ağacını oluşturur.



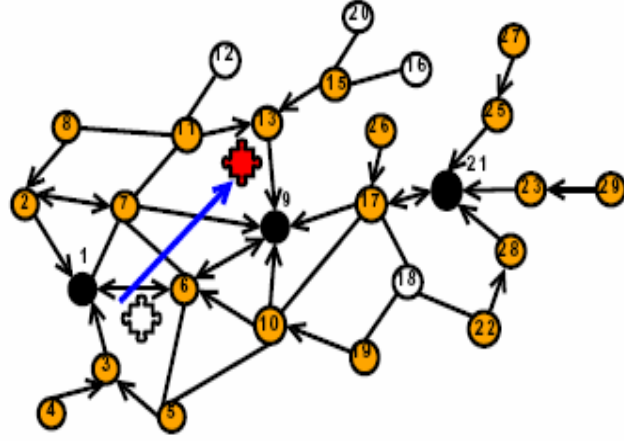
Şekil 5.6. DÇKAİA' da Önceden Ağaçların Oluşturulması

Şekil 5.6 (b)'de, 9 numaralı düğüm kendisine gelen SEN_KÖK_DÜĞÜMSÜN mesajının içerisindeki Önceden_Oluşturulacak_Kapsama_Ağacı sayısını bir azaltır ve bu değer sıfırdan büyükse, mesajın içerisinde belirtilen yön doğrultusunda mesajı iletmeye devam eder. Şekil 5.6 (b)'de de görüldüğü üzere, 9 numaralı lider düğüm, Önceden_Oluşturulacak_Kapsama_Ağacı değeri sıfırdan büyük olduğu için hedefin yönüne en yakın komşusu olan 17 numaralı düğüme, 17 numaralı düğüm de 21 numaralı düğüme mesajı iletmektedir.

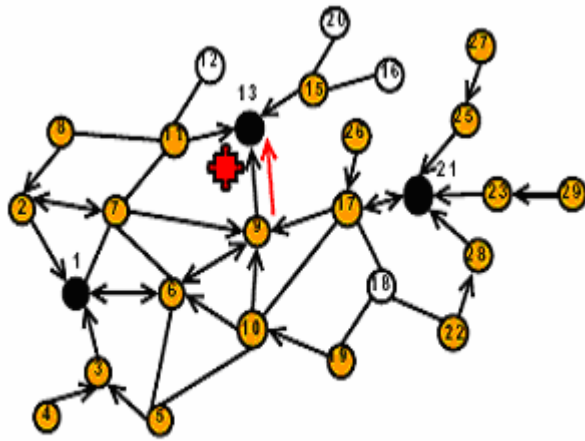


Şekil 5.7. DÇKAİA' da Önceden Ağaçlar Hazırlanmış Durumda

Şekil 5.7’de 1 numaralı mevcut kök düğümün tetiklemesiyle oluşturulmuş olan, 9 ve 21 numaralı kök düğümlere sahip kapsama ağaçlarının hazır halde hedefi bekledikleri görülmektedir.



(a)



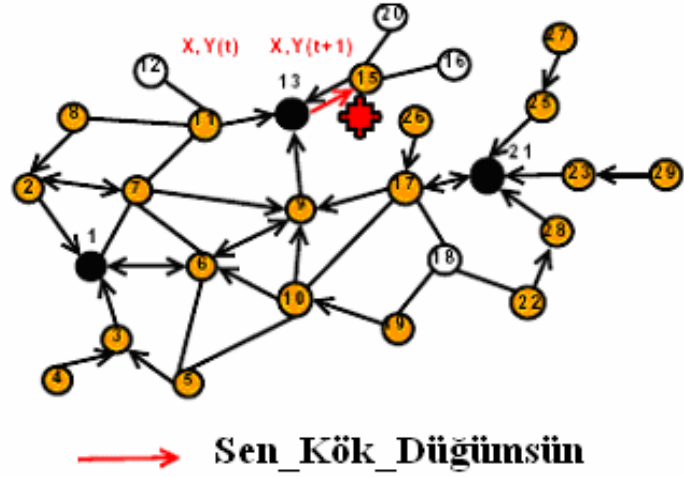
→ Sen_Kök_Düğümün

(b)

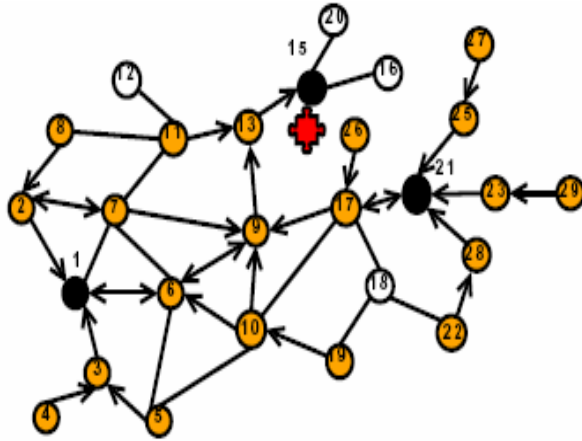
Şekil 5.8. DÇKAİA’da Dinamik Kök Değişimi

Şekil 5.8 (a)’da hedef ilerledikçe, 9 numaralı önceden oluşturmuş olduğumuz kapsama ağaçlarının kök düğümü, elde ettiği sinyal gücü değerlerinden, hedefin 13 numaralı düğüme daha yakın olduğunu fark eder. Dinamik olarak kök düğüm değişim işlemi gerçekleşir ve yeni kök düğüm artık 13 numaralı düğüm olur. Eski kök düğüm olan 9 numaralı düğümün ağacındaki düğümler, algıladıkları veriyi 9

numaralı düğüm üzerinden, yeni kök düğüm 13'e iletirler. Şekil 5.8 (b)'de dinamik kök düğüm ve yol değişim işlemleri gösterilmektedir.



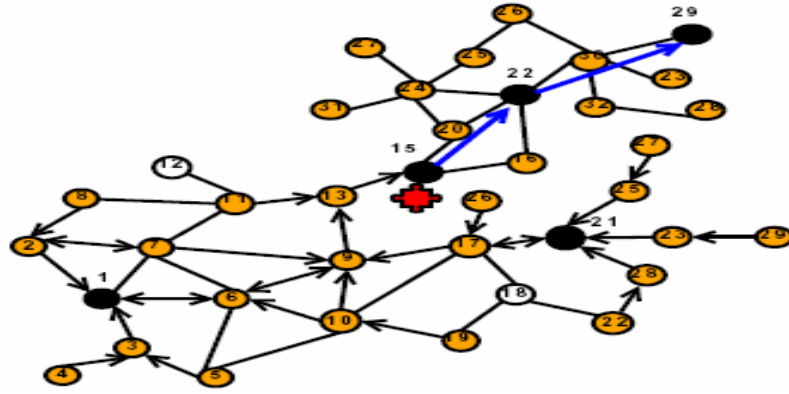
(a)



(b)

Şekil 5.9. DÇKAİA'da Hedefin Yeni Yönünün Belirlenmesi

13 numaralı kök düğüm, ağacındaki düğümlerden, algıladıkları veriyi topladığında, en yüksek sinyal gücü değerinin 15 numaralı düğümde olduğunu fark eder ve tekrar kök düğüm değişim işlemi gerçekleşir. Şekil 5.9 (a) ve (b) 'de 15 numaralı düğümün yeni kök düğüm olması ve düğümler arasındaki yolun yeniden yapılandırılması işlemi gösterilmektedir. Şekilden de görüldüğü gibi hedefin yön değişimi önceden oluşturmuş olduğumuz 9 numaralı kök düğümün kapsama ağacı tarafından tespit edilmiştir ve böylece önceden oluşturulan ağaç sayesinde hedefin kaybedilmesi gibi bir durum önlenmiştir.



Şekil 5.10. DÇKAİA'da Hedefin Yeni Yönü Doğrultusunda Ağaçların Oluşturulması

Eski kök düğümler olan 9 ve 13 numaralı düğümler, ağaçtan her veri topladığında maksimum sinyal gücü değerinin hangi düğümde olduğunu belirlerken, aynı zamanda hedefin konumlarını da hesaplayıp, bu bilgiyi de SEN_KÖK_DÜĞÜMSÜN mesajının içerisinde göndermektedirler. Dolayısıyla 15 numaralı düğüm SEN_KÖK_DÜĞÜMSÜN mesajını aldığında elinde iki konum bilgisi bulunmaktadır. Bu konum bilgilerini kullanarak, yeni kök düğüm olan 15 numaralı düğüm, hedefin hızını ve yönünü hesaplayarak, yeni yön doğrultusunda kapsama ağaçlarını oluşturma işlemini başlatır. Şekil 5.10'da 15 numaralı kök düğümün önceden ağaç oluşturma işlemi gösterilmektedir.

5.4 DÇKAİA Algoritmasının Mesaj Karmaşıklık Analizi

Bu bölümde, DÇKAİA algoritmasının her bir aşamasındaki kullandığı mesaj sayısı adım adım hesaplanarak, toplam mesaj karmaşıklığı belirlenmiştir.

Teorem 5.4.1: DÇKAIA algoritmasının mesaj karmaşıklığı $O(n)$ 'dir.

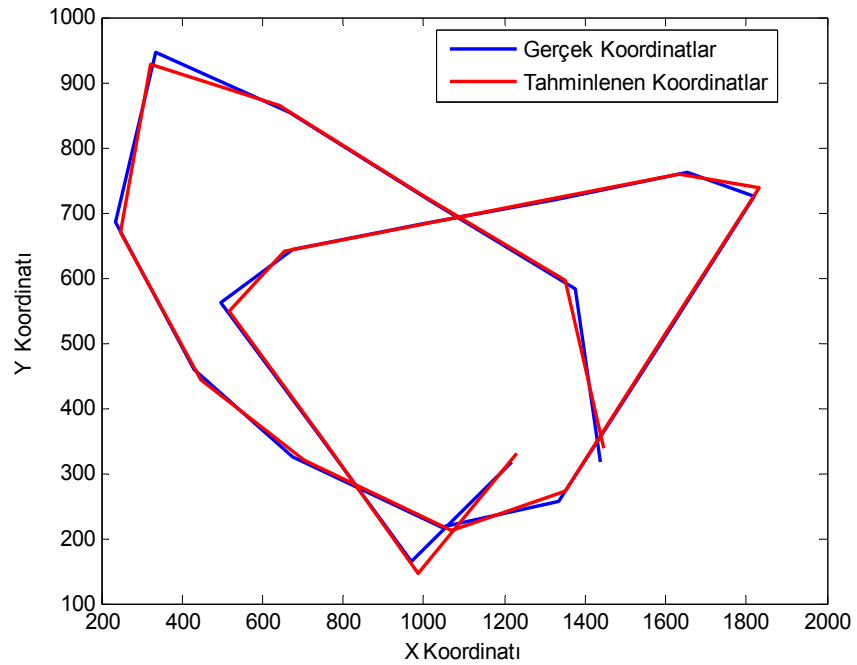
İspat:

- **Kök düğümü belirleme aşamasında**, bir düğümün ortalama komşu sayısı k ve hedefi algılayan düğüm sayısı a için bu aşamadaki mesaj sayısı: $k*a$ 'dır.
- **İlk kapsama ağacının oluşturulması aşamasında**, seçilen kök düğüm, k adet komşu düğümüne BEN_KÖK_DÜĞÜMÜM mesajı gönderir. Bu algorithmada kullanmış olduğumuz ağaçların derinliği 2 olduğu için, bu aşamadaki mesaj sayısı: $k+k*k$ 'dır.
- **Üye düğümlerin elde ettikleri verileri kök düğüme iletmesi aşamasında**, kapsama ağacı içerisindeki $k+k*k$ adet üye düğüm iki farklı zamanda elde ettikleri veriyi lider düğüme iletir. Bu aşamadaki mesaj sayısı: $2*(k+k*k)$ 'dır.
- **Kök düğümün hedefin yönü doğrultusunda ağaç oluşturma mesajı göndermesi aşamasında**, SEN_KÖK_DÜĞÜMSÜN mesajı, hedefin yönüne en yakın düğüme sıçrama sayısı kadar (h) iletilir. Bu durumda tek bir mesaj h kadar iletilmiş olur. Mesaj sayısı: h 'dır.
- **Önceden ağaçların oluşturulması aşamasında**, her bir önceden oluşturulan kök düğüm ağacını oluşturabilmek için k adet komşu düğümüne BEN_KÖK_DÜĞÜMÜM mesajı gönderir ve sıçrama sayısı (h) kadar kapsama ağacı oluşturulacağı için bu aşamadaki mesaj sayısı: $h*(k+k*k)$ 'dır.
- Ortamda n adet düğümün olduğunu düşünürsek, simülasyon süresi boyunca ortamda $n/(1+k+k*k)$ adet ağaç oluşturulabilir. $(1+k+k*k)$, bir ağaçta bulunabilecek düğüm sayısını belirtmektedir.

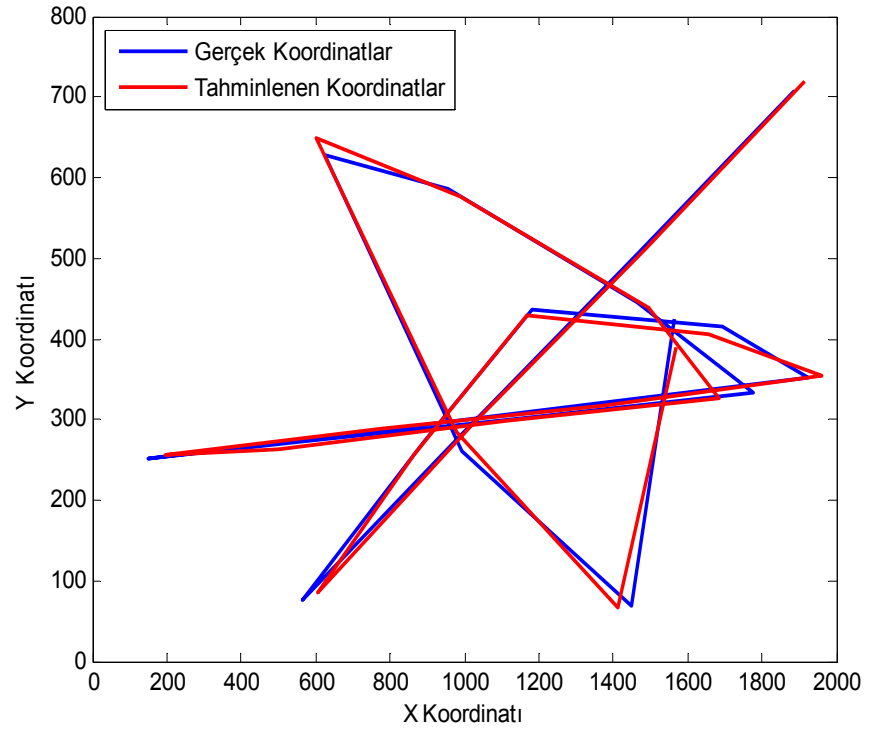
Sonuç olarak, yukarıda belirtilen mesaj sayıları toplandığında, DÇKAIA'nın hedefi takip edebilmesi için kullandığı toplam mesaj sayısı $(k*a+3*(k+k*k)+h+h*(k+k*k))* (n/(1+(k+k*k)))$ olarak hesaplanır ki bu da mesaj karmaşıklığının $O(n)$ olduğu anlamına gelir.

5.5 Simülasyon Sonuçları

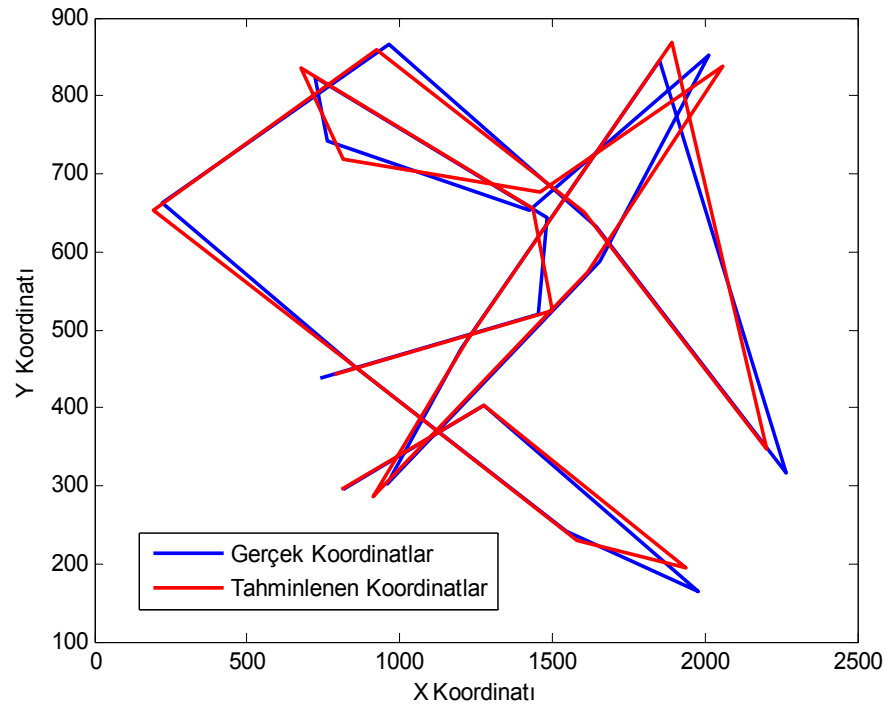
Önermiş olduğumuz Dinamik Çok Kümeli Kapsama Ağacı Tabanlı Hedef İzleme Algoritmasını, çok hızlı hareket eden nesneler için uygulayıp, hedef takibi ve hedefi kaybetme oranları için simülasyon sonuçları elde edilmiştir. Sonuçlar, 70'er deneyin ortalaması alınarak elde edilmiştir. Şekil 5.11.'de çok hızlı hareket eden hedefin, 30 m/s ile 100 m/s arasında değişen hız değerleri için takibi gösterilmektedir. Hız değerleri arttıkça gerçek rota ile hesaplanan rota arasındaki farkta artış olsa da, hedefin çok hızlı hareket ettiği durumlarda rota da çok büyük bir sapma olmamıştır.



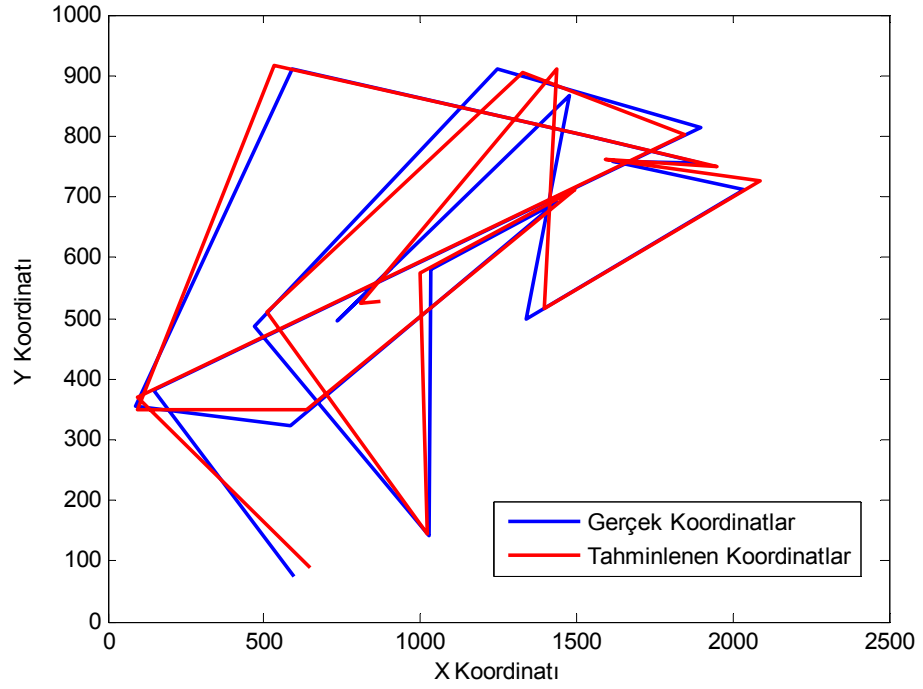
(a) 30-40 m/s



(b) 50-60 m/s



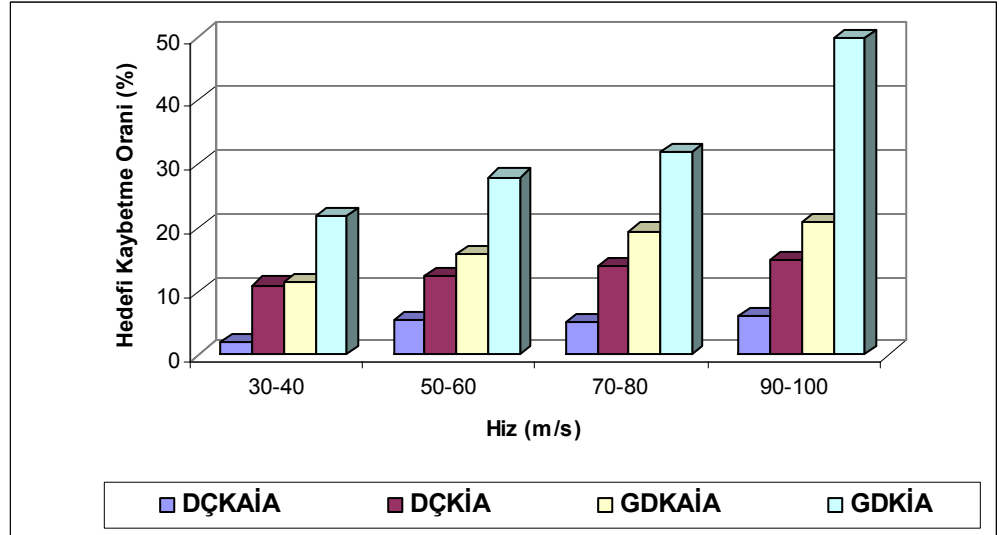
(c) 70-80 m/s



(d) 90-100 m/s

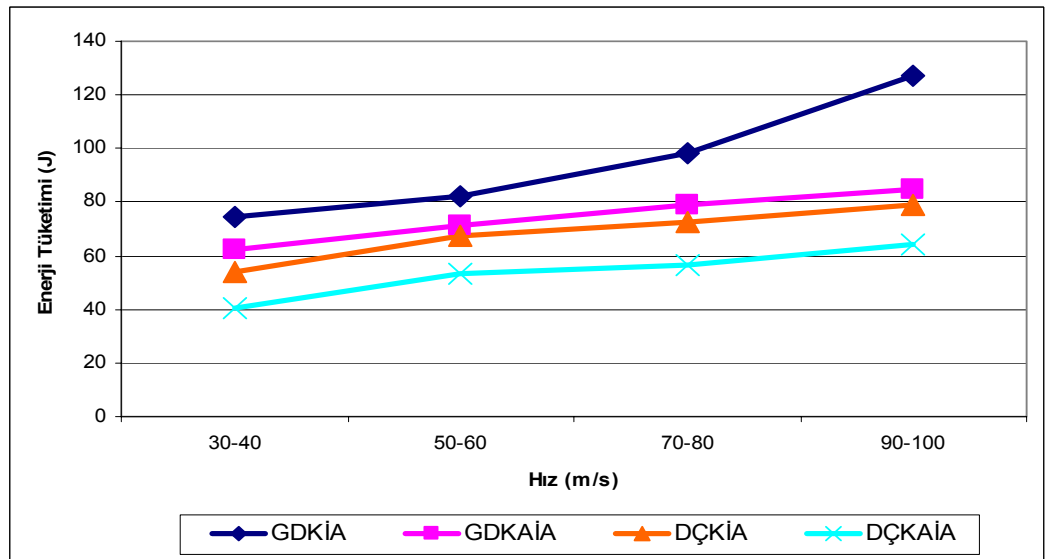
Şekil 5.11. DÇKAİA ile Hedef Takibi

Önermiş olduğumuz algoritma ile hedefi kaybetme olasılığını azaltmayı hedeflemiş bulunmaktayız. Şekil 5.12.'de, önerilen DÇKAİA ve DÇKİA Algoritmaları ile mevcut yaklaşımlar olan GDKAİA ve GDKİA Algoritmaları hedefi kaybetme oranları için karşılaştırılmıştır. Buna göre, her bir hız değeri için önerilen yaklaşımlar mevcut yaklaşıma göre hedefi kaybetme olasılığı açısından daha iyi sonuç verdiğini görmekteyiz. Bunun yanında hız değeri arttıkça, önerilen yaklaşımlar ile genel yaklaşımlar arasındaki hedefi kaybetme oranları arasındaki farkın daha da arttığı görülebilmektedir.



Şekil 5.12. GDKAİA, GDKİA, DÇKİA ve DÇKAİA Algoritmalarının Hedefi Kaybetme Olasılığı için Karşılaştırılması

Şekil 5.12.'de önermiş olduğumuz iki yaklaşım olan DÇKİA ile DÇKAİA Algoritmaları da hedefi kaybetme oranları için karşılaştırılmıştır. Simülasyon sonuçlarından, DÇKAİA Algoritmasının DÇKİA Algoritmasına göre daha iyi sonuç verdiği görülmektedir. Kapsama ağacında bulunan düğüm sayısının küme tabanlı yaklaşıma göre daha fazla olması nedeniyle, DÇKAİA'da hedefi kaybetme oranı daha düşüktür.



Şekil 5.13. GDKAİA, GDKİA, DÇKİA ve DÇKAİA Algoritmalarının Enerji Tüketimleri için Karşılaştırılması

Şekil 5.13.'te, önerilen DÇKAİA ve DÇKİA Algoritmaları ile mevcut yaklaşımlar olan GDKAİA ve GDKİA Algoritmaları, enerji tüketimleri için karşılaştırılmıştır. Enerji tüketimleri, simülasyon süresi sonunda her bir düğümde kalan enerji miktarı toplanarak hesaplanmaktadır. Şekil 5.13'ten de görüldüğü gibi en yüksek enerji tüketimi mevcut yaklaşım olan GDKİA'da elde edilirken, en düşük enerji tüketimi önermiş olduğumuz DÇKAİA algoritmasında elde edilmiştir. Önermiş olduğumuz yaklaşımlarda önceden oluşturduğumuz kümeler ve kapsama ağaçları sayesinde hedefin kaybedilme oranı oldukça düşmektedir. Hedefin kaybedilme oranının düşmesi ile, düzeltme mekanizmasına olan ihtiyaç da düşmekte ve bu da enerji tüketimini azaltarak ağın genel yaşam süresini arttırmaktadır.

6 DOĞRUSAL HAREKETLER İÇİN DİNAMİK KONİ TABANLI HEDEF İZLEME ALGORİTMASI (DKOİA)

Şu ana kadar hedef takibi için küme ve kapsama ağacı tabanlı olmak üzere iki algoritma önermiş bulunmaktayız. Simülasyon sonuçlarından elde ettiğimiz verilere göre kapsama ağacı tabanlı yaklaşım, küme tabanlı yaklaşıma göre çok daha iyi sonuç vermektedir. Bu sonuçlardan yola çıkarak tez kapsamında, literatürde daha önce rastlamadığımız, “Koni Tabanlı Hedef İzleme” isimli yeni bir algoritmasını öneriyoruz.

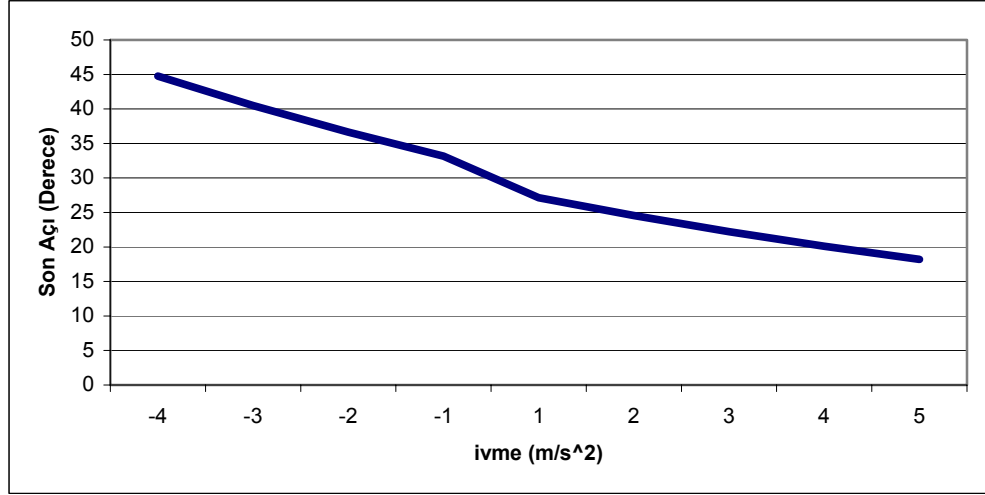
6.1 Algoritmanın Tanımı

Bu bölümdeki amacımız, hedefin hesaplanan yönünden α açısı kadar sağa ve sola doğrular çizerek dinamik bir koni oluşturmak ve bu koni içerisinde kapsama ağacı kümeleri yapılandırarak, minimum kaybetme riski ile hedefi takip etmektir. Böylece hedef, tahminlenen konumuna daha ulaşmadan, kapsama ağacı içeren kümeler hazır olacak ve düğümler de uyanık duruma geçerek hedefin gelmesini bekliyor olacaktır. Oluşturulacak olan koninin α açısı ve koni boyunca oluşturulacak ağaç sayısı, hedefin hızına bağlı olarak belirlenecektir. Hedefin hızı arttıkça koninin α açısı küçülecek fakat boyu uzayacak, hız azaldıkça da α açısı büyüyecek ve boyu kısalmaktadır. Böylece hedefin hız durumuna göre koninin boyutu dinamik olarak değişim gösterecektir.

Koni açısını belirlemek için kullandığımız formül:

$$\alpha_{t+1} = \alpha_t * e^{(-a/k)} \quad (1)$$

(1)’de kullanılan a değeri ivmeyi, k değeri de sabit bir sayıyı, 10, ifade etmektedir. Formüle göre hedef hızlandıkça, koninin açısı daralmakta, yavaşladıkça da açı genişlemektedir. İvmenin sıfır olduğu, sabit hızla hareket eden hedef takibinde, her bir hız değeri için farklı α_t değerleri kullanılmıştır. α_t , yüksek hızlar için düşük, düşük hızlar için de yüksek bir değer içermektedir. Şekil 6.1 ivme ile α_{t+1} arasındaki ilişkiyi göstermektedir. Şekilden de görüldüğü hedef hızlandıkça koni daralmakta, yavaşladıkça da koni genişlemektedir.



Şekil 6.1. İvme ve Açılış İlişkisi

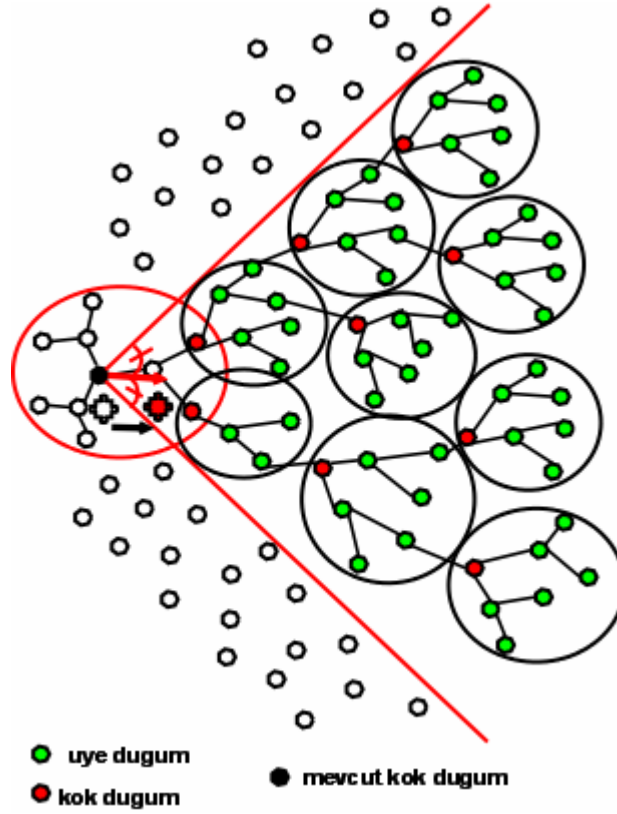
Koni boyunu belirlemek için kullandığımız formül:

$$\text{Sabit Hızlı Hareket İçin} \rightarrow L = (V \cdot t) / d \quad (2)$$

$$\text{İvmeli Hareket İçin} \rightarrow L = (V \cdot t + 0.5 \cdot a \cdot t^2) / d$$

(2)'e göre hedef hızlandıkça, koninin boyu uzayacak, yavaşladıkça da kısılacaktır. Burada koni boyu olarak bahsettiğimiz, mevcut kök düğümünden kaç sıçrama mesafesine kadar ağaç oluşturma işleminin devam edeceği anlamındadır. Bu yüzden formülde öncelikle hedefin belirli bir süre içinde ne kadar mesafe gideceği hesaplanmış, daha sonra da düğümler arasındaki ortalama mesafe değerine bölünerek sıçrama sayısına ulaşılmıştır. Formülde kullanılan V değeri hedefin hızını, t değeri süreyi, d değeri düğümler arasındaki ortalama mesafeyi, a değeri de ivmeyi ifade etmektedir. Hedefin sabit ve ivmeli hareketine göre aynı süre içerisinde aldıkları mesafe farklı olacağı için, bu iki hareket modeli için farklı formüller kullanılmıştır.

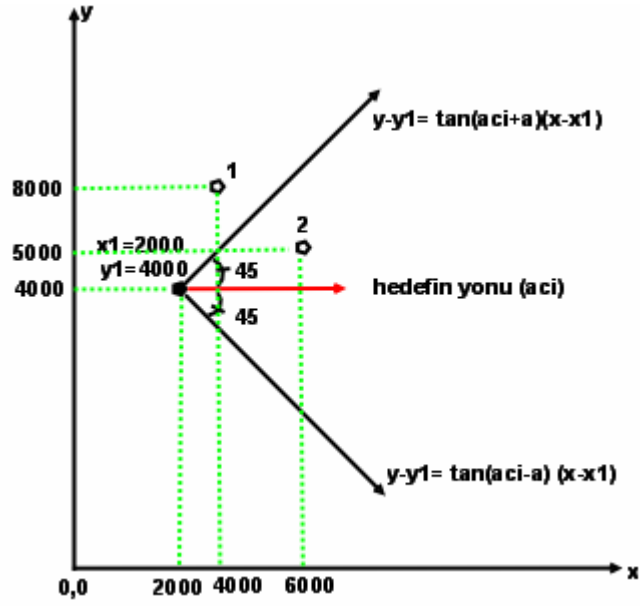
Şekil 6.2.'de gerçekleştirilmek istenen algoritmanın işleyişi belirtilmiştir.



Şekil 6.2. DKOİA'nın Genel İşleyişi

Şekil 6.2.'den de görüldüğü gibi, mevcut kök düğüm, kapsama ağacı üzerindeki hedefi algılayan düğümlerden aldığı bilgilerle hedefin konumu, yönünü ve hızını tespit eder. Hesaplamış olduğu konum bilgisini öncelikle, akış tabanlı dağıtık kapsama ağacı üzerinden ana düğüme iletir. Daha sonra, hedefin hesaplamış olduğu yön ve hıza göre konisini oluşturur. Şekilde de görüldüğü gibi kök düğüm hedefin sağa doğru gittiğini belirlemiş ve bu yönden sağa ve sola doğru açılar yapacak şekilde konisinin üst ve alt doğrularını oluşturarak kapsama ağaçlarının yapılandırılması işlemini başlatmıştır.

Şekil 6.3'te, mevcut kök düğümün, koni sınırlarını belirleme ve koni içinde kalan komşu düğümlerini tespit etme işlemleri gösterilmektedir.



Şekil 6.3. Koni Çizme ve Koni İçinde Kalan Dğümleri Belirleme

Şekilden de görüldüğü gibi:

- (x_1, y_1) koordinatlarındaki mevcut kök düğüm, hedefin yönünü tespit eder ve (1)'deki formül ile hesaplanan α açısı kadar sağa ve sola doğru açılar oluşturacak şekilde doğru denklemleri belirler.
- Başlangıç noktası (mevcut kök düğümün koordinatları), eğim($\tan(\text{aci} + \alpha)$) ve eğim($\tan(\text{aci} - \alpha)$) bilgilerinden üst ve alt doğruların denklemleri bulunur. Burada aci ile hedefin hesaplanan yön açısı kast edilmektedir.
- x_1, y_1 koordinatlarına sahip kök düğüm, x ve y komşu düğümlerinin koordinatları olmak üzere,

$$y \leq \tan(\text{aci} + \alpha)(x - x_1) + y_1 \quad \text{ve} \quad y \geq \tan(\text{aci} - \alpha)(x - x_1) + y_1$$

şartlarını sağlayan komşu düğümlerine SEN_KÖK_DÜĞÜMSÜN mesajı gönderir. Örneğin Şekil 6.3'te kök düğüm, 1 numaralı komşusunun koordinatlarını eşitliğe yerleştirir ve şartı sağlamadığı için bu düğüme mesaj göndermez. Bunun yanında 2 numaralı komşu düğümü şartı sağladığı için bu komşu düğümüne mesaj gönderir.

1 numaralı düğüm için;

1 numaralı düğümün koordinatları $x=4000$ $y=8000$

$8000-2000 \leq 1 \cdot (4000-2000) \rightarrow$ şart sağlanmadığı için 1 numaralı düğüm koni dışındadır. Bu yüzden, kök düğüm, 1 numaralı düğüme mesaj göndermez.

2 numaralı düğüm için;

2 numaralı düğümün koordinatları $x=6000$ $y=5000$

$5000-2000 \leq 1 \cdot (6000-2000) \rightarrow$ sağlıyor $5000-2000 \geq -1 \cdot (6000-2000) \rightarrow$ sağlıyor

2 numaralı düğüm, iki şartı da sağladığı için kök düğüm, 2 numaralı komşusunun koni içerisinde olduğunu anlar ve bu düğüme SEN_KÖK_DÜĞÜMSÜN mesajı gönderir.

Algoritma 6.1 'de önerilen Dinamik Koni Tabanlı Hedef İzleme Algoritmasının sözde kodu gösterilmektedir.

Algoritma 6.1 : dugumu için *DKOIA* Algoritmasının Sözde Kodu

```

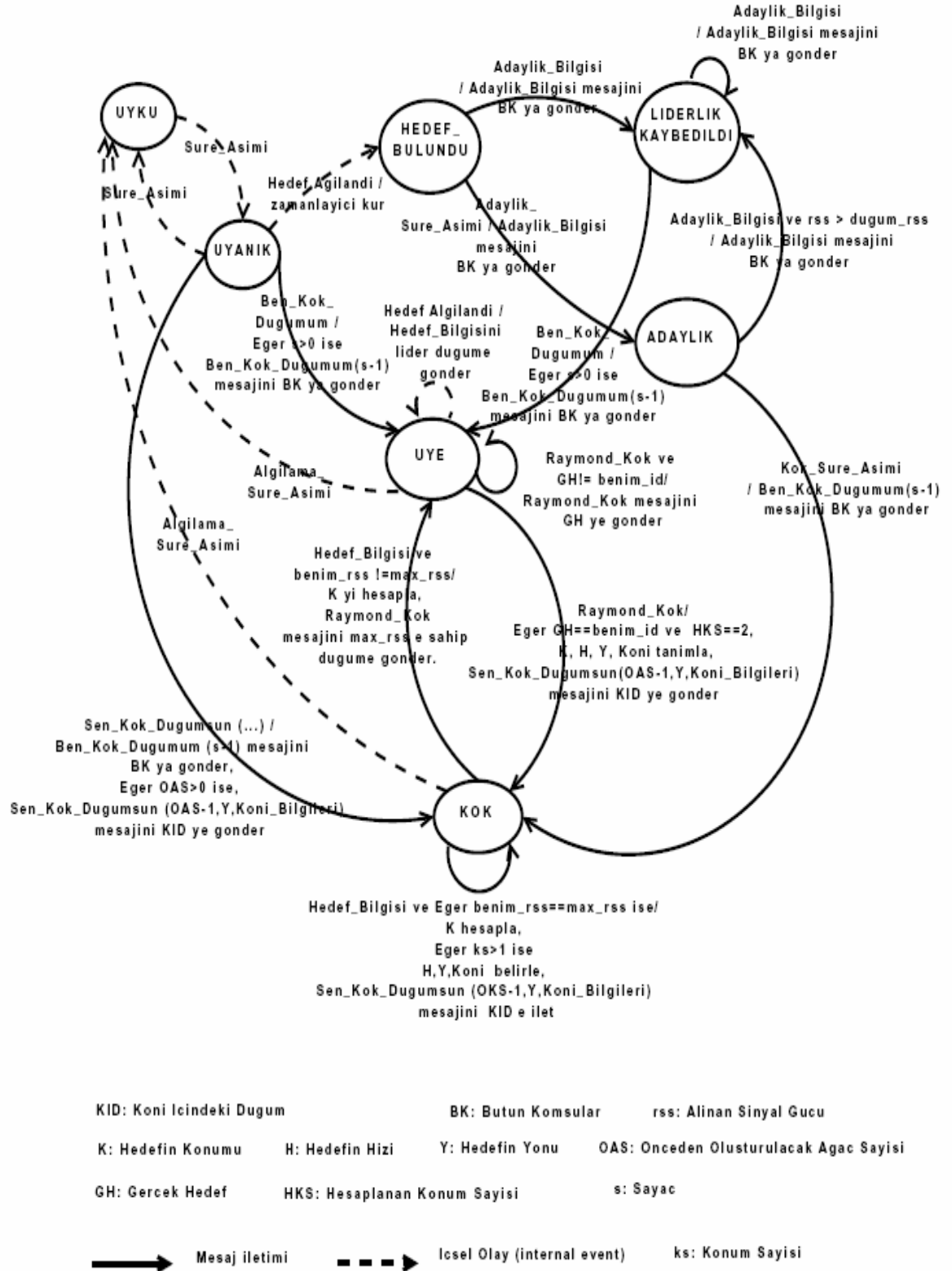
1: Eger Ben bir normal dugum isem
2:   Eger Hedef algılanmış ise
3:     Bir kokj dugumu sec
4:   Eger Ben_Kok_Dugumum mesajı alınmış ise
5:     uye dugum ol
6:   Eger Sayac sifirdan büyük ise
7:     Ben_Kok_Dugumum mesajını komşu dugumlere gonder
8: Eger Ben bir uye dugum isem
9:   Eger Hedef algılanmış ise
10:    kokj dugumune Hedef_Bilgisi mesajını gonder
11:  Eger Sen_Kok_Dugumun mesajı alınmış ise
12:    Eger Sayac sifirdan büyük ise
13:      Sen_Kok_Dugumun mesajını koni içindeki dugumlere gonder
14:    Eger Sayac sifira esit ise
15:      kok dugum ol
16:    Eger Onceden_Olusturulacak_Kapsama_Agaci_Sayisi sifirdan büyük ise
17:      Sen_Kok_Dugumun mesajını koni içindeki dugumlere gonder
18:  Eger Raymond_Kok mesajı alınmış ise
19:    kok dugum ol
20: Eger Ben bir kok dugum isem
21:   Eger Hedef_Bilgisi mesajı alınmış ise
22:     Hedefin konumunu, yonunu, hizini hesapla
23:     aci ve koni denklemlerini hesapla
24:     Olusturulacak kapsama agaci sayisini hesapla
25:     Sen_Kok_Dugumun mesajını koni içindeki dugumlere gonder

```

Algoritma 6.1. DKOIA Algoritmasının Sözde Kodu
6.2 Sonlu Durum Makinesi

Şekil 6.3, DKOIA Algoritmasının sonlu durum makinesini göstermektedir. Buna göre bir düğüm, UYKU, UYANIK, HEDEF_BULUNDU, LİDERLİĞİ_KAYBETTİ, ADAY, ÜYE, KÖK durumlarından birinde olabilir ve ADAYLIK_BİLGİSİ, BEN_KÖK_DÜĞÜMÜM, SEN_KÖK_DÜĞÜMSÜN, RAYMOND_KÖK, HEDEF_BİLGİSİ mesajlarından birini alıp, gönderebilir. Bunların yanında belirli bir süre beklenildiğini ve bu bekleme süresi dolduğunda zaman aşımı olduğunu belirten içteki mesajlar da SÜRE_AŞIMI, ADAYLIK_SÜRE_AŞIMI, LİDERLİK_SÜRE_AŞIMI, ALGILAMA_SÜRE_AŞIMI olarak ifade edilmiştir. Bu mesaj tiplerinden birini alması halinde, bir düğümün durumunda olacak değişiklik Şekil 6.4'teki sonlu durum makinesi üzerinde gösterilmiş ve devamında sonlu durum makinesinde

kullanılan durum ve mesajların açıklamaları yapılmıştır.



Şekil 6.4. DKOİA'nın Sonlu Durum Makinesi

Şekil 6.4.'te sonlu durum makinesi belirtilen DKOİA Algoritmasında, dinamik olarak kök düğüm değişim işlemi RAYMOND_KÖK mesajı ile gerçekleşmektedir. Eğer ÜYE durumundaki düğüme gelen RAYMOND_KÖK mesajının içindeki hedef düğüm numarası, üye düğümün numarası ise, düğüm KÖK durumuna geçer, aksi halde bu mesajı gerçek hedefe iletir.

Mevcut kök düğüm, hedefin yönü ve hızını dikkate alarak oluşturmuş olduğu koni içerisindeki düğümlere SEN_KÖK_DÜĞÜMSÜN mesajı göndererek kapsama ağaçlarını oluşturur.

Kapsama ağaçları oluşturmak için koni içindeki düğümlere gönderilen SEN_KÖK_DÜĞÜMSÜN mesajının içerisindeki temel parametreler ve açıklamaları aşağıda belirtilmiştir.

- **Kök Düğümün Koordinatları (x1,y1), Hedefin Açısı (aci), Koninin Üst ve Alt Doğrularını Oluşturacak açı (a)**

Bu bilgilerin iletilmesinin nedeni her düğüm sadece koni içinde kalan komşu düğümlerine mesaj gönderecektir. Bunu yapabilmesi için de düğümün, koniyi oluşturan iki doğrunun denklemini bilmesi gerekir.

Örneğin düğüm kendisine gelen SEN_KÖK_DÜĞÜMSÜN mesajı içerisindeki x1,y1 ve açı bilgilerinden aşağıdaki denklemleri oluşturur.

$$y \leq \tan(\text{aci} + \alpha)(x - x1) + y1 \text{ ve } y \geq \tan(\text{aci} - \alpha)(x - x1) + y1$$

Bu denklemleri sağlayan komşu düğümleri koni içerisinde ve kök düğüm bu düğümlere SEN_KÖK_DÜĞÜMSÜN mesajını gönderir.

- **Önceden Oluşturulacak Ağaç Sayısı**

Bu değer, koni içerisinde boylamasına kaç adet ağaç oluşturulacağını belirleyen değerdir. Kapsama ağacı oluşturan her düğüm bu değeri bir azaltacaktır ve bu değer 0'a eşit olduğunda kapsama ağacı oluşturma işlemi sona erecektir.

- **Sayac**

Oluşturulan kapsama ağaçları 2 sıçramalık olacaktır. Sayac parametresi kapsama ağacının 2 sıçramalık olmasını sağlamak için kullanılır.

- **Uzaklık**

Bu parametre, kapsama ağaçlarının çok fazla üst üste (Ing. overlap) çakışmasına engel olmak için kullanılmıştır. Bir düğüm kapsama ağacı oluşturduğunda, kendisinden uzaklık parametresinde belirtilen değer kadar uzaklıktaki ve koni içinde bulunan düğüme SEN_KÖK_DÜĞÜMSÜN mesajını gönderir ve bu ağacın kapsama ağacı oluşturmasını sağlar.

Önermiş olduğumuz Dinamik Koni Tabanlı Hedef İzleme Algoritmasının dağıtık açık kodu Algoritma 6.2’de verilmiştir.

Algoritma 6.2 dugum j için DKOİA Algoritması Dağıtık Kodu

```

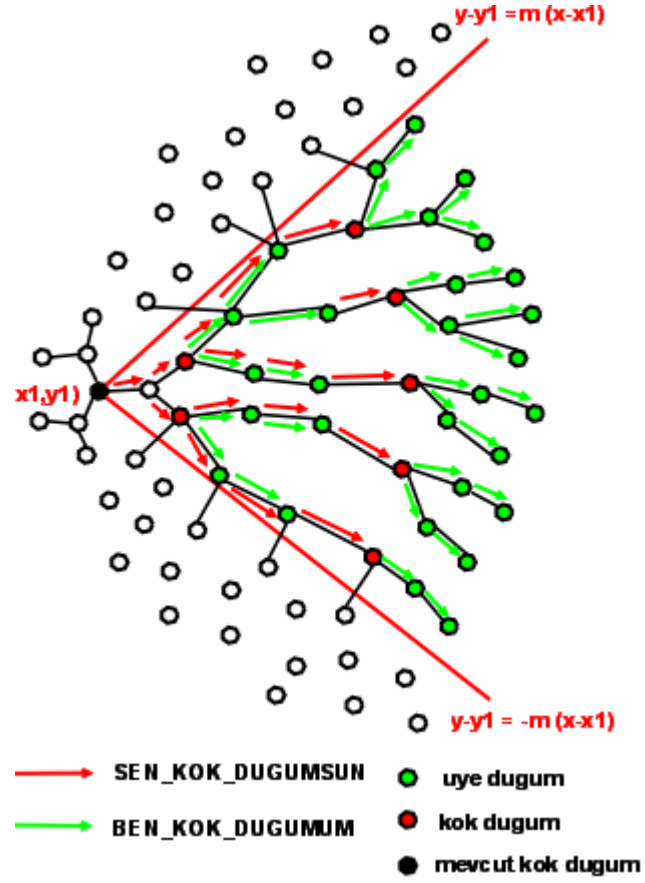
1: baslangic  $mevcut\_durum_j = UYANIK$ 
2:  $Lejant : \square DURUM \wedge gelen\_mesaj \rightarrow eylem$ 
3:  $S$ : Hedefin Hizi,  $K_t$ : Hedefin  $t$  anındaki konumu,  $Y$ : Hedefin Yonu
4:  $k_Y$ :  $Y$ 'ye en yakın komşu,  $\Gamma_j$ :  $j$ 'nin komşuları  $k_{kid}$ : Koni içindeki dugumlar
5:  $OAS$ : Önceden oluşturulacak kapsama ağacı sayısı
6:  $s$ : sayac  $u$ : Ağaçlar arası uzaklık
7:  $gh$ : Gerçek hedef  $hks$ : Hesaplanan konum  $kd$ : Koni denklemleri
8:
9:  $\square UYANIK \wedge Ben\_Kok\_Dugumum(i, s) \rightarrow kok_j \leftarrow id_i, mevcut\_durum_j \leftarrow UYE, Zamanlayici \text{ ata}$ 
10:  $Eger s > 0 \text{ ise } Ben\_Kok\_Dugumum(i, s - 1)$ 
11:  $mesajini \Gamma_j \text{ a ilet}$ 
12:  $\wedge Hedef\_Sinyal \rightarrow mevcut\_durum_j \leftarrow HEDEF\_BULUNDU, Zamanlayici \text{ ata}$ 
13:  $\wedge Sure\_Asimi \rightarrow mevcut\_durum_j \leftarrow UYKU, Zamanlayici \text{ ata}$ 
14:  $\wedge Sen\_Kok\_Dugumum(OAS, Y, gh, u, kd) \rightarrow Eger gh <> benim\_id \wedge u > 0 \text{ ise}$ 
15:  $Sen\_Kok\_Dugumum(OAS, Y, gh, u - 1, kd) \text{ mesajini } k_{kid}'e$ 
16:  $gonder,$ 
17:  $Eger gh <> benim\_id \wedge u > 0 \text{ ise}$ 
18:  $Ben\_Kok\_Dugumum(i, s - 1) \text{ mesajini } \Gamma_j'a$ 
19:  $gonder, mevcut\_durum_j \leftarrow KOK, Zamanlayici \text{ ata}$ 
20:  $Eger gh == benim\_id \text{ ise } mevcut\_durum_j \leftarrow KOK,$ 
21:  $Zamanlayici \text{ ata}$ 
22:  $\square UYKU \wedge Sure\_Asimi \rightarrow kok_j \leftarrow id_i, mevcut\_durum_j \leftarrow UYANIK, Zamanlayici \text{ ata}$ 
23:  $\square UYE \wedge Ben\_Kok\_Dugumum(i, s) \rightarrow kok_j \leftarrow id_i, mevcut\_durum_j \leftarrow UYE, Zamanlayici \text{ ata}$ 
24:  $Eger s > 0 \text{ ise } Ben\_Kok\_Dugumum(i, s - 1)$ 
25:  $mesajini \Gamma_j \text{ a ilet}$ 
26:  $\wedge Hedef\_Sinyal \rightarrow HEDEF\_BILGISI'ni kok_j \text{ dugumune gonder, Zamanlayici ata}$ 
27:  $\wedge Sure\_Asimi \rightarrow mevcut\_durum_j \leftarrow UYANIK, Zamanlayici \text{ ata}$ 
28:  $\wedge Sen\_Kok\_Dugumum(OAS, Y, gh, u, kd) \rightarrow Eger gh <> benim\_id \wedge u > 0 \text{ ise}$ 
29:  $Sen\_Kok\_Dugumum(OAS, Y, gh, u - 1, kd) \text{ mesajini } k_{kid}'e$ 
30:  $gonder,$ 
31:  $Eger gh <> benim\_id \wedge u > 0 \text{ ise}$ 
32:  $Ben\_Kok\_Dugumum(i, s - 1) \text{ mesajini } \Gamma_j'a$ 
33:  $gonder, mevcut\_durum_j \leftarrow KOK, Zamanlayici \text{ ata}$ 
34:  $Eger gh == benim\_id \text{ ise } mevcut\_durum_j \leftarrow KOK,$ 
35:  $Zamanlayici \text{ ata}$ 
36:  $\wedge Raymond\_Kok \rightarrow mevcut\_durum_j \leftarrow KOK$ 
37:  $\square KOK \wedge Hedef\_Bilgisi \rightarrow Hedefin t \text{ anındaki } K_t' \text{ yi hesapla}$ 
38:  $K_t \text{ ve } K_{t-1} \text{ 'den } Y, S \text{ ve } OAS'yi \text{ hesapla,}$ 
39:  $koni \text{ ve aci denklemlerini hesapla,}$ 
40:  $Sen\_Kok\_Dugumum(OAS, Y, gh, u, kd) \text{ mesajini } k_{kid} \text{ gonder}$ 
41:  $\wedge Sure\_Asimi \rightarrow mevcut\_durum_j \leftarrow UYANIK, Zamanlayici \text{ ata}$ 
42:  $\wedge Ben\_Kok\_Dugumum(i, s) \rightarrow mevcut\_durum_j \leftarrow UYE, Zamanlayici \text{ ata, } kok_j \leftarrow id_i$ 
43:  $\square HEDEF\_BULUNDU \wedge Adaylik\_Bilgisi(i) \rightarrow Adaylik\_Bilgisi(i)'ni \Gamma_j' \text{ a ilet,}$ 
44:  $mevcut\_durum_j \leftarrow LIDERLIK\_KAYBEDILDI$ 
45:  $\wedge Sure\_Asimi \rightarrow \Gamma_j'ye Adaylik\_Bilgisi \text{ gonder, Zamanlayici ata, } mevcut\_durum_j \leftarrow ADAY$ 
46:  $\square LIDERLIK\_KAYBEDILDI \wedge Adaylik\_Bilgisi(i) \rightarrow Adaylik\_Bilgisi(i)'ni \Gamma_j'ya \text{ ilet}$ 
47:  $\wedge Ben\_Kok\_Dugumum(i, s) \rightarrow mevcut\_durum_j \leftarrow UYE, kok_j \leftarrow id_i,$ 
48:  $\square ADAY \wedge Adaylik\_Bilgisi(i, RSS) \rightarrow eger RSS > kendi\_RSS \text{ ise}$ 
49:  $Adaylik\_Bilgisi(i, RSS)'ni \Gamma_j'e \text{ ilet,}$ 
50:  $mevcut\_durum_j \leftarrow LIDERLIK\_KAYBEDILDI$ 
51:

```

Algoritma 6.2. DKOİA Algoritmasının Açık Dağıtık Kodu

6.3 Örnek Bir işlem

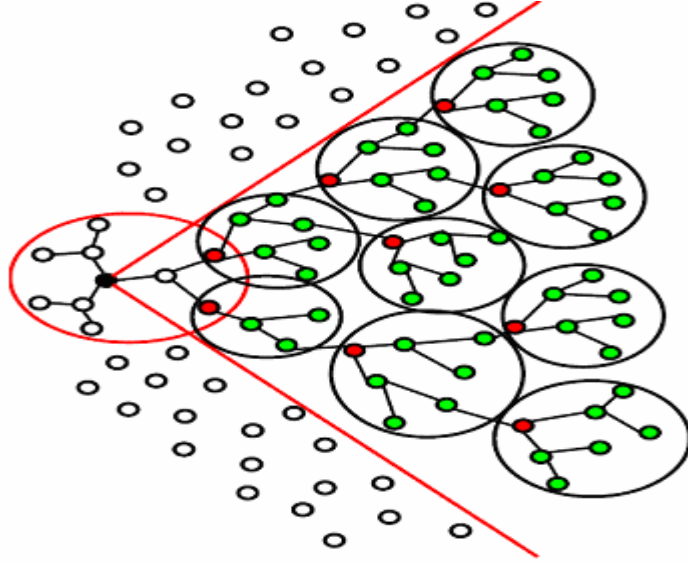
Şekil 6.5, Dinamik Koni Tabanlı Hedef İzleme Algoritmasının bir örnek üzerinden çalışmasını göstermektedir.



Şekil 6.5. Kök Düğümün Koni İçinde Kalan Düğümlerle Kapsama Ağaçları Oluşturması

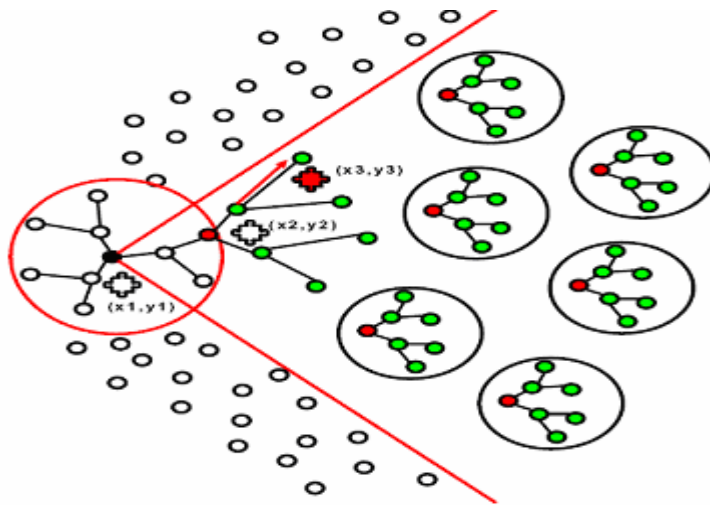
Şekil 6.5.'ten de görüldüğü gibi, $x1, y1$ koordinatlarında bulunan mevcut kök düğüm öncelikle, hedefin hızı ve yönü ile ilişkili olarak koninin üst ve alt doğrularının denklemlerini oluşturmuş ve koni içerisinde kalan düğümlere, yukarıda belirtilen parametreleri içeren SEN_KÖK_DÜĞÜMSÜN mesajını göndermiştir. Mesajın içerisindeki uzaklık değeri sıfıra eşit oldukça, ilgili düğüm KÖK durumuna geçmiş ve kendisinden iki sıçrama mesafesindeki düğümlerle kapsama ağacını oluşturmuştur. KÖK durumuna geçen ve kapsama ağaçlarını oluşturan düğüm, daha sonra mesajın içerisindeki Önceden_Oluşturulacak_Kapsama_Ağacı değerini bir azaltır ve bu değerın sıfıra eşit olup olmadığını kontrol eder. Eğer bu değer hala sıfırdan büyükse, SEN_KÖK_DÜĞÜMSÜN mesajını koni içerisindeki düğümlere göndermeye devam eder.

Bütün bu işlemlerden sonra elde edilmiş olan yapı Şekil 6.6'da gösterilmektedir.



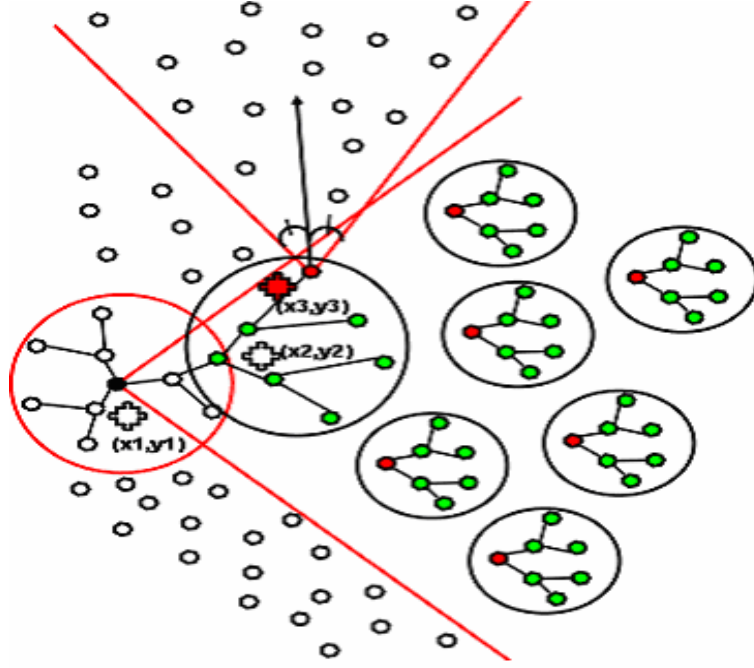
Şekil 6.6. DKOİA'nın Çalışması Sonucunda Elde Edilen Yapı

Hedef, koni içerisinde hareket ettikçe, hedefin içinde bulunduğu kapsama ağacı üzerinde yakın olduğu düğümler sürekli değişim gösterecektir. Her bir önceden oluşturulan kapsama ağacında dinamik olarak kök değişim işlemi gerçekleşecek ve Raymond Algoritmasından esinlenerek ağacın düğümleri arasındaki yollar yeniden yapılandırılacaktır.



Şekil 6.7. DKOİA'da Kök Düğümün Dinamik Değişimi

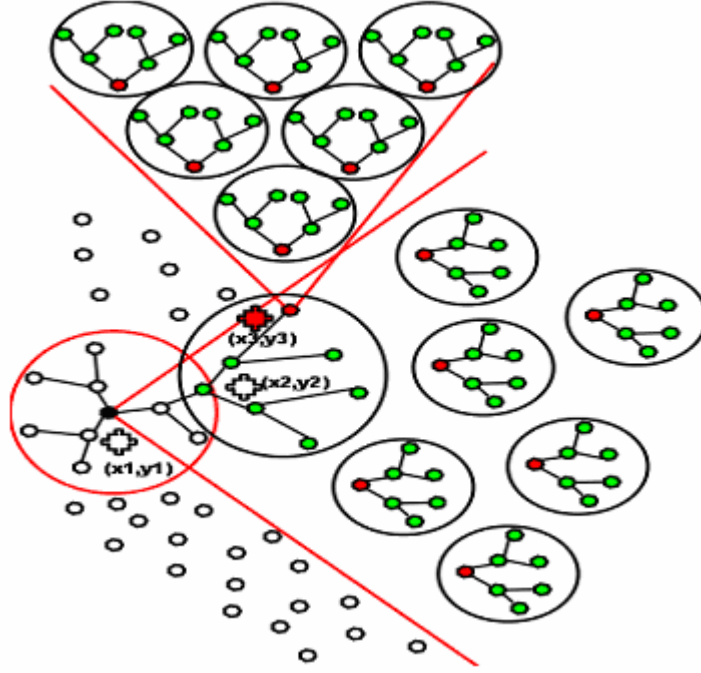
Şekil 6.7’de hedefin hareket etmesiyle, farklı zamanlardaki konumları $(x1,y1)$, $(x2,y2)$, $(x3,y3)$ gösterilmektedir. Hedefin $x3,y3$ konumu mevcut kök düğümden uzakta olup, kök düğüm üyelerinden birine daha yakın bulunmaktadır. Kök düğüm, ağacındaki düğümlerden elde ettiği sinyal gücü değerlerinden bunu fark eder ve en yüksek sinyal gücü değerine sahip düğüme RAYMOND_KÖK mesajı gönderir. Böylece hedefe en yakın olan düğüm, dinamik olarak kök düğüm seçilmiş olur.



Şekil 6.8. DKOİA’da Yeni Kök Düğümün Koni Oluşturması

Şekil 6.8, yeni kök düğümün hedefin yönünü belirlemesini ve bu yöne bağlı olarak oluşturmuş olduğu koniyi göstermektedir. Buna göre, yeni seçilen kök düğüm, kendisine gelen RAYMOND_KÖK mesajı içerisindeki yön bilgisinden α açısı kadar sağa ve sola doğrular çizerek bu doğruların denklemlerini belirler ve şartları oluşturur. Hedefin hızına bağlı olarak da koninin boyunca oluşturulacak ağaç sayısını ifade eden Önceden_Oluşturulacak_Ağaç_Sayısı parametresine değer atar.

Şekil 6.9’da hedefin yeni yönü doğrultusunda oluşturulan koninin içinde kapsama ağaçlarının oluşturulması gösterilmektedir.



Şekil 6.9. DKOİA’da Yeni Kök Düğümün Koni İçinde Kapsama Ağaçları Oluşturması

6.4 DKOIA Algoritmasının Mesaj Karmaşıklık Analizi

Bu bölümde, DKOIA algoritmasının her bir aşamasındaki kullandığı mesaj sayısı adım adım hesaplanarak, toplam mesaj karmaşıklığı belirlenmiştir.

Teorem 5.4.1: DKOIA algoritmasının mesaj karmaşıklığı $O(n)$ ’dir.

İspat:

- **Kök düğümü belirleme aşamasında** kullanılan yöntem DÇKIA ve DÇKAIA ile aynıdır, dolayısıyla mesaj sayısı: $k \cdot a$
- **İlk kapsama ağacının oluşturulması aşamasında**, seçilen kök düğüm, k adet komşu düğümüne BEN_KÖK_DÜĞÜMÜM mesajı gönderir. Bu algorithmada kullanmış olduğumuz ağaçların derinliği 2 olduğu için, bu aşamadaki mesaj sayısı: $k + k \cdot k$

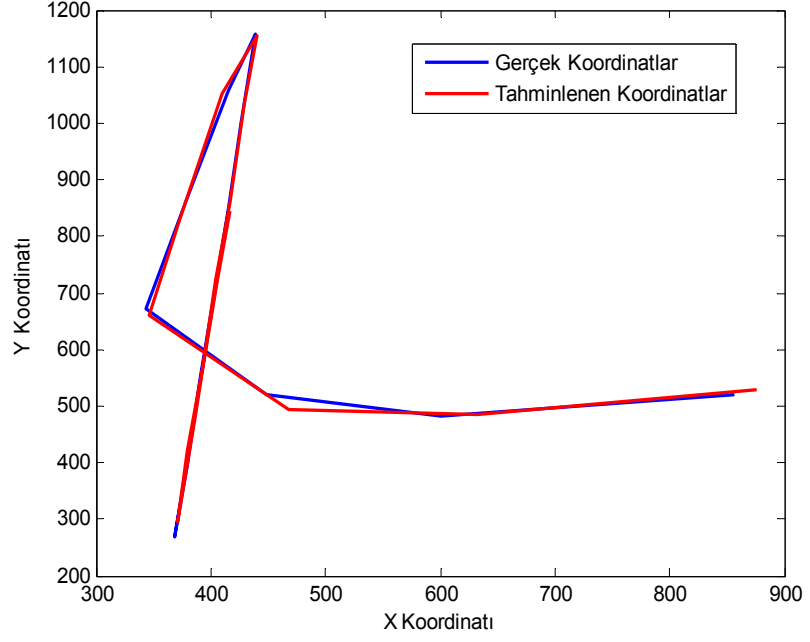
- **Üye düğümlerin elde ettikleri verileri kök düğüme iletmesi aşamasında**, kapsama ağacı içerisindeki $k+k*k$ adet üye düğüm iki farklı zamanda elde ettikleri veriyi lider düğüme iletir. Bu aşamadaki mesaj sayısı: $2*(k+k*k)$
- **Kök düğümün hedefin yönü doğrultusunda ağaç oluşturma mesajı göndermesi aşamasında**, SEN_KÖK_DÜĞÜMSÜN mesajı, kök düğümün belirlemiş olduğu koni içinde kalan düğümlere iletilir ve her bir düğüm de sıçrama sayısı (h) sıfır olana kadar bu mesajı iletmeye devam eder. Bu durumda koni içindeki her bir düğüm aldığı mesajı iletir ve bu iletme işlemi h kadar sürer. Kids, koninin belirtilen sınırları içerisinde kalan düğüm sayısı olmak üzere, bu aşamada kullanılan mesaj sayısı: **kids**
- **Önceden ağaçların oluşturulması aşamasında**, her bir önceden oluşturulan kök düğüm, ağacını oluşturabilmek için k adet komşu düğüme BEN_KÖK_DÜĞÜMÜM mesajı gönderir. Koni içinde oluşturulabilecek kapsama ağacı sayısına kia dersek, bu aşamadaki mesaj sayısı: **kia*(k+k*k)**
- Ortamda n adet düğümün olduğunu düşünürsek, simülasyon süresi boyunca ortamda $n/(1+k+k*k)$ adet ağaç oluşturulabilir. $(1+k+k*k)$, bir ağaçta bulunabilecek düğüm sayısını belirtmektedir.

Sonuç olarak, yukarıda belirtilen mesaj sayıları toplandığında, DKOIA'nın hedefi takip edebilmesi için kullandığı toplam mesaj sayısı $(k*a+3*(k+k*k)+kids+kia*(k+k*k))* (n/(1+(k+k*k)))$ olarak hesaplanır ki bu da mesaj karmaşıklığının **O(n)** olduğu anlamına gelir.

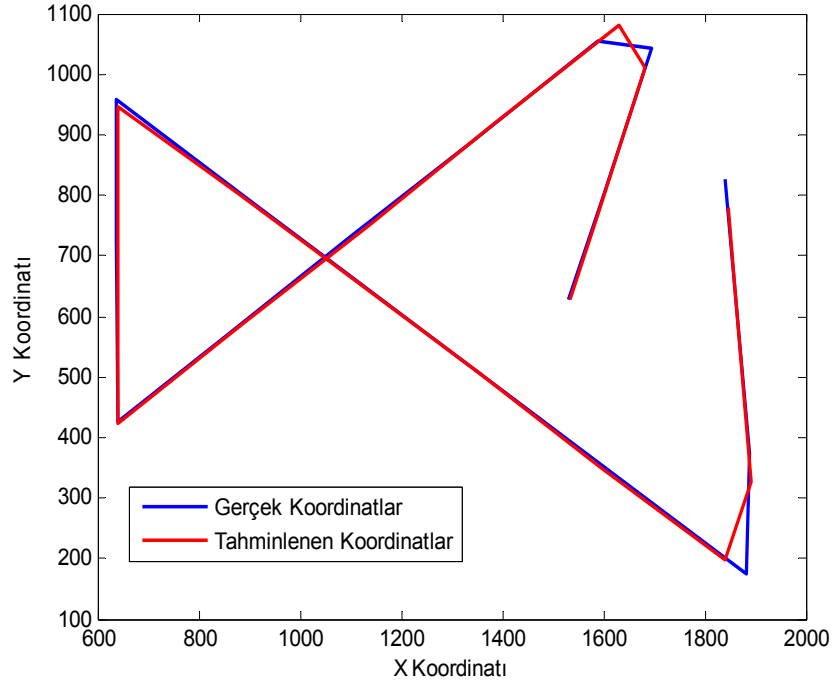
6.5 Simülasyon Sonuçları

Önermiş olduğumuz Dinamik Koni Tabanlı Hedef İzleme Algoritmasını, çok hızlı hareket eden hedefler için uygulayıp, hedef takibi ve hedefi kaybetme oranları için simülasyon sonuçları elde edilmiştir. Ölçüm sonuçları, 70'er adet deneyin ortalaması alınarak elde edilmiştir. Şekil 6.10.'da çok hızlı hareket eden hedefin, 30 m/s ile 100 m/s arasında değişen hız değerleri için takibi gösterilmektedir. Hız değerleri arttıkça gerçek rota ile hesaplanan rota arasındaki farkta artış olsa da,

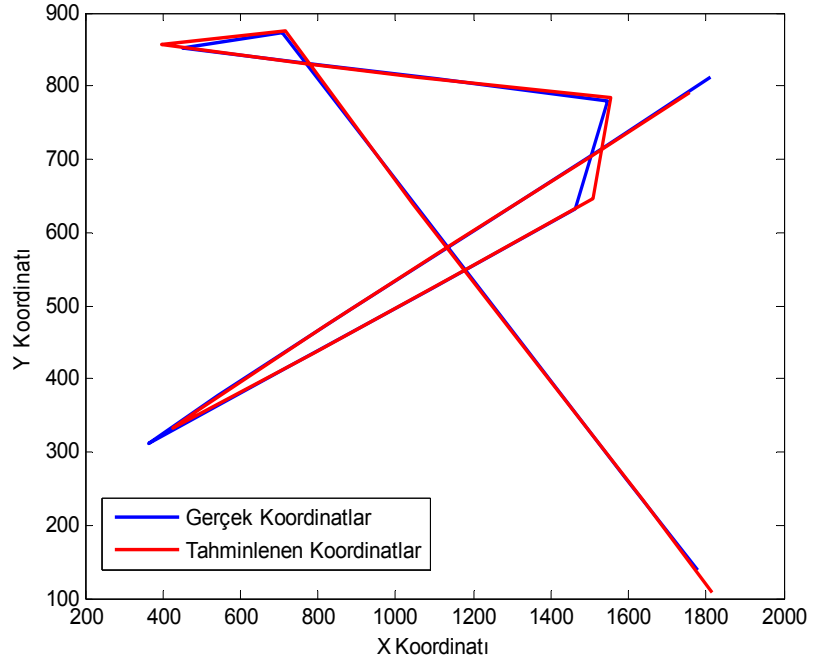
hedefin çok hızlı hareket ettiği durumlarda rota da çok büyük bir sapma olmamıştır.



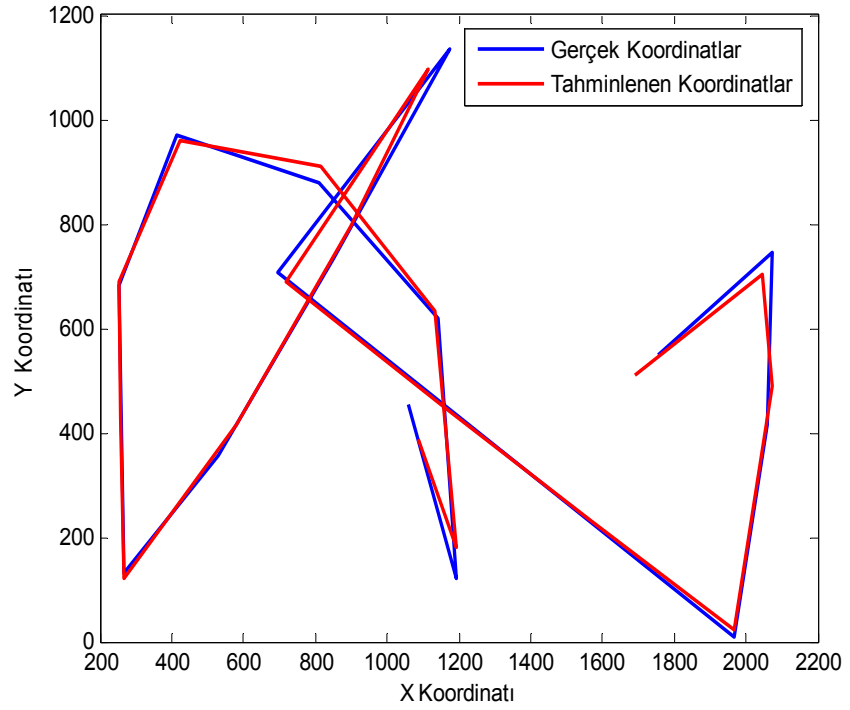
(a) 30-40 m/s



(b) 50-60 m/s



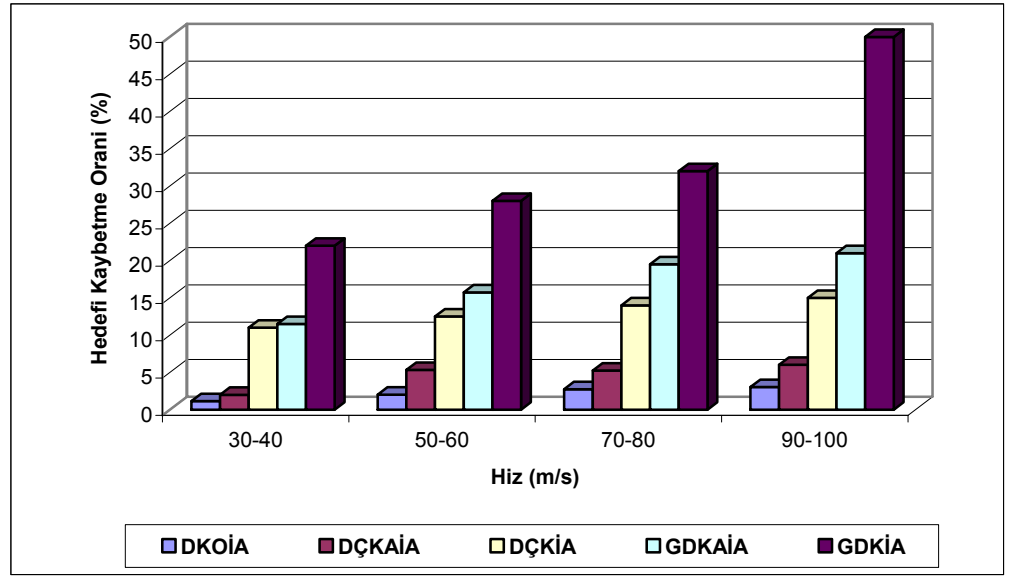
(c) 70-80 m/s



(d) 90-100 m/s

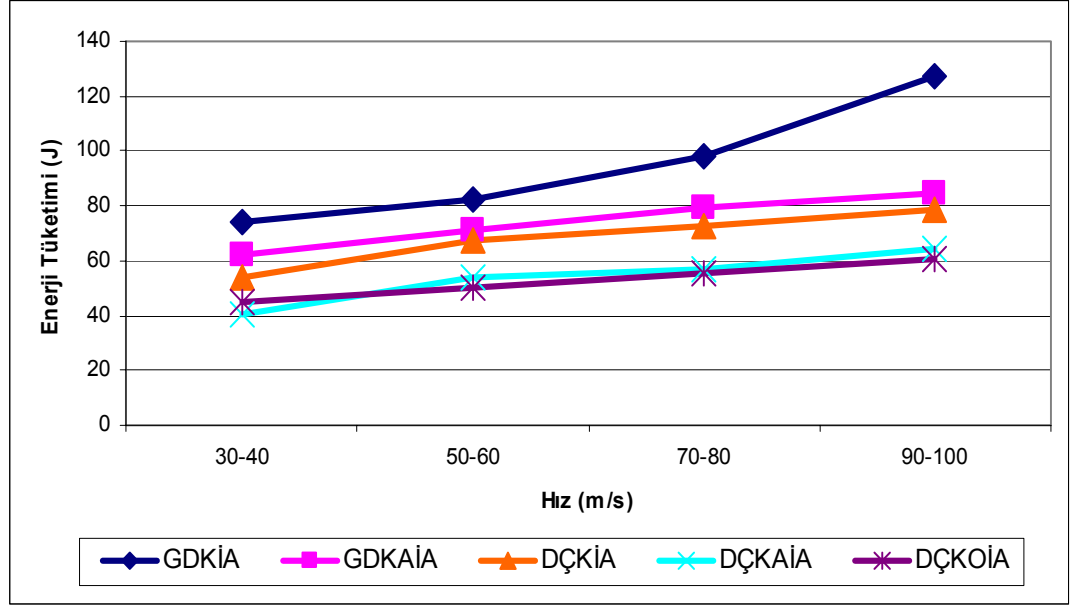
Şekil 6.10. DKOİA ile Hedef Takibi

Önermiş olduğumuz algoritma ile hedefi kaybetme olasılığını azaltmayı hedeflemiş bulunmaktayız. Şekil 6.11’de önerilen DKOİA, DÇKAİA, DÇKİA Algoritmaları ile mevcut yaklaşımlar olan GDKAİA ve GDKİA Algoritmaları hedefi kaybetme oranları için karşılaştırılmıştır. Buna göre, her bir hız değeri için önerilen algoritmaların mevcut yaklaşımlara göre hedefi kaybetme olasılığı açısından çok daha iyi sonuç verdiğini görmekteyiz. Bunun yanında hız değeri arttıkça, önerilen ve mevcut yaklaşımlar arasındaki hedefi kaybetme oranları arasındaki farkın daha da arttığı görülebilmektedir.



Şekil 6.11. DKOİA , DÇKAİA,DÇKİA, GDKAİA ve GDKİA Algoritmalarının Hedefi Kaybetme Olasılığı için Karşılaştırılması

Ayrıca Şekil 6.11’de önermiş olduğumuz üç yaklaşım, DKOİA, DÇKAİA, DÇKİA Algoritmalarını, hedefi kaybetme oranları için karşılaştıracak olursak, DÇKAİA Algoritmasının DÇKİA Algoritmasına göre, DKOİA Algoritmasının da DÇKAİA Algoritmasına göre daha iyi sonuç verdiği görülmektedir.



Şekil 6.12. DKOİA , DÇKAİA,DÇKİA, GDKAİA ve GDKİA Algoritmalarının Enerji Tüketimleri için Karşılaştırılması

Şekil 6.12.'de, önerilen DKOİA, DÇKAİA ve DÇKİA Algoritmaları ile mevcut yaklaşımlar olan GDKAİA ve GDKİA Algoritmaları enerji tüketimleri için karşılaştırılmıştır. Önermiş olduğumuz yaklaşımların her birinde hedefi kaybetme oranımız düşük olduğu için, daha az sıklıkta düzeltme mekanizması kullanılmış ve dolayısıyla da her bir önerilen yaklaşımdaki enerji tüketimi mevcut yaklaşımlara göre daha iyi sonuç vermiştir. Enerji tüketimi için en iyi performansı önermiş olduğumuz DÇKOİA gösterirken, en kötü performansı hedefi kaybetme oranının en yüksek olduğu mevcut yaklaşım olan GDKİA göstermiştir.

7 DOĞRUSAL OLMAYAN HAREKETLER İÇİN KONİ TABANLI HEDEF İZLEME ALGORİTMASI

Literatürdeki hedef takibi çalışmalarının çoğunda hedefin doğrusal hareket ettiği düşünülmüş ve izleme problemi, doğrusal durum uzayı (linear state space) modelleri kullanılarak formüle edilmiştir. Fakat gerçek hayattaki hedef takibi uygulamalarında hedef, ani ivmelenme ve dönme gibi hareket tipleri içeren, doğrusal olmayan hareketler yapabilmektedir.

Hedef takibi uygulamalarında, hedefin bir sonraki konumunu tahmin edebilmek için literatürde bir çok tahminleme yöntemi mevcuttur. Kalman Filtreleri, tahminleme için kullanılan en popüler yöntemlerden biridir. Doğrusal olmayan sistemlerde tahminleme için, Kalman filtreleri tabanlı Genişletilmiş Kalman Filtreleri önerilmiştir. Genişletilmiş Kalman Filtreleri ile doğrusal olmayan model doğrusallaştırılarak hedefin bir sonraki konumu tahminlenmektedir. Bu doğrusallaştırma işlemi, genişletilmiş kalman filtresinin, manevra yapan hedefin takip edilebilirliğini olumsuz yönde etkilemektedir.

Doğrusal olmayan hareketlerde tahminleme işlemi için kullanılan bir başka yaygın tahminleme yöntemi ise Parçacık Filtreleridir. Olasılık tabanlı olan parçacık filtreleri, hedefin bir sonraki konumunu tahminleyebilmek için parçacıkların olasılıksal dağılımından yararlanır. Bu filtrelerde gerçekleştirilen temel işlem, parçacıklara uygun ağırlık değerleri atayarak, bu ağırlıkları güncelleyip hedefin konumunu tahminlemektir (Kambhampati, 2008). Parçacık filtrelerinin, Kalman ve Genişletilmiş Kalman Filtrelerine göre en büyük avantajı, herhangi bir doğrusallaştırma işlemine gerek olmamasıdır.

Gerçek hayattaki hedef hareket modellerinin doğrusal olmaması, bizi doğrusal olmayan hareket modellerini incelemeye yönlendirmiş, ilerleyen alt bölümlerde, doğrusal olmayan hareketlerde tahminleme işlemi için sıkça kullanılan Genişletilmiş Kalman Filtreleri ve Parçacık Filtreleri ile ilgili daha detaylı bilgiler verilmiştir. Şu ana kadar, önermiş olduğumuz DÇKİA, DCKAİA ve DKOİA algoritmalarından doğrusal hareketler için, hedef takibinde en iyi performansı veren DKOİA algoritmasına, Genişletilmiş Kalman Filtreleri ile Parçacık Filtreleri entegre edilerek, doğrusal olmayan hareketler hedef takibi gerçekleştirilmiştir.

7.1 Sistem Modeli

Dinamik bir sistemin analizi ve tahminlenmesi için, sistem durum modeli ve ölçüm modeli olmak üzere en az iki modele gereksinim duyulmaktadır. Sistem durum modeli, sistemin zamanla değişimini ifade ederken, ölçüm modeli ortamdan elde edilen ölçümleri belirten modeldir.

Doğrusal olmayan hareketler yapan bir hedefin, takip sistem modeli, aşağıdaki dinamik durum uzayı ile belirtilebilmektedir.

$$\begin{aligned} X_k &= F X_{k-1} + w_k \\ Z_k &= h_k(X_k, n_k) \end{aligned} \quad (1)$$

Bu eşitliklerdeki k zamanı, X_k sistemin k anındaki durumunu, F geçiş matrisini ve w_k da sistem gürültüsünü belirtmektedir. Z_k , duyargaların elde etmiş olduğu ölçümlerden oluşan gözlem modelini ve n_k da ölçüm gürültüsünü belirtmektedir. X_k , hedefin k anındaki konum, hız, ivme gibi durum bilgilerini içerir. F geçiş matrisi, hedefin hareket türüne göre değişen değerler alır.

Bir çok hedef takip sistemi, kolay uygulanabilirliğinden dolayı sabit hızla giden hedefleri takip etmeyi tercih etmişlerdir. Fakat gerçek hayat uygulamalarında hedefin aniden dönüş yapması ve ivmelenmesi gibi durumlar söz konusu olabilir.

Önermiş olduğumuz hedef takip sisteminde, hedef sabit hızlı, ivmeli ve dönüşlü hareketler yapabilmekte ve her bir hareket tipi için hedef takip edilebilmektedir. Sistemimizde hedefin hareket modelleri, rasgele ve farklı hareket türlerini içerecek şekilde üretilmiştir.

Her bir hareket türü için hedefin durum vektörü X ve geçiş matrisi F aşağıdaki gibi tanımlanmıştır (Yang et al., 2006):

Sabit Hızlı Hareket Modeli:

$$X_{k(CV)} = [x, v_x, y, v_y]^T$$

$$F_{k(CV)} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Sabit İvmeli Hareket Modeli:

$$X_{k(CA)} = [x, v_x, a_x, y, v_y, a_y]^T$$

$$F_{k(CA)} = \begin{bmatrix} 1 & T & 1/2T^2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 1/2T^2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Sabit Dönüslü Hareket Modeli:

$$X_{k(CT)} = [x, v_x, y, v_y, \Omega]^T$$

$$F_{k(CT)} = \begin{bmatrix} 1 & \sin(\Omega(k)T)/\Omega(k) & 0 & \cos(\Omega(k)T) - 1/\Omega(k) & 0 \\ 0 & \cos(\Omega(k)T) & 0 & -\sin(\Omega(k)T) & 0 \\ 0 & -\cos(\Omega(k)T) + 1/\Omega(k) & 1 & \sin(\Omega(k)T)/\Omega(k) & 0 \\ 0 & \sin(\Omega(k)T) & 0 & \cos(\Omega(k)T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Yukarıda belirtilen matrislerde kullanılan, T örnekleme aralığını, x ve y hedefin koordinatlarını, v_x ve v_y hedefin hızını, a_x ve a_y hedefin ivmesini belirtmektedir.

7.2 Genel Tahminleme ve Güncelleme Çerçevesi

Sistem durum uzayının formüle edilmesi ve yeni ölçümler ile sistemin güncellenmesi Bayes yaklaşımına benzemektedir. Bu benzerlik, dinamik durum tahminleme problemleri için genel bir çerçevenin oluşturulabilmesini sağlar (Arulampalam et al., 2001).

Bayes Teoremi, Thomas Bayes tarafından 18. yüzyılda geliştirilmiştir ve istatistiksel anlamda bağımlı olayların şartlı olasılık değerlerinin hesaplanmasını sağlayan bir teoremdir (Serper, 1992). Bayes Teoremi, tahminlemede, mevcut olan bilgiyi ek bilgi ile birleştirmek ve son olasılık değerlerini oluşturmak için kullanılmaktadır. (Altındış, 2002).

Bayes yaklaşımında, bir sonraki sistem durumunu tahminlemek için, elde edilen ölçüm değerlerine dayalı olarak sonraki olasılıksal dağılım fonksiyonu (Ing. Posterior Probability Density Function) oluşturulmaya çalışılır. Sonraki olasılıksal dağılım fonksiyonu, eldeki bütün veriyi kullandığı için tahminleme probleminde iyi bir çözüm yöntemi olduğu söylenebilir.

Bayes teoremi, bir olayın meydana gelme olasılığının, olaya yönelik ek bilgi edinilmesi halinde nasıl değişeceğini gösterir. Bir başka ifadeyle, B olayına koşullu bir A olayı için olasılık değeri ile A olayına koşullu bir B olayı için olasılık değeri arasındaki ilişkiye Bayes Teoremi denmektedir (Wikipedia).

Bayes teoreminin genel formülü aşağıda verilmiştir:

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)} \quad (5)$$

Bu formüle göre, $P(A)$ terimine önsel olasılık adı verilir. Çünkü B olayı hakkında önceden herhangi bir bilgi içermektedir. $P(A|B)$ terimi, verilmiş B için A'nın koşullu olasılığını belirtmektedir. $P(B|A)$, verilmiş A için B'nin koşullu olasılığını belirtir. $P(B)$ terimi, B olayı için önsel bir olasılıktır ve matematiksel olarak da normalizasyon için bir sabit olarak belirtilir (Wikipedia).

Bayes yaklaşımı özyineli bir yapıya sahiptir. Buna göre veri depolama yapıp sonradan bütün veriler üzerinde işlem yapmak yerine, yeni ölçüm değerleri elde edildikçe tahminleme işlemi sırasıyla gerçekleşebilmektedir. Böyle bir yapı,

tahminleme ve güncelleme olmak üzere iki temel aşamadan oluşmaktadır. Tahminleme aşamasında sistem modeli kullanılarak, sistemin bir sonraki olasılıksal dağılım fonksiyonu bulunmaya çalışılırken, güncelleme aşamasında gerçekleştirilen tahmin, elde edilen yeni ölçüm değerleri ile güncellenir (Arulampalam et al., 2001).

Tahminleme Aşaması:

$$p(X_{k-1}|Z_{1:k-1}) \longrightarrow p(X_k|Z_{1:k-1}) \quad (6)$$

denklemini ifade edilebilirken, güncelleme aşaması:

$$p(X_k|Z_{1:k-1}), Z_k \longrightarrow p(X_k|Z_{1:k}) \quad (7)$$

denklemini ifade edilebilir. Bu denklem ile mevcut elde edilen ölçüm değerleri ile gerçekleştirilmiş olan tahmin güncellenmiştir olur.

(k-1) anında sonraki olasılıksal dağılım fonksiyonunun

$$p(X_{k-1}|Z_{1:k-1}) \quad (8)$$

olduğunu düşünürsek hedefin, herhangi bir ölçüm değeri olmadan k anındaki durumu

$$p(X_k|Z_{1:k-1}) = \int p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1})dx_{k-1} \quad (9)$$

denklemini ifade edilebilir ve bir sonraki sistem durumu, yeni elde edilen ölçüm değerleri ile aşağıdaki formül ile hesaplanabilir:

$$p(X_k|Z_{1:k}) = \frac{p(Z_k|X_k)p(X_k|Z_{1:k-1})}{p(Z_k|Z_{1:k-1})} \quad (10)$$

Yukarıda belirttiğimiz formülü adım adım açacak olursak:

$$p(X_k|Z_{1:k}) \quad (11)$$

Denklem, Bayes Kuralı'ndan,

$$\frac{p(Z_{1:k}|X_k)p(X_k)}{p(Z_{1:k})} \quad (12)$$

ile ifade edilebilir. Yukarıdaki ifadede,

$$p(Z_{1:k}) \quad (13)$$

ifadesini

$$p(Z_k, Z_{1:k-1}) \quad (14)$$

şeklinde ayrıştıracak olursak:

$$\frac{p(Z_k, Z_{1:k-1}|X_k)p(X_k)}{p(Z_k, Z_{1:k-1})} \quad (15)$$

ifadesini elde etmiş oluruz.

$$p(a,b|c)=p(a|b,c)p(b|c) \quad (16)$$

ve

$$p(a,b)=p(a|b)p(b) \quad (17)$$

kurallarından

$$\frac{p(Z_k|Z_{1:k-1}, X_k)p(Z_{1:k-1}|X_k)p(X_k)}{p(Z_k|Z_{1:k-1})p(Z_{1:k-1})} \quad (18)$$

ifadesi elde edilir.

Denklem, bayes kuralından,

$$\frac{p(Z_k | Z_{1:k-1}, X_k) p(X_k | Z_{1:k-1}) p(Z_{1:k-1}) p(X_k)}{p(Z_k | Z_{1:k-1}) p(Z_{1:k-1}) p(X_k)} \quad (19)$$

şeklinde yazılabilir.

Gözlemlerden bağımsız olarak dış değerleri denklemden çıkaracak olursak:

$$\frac{p(Z_k | X_k) p(X_k | Z_{1:k-1})}{p(Z_k | Z_{1:k-1})} \quad (20)$$

güncelleme denklemi elde edilir (Mühlich, 2003).

7.3 Genişletilmiş Kalman Filtreleri

Kalman filtreleri doğrusal hareket eden nesnelerin bir sonraki konumunu tahminlemek için kullanılan en yaygın yöntemlerden biridir. Bu yaklaşımda, bir sonraki durum tahmini için önceki durum tahmini ve yeni ölçüm değerleri kullanılmaktadır.

Kalman Filtreleri özyineli bir yapıya sahiptir. Tahminleme ve güncelleme olmak üzere iki temel aşaması vardır. Tahminleme işlemi için hareket bağıntılarından elde edilen geçiş matrisinden ve önceki tahminlemeden yararlanır. Güncelleme işlemi içinse, duyarga düğümlerinden elde ettiği ölçüm bilgilerini kullanır. Kalman filtresinde gerçekleştirdiğimiz işlemlerde, tahminleme hatasının kovaryansının da etkisi vardır. Tahminleme hatasının kovaryansı ve tahmini ölçüm ile gerçek ölçüm arası farkın kovaryansı Kalman kazancını belirler. Kalman kazancı, yeni gerçekleştirilecek olan tahminde, duyarga düğümlerinden elde edilen ölçümlerin ne derecede etkili olacağını belirleyen katsayıdır. Standart Kalman Filtre denklemleri aşağıdaki gibi gösterilebilir (Tasdemir ve Kent, 2006; Lana, 2001; Welch and Bishop, 2001):

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} \quad (21)$$

Yukarıda belirtilen denklemde F_k durum geçiş matrisi ve $\hat{x}_{k|k-1}$, k anındaki ilk tahminlemedir.

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (22)$$

$P_{k|k-1}$, hata kovaryansıdır. Durum geçiş matrisi F_k ile başlangıç hata kovaryans değerine $P_{k-1|k-1}$, işlem gürültüsünün Q_k , eklenmesi ile elde edilir.

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \quad S_k = H_k P_{k|k-1} H_k^T + R_k \quad (23)$$

$\hat{y}_k$, gerçek ölçüm ile tahmini ölçüm arasındaki farkı ve S_k da kovaryansı belirtmektedir. Denklemdaki H_k gözlem modeli, R_k ise ölçüm gürültüsüdür.

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (24)$$

K_k , yeni tahminlemede duyarga düğümlerinden elde edilen ölçümlerin ne derece etkili olacağını belirleyen Kalman kazancını belirtir.

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \quad (25)$$

$\hat{x}_{k|k}$, en son yapılan durum tahminlemesini belirtir

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (26)$$

$P_{k|k}$, bir sonraki durum tahminlemesi için kullanılacak hata kovaryansını belirtir.

Yukarıda belirtilen denklemler, standart Kalman Filtresinde doğrusal hareketler yapan hedefin bir sonraki konumu tahminlemek ve güncellemek için kullanılan denklemlerdir. Fakat gerçek hedef takibi sistemlerinde, hedef ivmelenme ve dönme gibi hareketler yapabilmektedir. Bu şekilde doğrusal hareket etmeyen hedefin konumunu tahminlemek için Kalman filtresi temelli Genişletilmiş Kalman Filtreleri önerilmiştir.

Genişletilmiş Kalman Filtresinde, hedefin bir sonraki durumunun tahminlenebilmesi ve güncellenebilmesi için öncelikle, hedefin doğrusal olmayan hareket modelinin doğrusallaştırılması gerekir (Canan, 2004).

Örneğin;

$$\begin{aligned}
 f_x(k) &= x(k+1) = x(k) + \Delta D(k) \cdot \cos(\phi_{k-1} + \frac{\Delta\phi(k)}{2}) + w_x(k) \\
 f_y(k) &= y(k+1) = y(k) + \Delta D(k) \cdot \sin(\phi_{k-1} + \frac{\Delta\phi(k)}{2}) + w_y(k) \\
 f_\phi(k) &= \phi(k+1) = \phi(k) + \Delta \phi(k) + w_\phi(k)
 \end{aligned} \tag{27}$$

gibi doğrusal olmayan hareket modeline sahip bir hedefin takibi için,

$$x_{k+1} = f(x_k, u_k) + w_k \tag{28}$$

$$z_{k+1} = h(x_k, u_k) + v_k$$

şeklindeki doğrusal olmayan durum ve gözlem denklemlerinin öncelikle doğrusallaştırılarak aşağıdaki

$$\begin{aligned}
 x_{k+1} &= A x_k + B u_k + w_k \\
 z_{k+1} &= H(x_k) + v_k
 \end{aligned} \tag{29}$$

formata dönüştürülmesi gerekir. Doğrusallaştırma işlemleri için Jacobien ifadeleri kullanılır. Örneğin yukarıda belirtilen denklemin A ve B matrislerinin Jacobien ifadeleri aşağıdaki denklemler ile belirtilir (Canan, 2004):

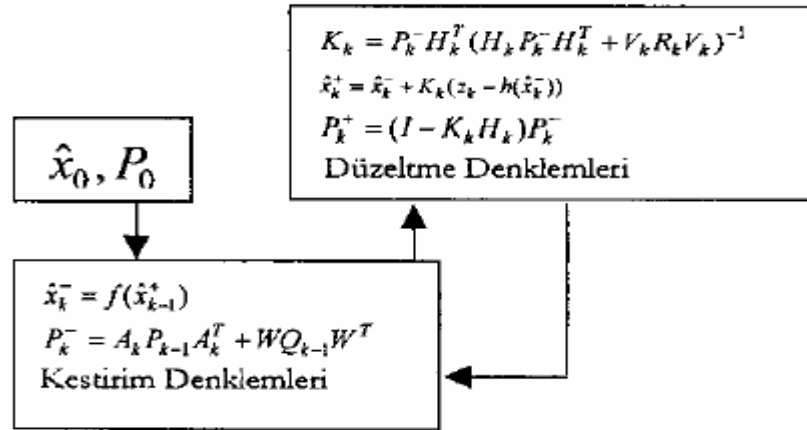
$$A = \begin{bmatrix} \frac{\partial f_x}{\partial x(k)} & \frac{\partial f_x}{\partial y(k)} & \frac{\partial f_x}{\partial \phi(k)} \\ \frac{\partial f_y}{\partial x(k)} & \frac{\partial f_y}{\partial y(k)} & \frac{\partial f_y}{\partial \phi(k)} \\ \frac{\partial f_\phi}{\partial x(k)} & \frac{\partial f_\phi}{\partial y(k)} & \frac{\partial f_\phi}{\partial \phi(k)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta D(k) \cdot \sin(\phi_{k-1} + \frac{\Delta\phi(k)}{2}) \\ 0 & 1 & \Delta D(k) \cdot \cos(\phi_{k-1} + \frac{\Delta\phi(k)}{2}) \\ 0 & 0 & 1 \end{bmatrix} \tag{30}$$

$$B = \begin{bmatrix} \frac{\partial f_x}{\partial \Delta D(k)} & \frac{\partial f_x}{\partial \Delta \phi(k)} \\ \frac{\partial f_y}{\partial \Delta D(k)} & \frac{\partial f_y}{\partial \Delta \phi(k)} \\ \frac{\partial f_\phi}{\partial \Delta D(k)} & \frac{\partial f_\phi}{\partial \Delta \phi(k)} \end{bmatrix} = \begin{bmatrix} \cos(\phi_k + \frac{\Delta\phi(k)}{2}) & \frac{-\Delta D(k)}{2} \sin(\phi_k + \frac{\Delta\phi(k)}{2}) \\ \sin(\phi_k + \frac{\Delta\phi(k)}{2}) & \frac{\Delta D(k)}{2} \cos(\phi_k + \frac{\Delta\phi(k)}{2}) \\ 0 & 1 \end{bmatrix} \tag{31}$$

Z_{k+1} ölçüm modelini X_k durum modeline bağlayan H matrisinin Jacobien denklemi aşağıda belirtilmektedir:

$$H = \begin{bmatrix} \frac{\partial h_x}{\partial h_x(k)} & \frac{\partial h_x}{\partial h_y(k)} & \frac{\partial h_x}{\partial h_\phi(k)} \\ \frac{\partial h_y}{\partial h_x(k)} & \frac{\partial h_y}{\partial h_y(k)} & \frac{\partial h_y}{\partial h_\phi(k)} \\ \frac{\partial h_\phi}{\partial h_x(k)} & \frac{\partial h_\phi}{\partial h_y(k)} & \frac{\partial h_\phi}{\partial h_\phi(k)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (32)$$

Doğrusallaştırma işlemi tamamlandıktan sonra artık Jacobien ifadeleri tahminleme ve kestirim işlemlerinde kullanılabilir. Şekil 7.1, Genişletilmiş Kalman Filtresinin çalışma prensibini göstermektedir (Canan, 2004):



Şekil 7.1 Genişletilmiş Kalman Filtresi Algoritmasının Genel İşleyişi

7.4 Parçacık Filtreleri

Parçacık Filtreleri, özyinelemeli Bayes tahminleme filtrelerinin bir türüdür ve ardışıl Monte Carlo metotlarına dayanmaktadır. Parçacık filtrelerindeki temel düşünce, dinamik sistemin olasılıksal dağılımını $p(x_k | z_{1:k})$, rasgele birbirinden bağımsız parçacıklar $\{x_k^i, i=1,2,...,N\}$ ile ifade etmektir. Bu yaklaşımda, her bir parçacığa bir ağırlık değeri $\{w_k^i, i=1,2,...,N\}$ atanır ve bu ağırlık değerleri o parçacığın tahminlemedeki önemini belirtir. Parçacıklar $x^i \approx q(x), i=1,..,N$ ile belirtilen önem derecesine göre üretilirler. Burada belirtilen X_k , sistemin t

anındaki durumunu, $Z_{1:k}$ başlangıçtan t anına kadar elde edilen gözlemleri ifade etmektedir (Evangelos et al., 2009).

Parçacık Filtreleme Algoritması, tahminleme ve güncelleme olmak üzere iki temel bölümden oluşur. Tahminleme aşamasında sonsal olasılık dağılımı tahminlenir ve güncelleme aşamasında, en son elde edilen ölçümler ile yapılan tahminleme güncellenir.

Parçacık Filtreleri yaklaşımında, N adet parçacık, başlangıçta, hedefin ilk konumu dikkate alınarak, $p(X_0 | Z_0)$ başlangıç dağılımına göre dağıtılırlar ve her birine eşit ağırlık değeri $1/N$ atanır. X_0 , izlemeyi başlatabilmek için gerekli olan ilk durum değeridir. Daha sonra parçacıklar, önsel dağılım fonksiyonuna $X_i^k \sim p(X_k | X_{k-1})$ göre yayılırlar. Diğer bir deyişle parçacıkların bir sonraki konumları $x_k^i = x_{k-1}^i + w_{k-1}^i$ $y_k^i = y_{k-1}^i + w_{k-1}^i$ ile ifade edilir. Parçacıklar izlenen hedefin dinamiklerine göre hareket ettirilir ve hedefe olan yakınlıklarına göre ağırlıklandırılırlar. Her bir parçacığa aşağıda belirtilen denklemler ile ağırlık değeri atanır:

$$w_k^i \propto \frac{p(X_k^i | Z_k)}{q(X_k^i | Z_k)} \quad (33)$$

Bu formülü açarsak aşağıdaki formülü elde ederiz (Arulampalam et al., 2001):

$$w_k^i \propto w_{k-1}^i \frac{p(Z_k | X_k^i) p(X_k^i | X_{k-1}^i)}{q(X_k^i | X_{k-1}^i, Z_k)} \quad (34)$$

Parçacıklara yeni ağırlık değerleri, bir önceki ağırlık değerleri ve yeni ölçüm değerleri göz önünde bulundurularak atanır. Bu formüllerden aşağıda belirtilen sonraki yoğunluk fonksiyonu elde edilir:

$$p(x_k | z_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i) \quad (35)$$

Son olarak, hedefin bir sonraki konumu, parçacıkların ağırlıklandırılmış konumlarının toplamı olarak tahminlenir:

$$\bar{x} = \sum_{i=1}^N w_k^i x_k^i \quad \bar{y} = \sum_{i=1}^N w_k^i y_k^i \quad (36)$$

Parçacık filtreleri yaklaşımında sıkça karşılaşılan bir problem de bozulma (degeneracy) problemidir. Tahminlemede kullanılan parçacıklardan bazıları, zamanla tahminlemedeki etkisini kaybeder. Bu parçacıkların, algoritmada kullanılması gereksiz işlem yükü getirecek ve bu da işlem maliyetini artıracaktır. Parçacıklardaki bozulma, etkin örnekleme sayısı N_{eff} değeri belirlenerek tespit edilebilir. Bu değer, aşağıdaki formülde de belirtildiği gibi, her bir parçacığın ağırlık değerlerinin karesinin toplamı ile hesaplanabilir:

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w_k^i)^2} \quad (37)$$

Bozulma problemini önleyebilmek için, N_{eff} değeri belirli bir eşik değeri N_{thr} ile karşılaştırılır. Eğer $N_{eff} < N_{thr}$ ise, ortamdaki işe yaramayan parçacık sayısı belirli bir eşik değerini geçmiştir ve parçacıkların yeniden örneklenmesi gerekmektedir. Yeniden örnekleme işleminde, yeni parçacıklar, mevcut parçacıklar kümesinden ağırlık değerleri ile ilişkili olarak örneklenirler ve mevcut parçacık kümesi, yeni parçacık kümesi ile yer değiştirir.

Algoritma 7.1 Genel Parçacık Filtresi Algoritmasının sözde kodunu göstermektedir (Ozfidan, et al. 2007).

Algoritma 7.1 Parcacik Filtresi Sozde Kodu

```

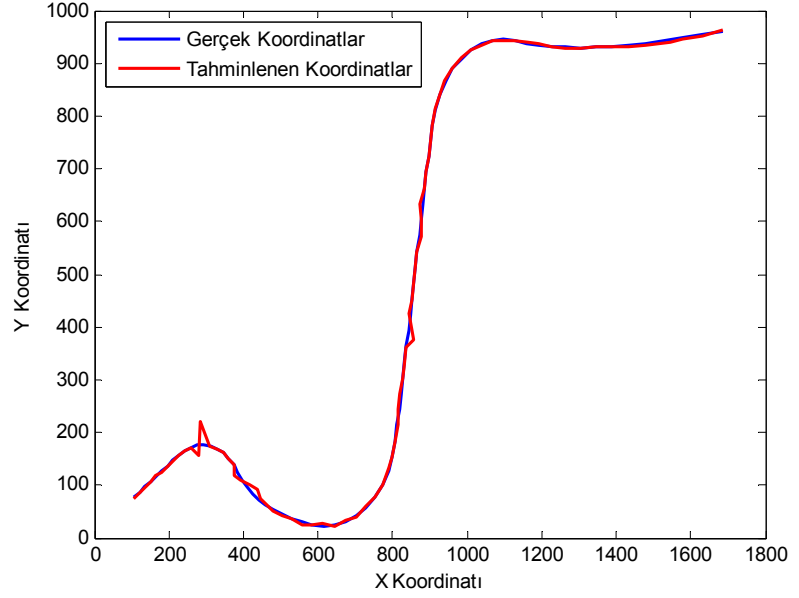
1: Baslangic
2:    $p(X_0|Z_0)$  dagilimi ile N parcacik ciz
3:   Her bir parcaciga  $1/N$  agirlik ata
4:   Ardisil Onem Ornekleme
5:    $X_k^i \sim p(X_k|X_{k-1})$  ile parcaciklari ornekle
6:   Airliliklari  $w_k^i \propto w_{k-1}^i p(Z_{n,k} | X_k^i)$  ile guncelle
7:   Normallestirme
8:   Airliliklari  $w_k^i = \frac{w_k^i}{\sum_{j=1}^N (w_k^j)}$  ile normallestir
9:   Yeniden Ornekleme
10:  Eger  $N_{eff} < N_{thr}$  ise parcaciklari yeniden ornekle

```

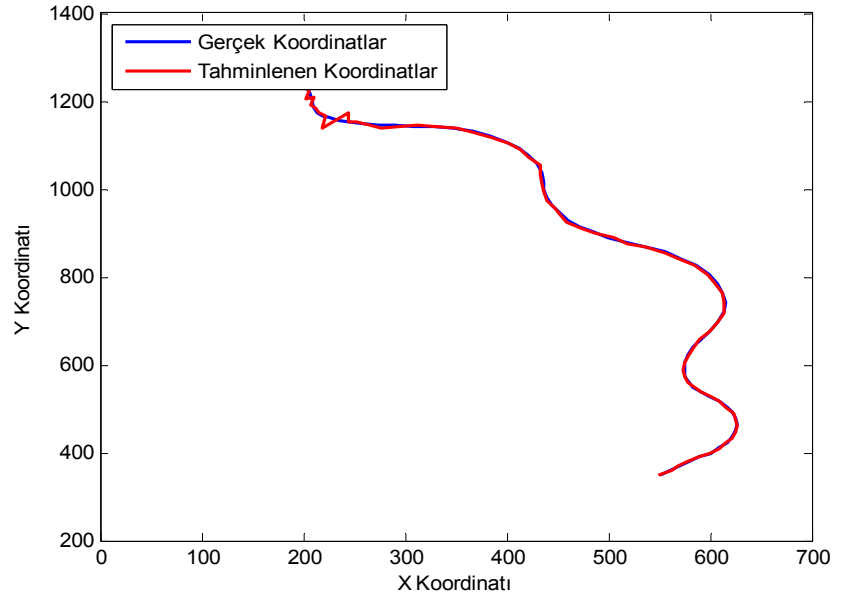
7.5 Simülasyon Sonuçları

Önerilen Dinamik Koni Tabanlı Hedef İzleme Algoritması ns-2.31 Simulator ortamında Parçacık ve Genişletilmiş Kalman Filtreleri ile gerçekleştirilmiştir. Simülasyon sonuçları rasgele doğrusal olmayan 30 adet hedef rotası üzerinde denenmiştir ve bu rotaların altı tanesi önermiş olduğumuz Parçacık Filtresi

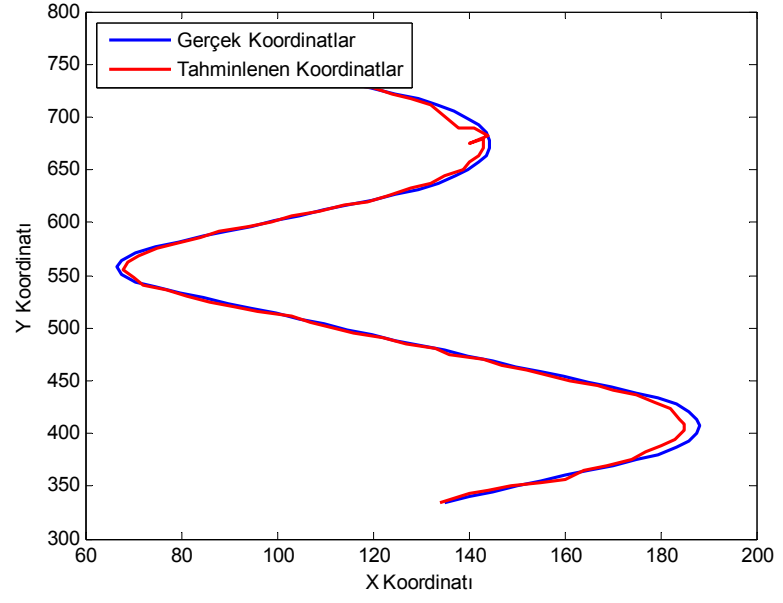
Tabanlı DKOİA Algoritmasının hedefi takip etmedeki doğruluğunu göstermek için Şekil7.2’de belirtilmiştir. Rotalar farklı hedef hızları, ivmeleri ve dönüş açıları içerecek şekilde rasgele oluşturulmuştur.



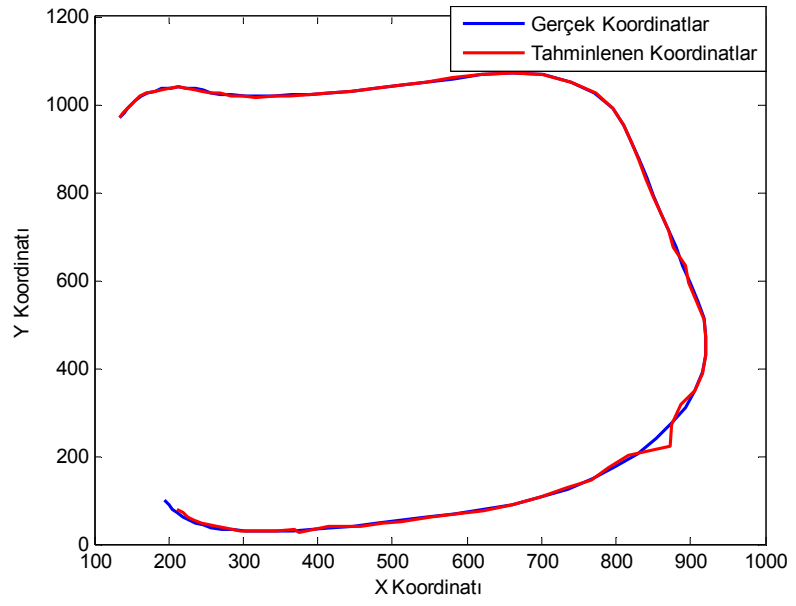
(a) Hareket1



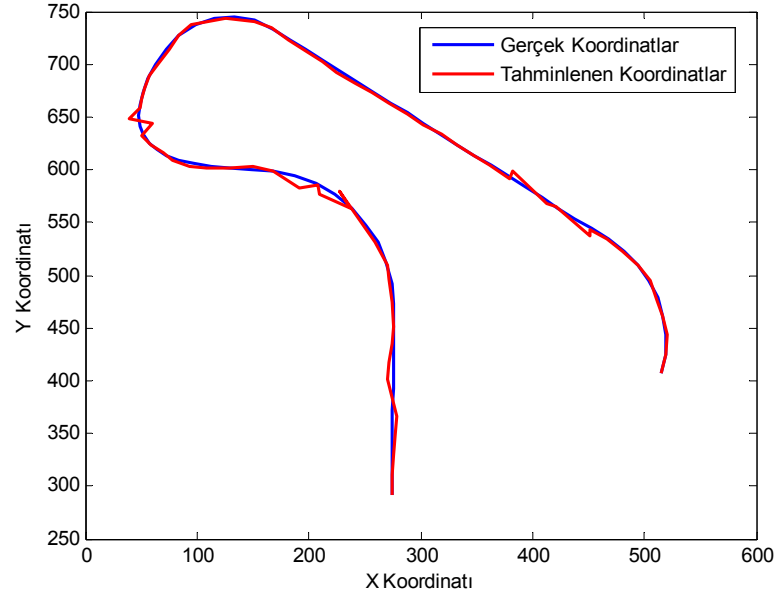
(b) Hareket2



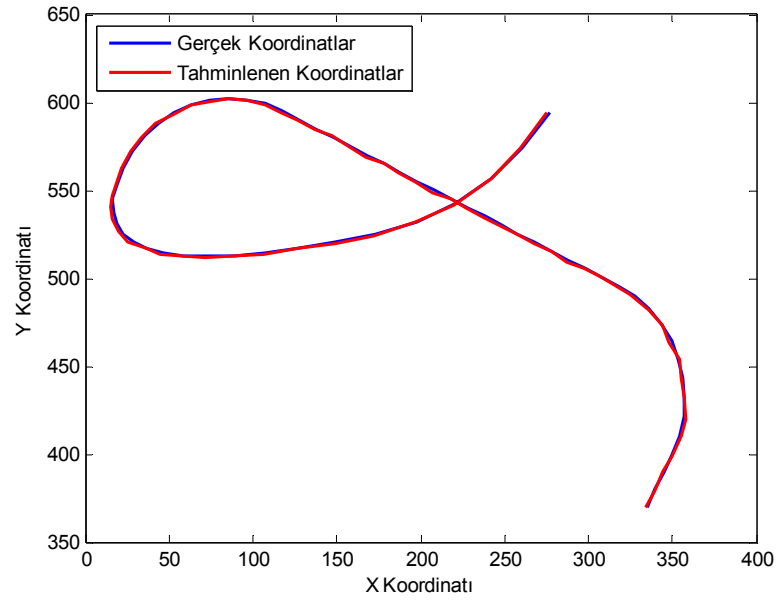
(c) Hareket3



(d) Hareket4



(e) Hareket5



(f) Hareket6

Şekil 7.2 Farklı Hareket Rotalarının PF_DKOİA ile Takip Edilmesi

Hareket1’de, hedef x ve y yönlerinde 15 ve 10 m/s hızlarla harekete başlamıştır. Daha sonra 9° ’lik açıyla dönüş yapıp, x ve y yönlerinde 2 ve 1 m/s^2 ’lik ivmeli hareket yapmıştır. Tekrar 9° ’lik açıyla dönüş yapıp, bir süre sabit hızla ilerlemiştir. Son olarak 1 m/s^2 ’lik ivmelenme ve sonrasında da 6° ’lik açıyla dönüş yaparak hareketini tamamlamıştır.

Hareket2’de, hedef x ve y yönlerinde 10 m/s’lik hızlarla harekete geçmiştir. Daha sonra -9° ’lik bir açıyla dönüş yapıp, x ve y yönlerinde -0.5 ve -1 m/s^2 ’lik ivmeli hareket yapmıştır. Devamında 9° ’lik bir açıyla dönüş yapıp, bir süre sabit hızla ilerlemiştir. Son olarak x ve y yönlerinde 1 ve -1 m/s^2 ’lik ivmelenme ve sonrasında da 9° ’lik açıyla dönüş hareketi yapmıştır.

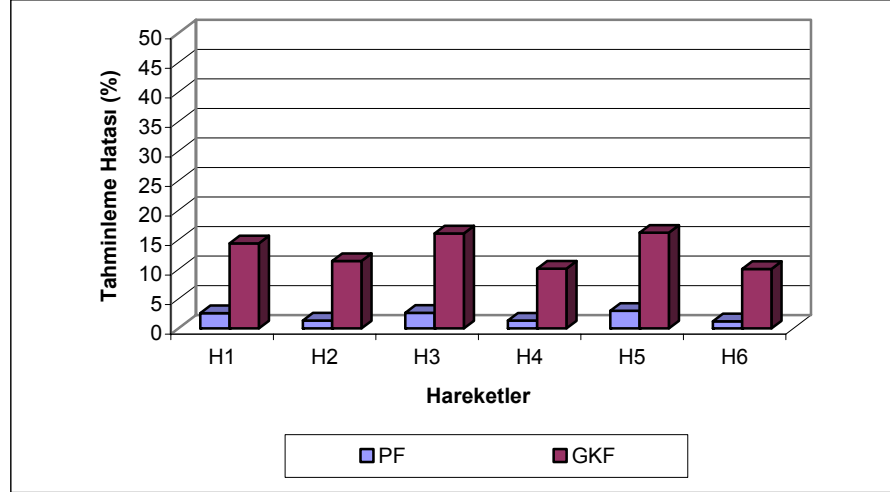
Hareket3 sadece dönüş ve sabit hızlı hareketler içermektedir. Hedef öncelikle 5 m/s hızla harekete geçmiştir. Daha sonra 9° ’lik bir açıyla dönüş yapıp, bir süre sabit hızla ilerlemiştir. Devamında -15° ’lik açıyla dönüş yapıp, bir süre sabit hızla ilerlemiştir. Son olarak $5,72^\circ$ ’lik bir açıyla dönüşün arkasından sabit hızla ilerleyerek hareketini tamamlamıştır.

Hareket4’te, hedef x ve y yönlerinde 5 ve 10 m/s’lik hızlarla harekete geçmiştir. Daha sonra -9° ’lik bir açıyla dönüş yapıp, x ve y yönlerinde -1 ve -1 m/s^2 ’lik ivmeli hareket yapmıştır. Devamında $-14,89^\circ$ ’lik bir açıyla dönüş yapıp, bir süre sabit hızla ilerlemiştir. -6° ’lik bir dönüşün ardından, x ve y yönlerinde -1 ve 1 m/s^2 ’lik ivmelenme ve sonrasında da -6° ’lik açıyla dönüş hareketi yapmıştır.

Hareket5’te, hedef x ve y yönlerinde 10 ve 15 m/s’lik hızlarla harekete geçmiştir. Daha sonra 9° ’lik bir açıyla dönüş yapıp, bir süre sabit hızla hareket etmiştir. Devamında $14,89^\circ$ ’lik bir açıyla dönüş yapıp, x ve y yönlerinde 1 ve 1 m/s^2 ’lik ivmelenme hareketi yapmıştır. -9° ’lik bir dönüşün ardından, bir süre daha sabit hızla ilerledikten sonra tekrar -9° ’lik açıyla dönüş yaparak hareketini tamamlamıştır.

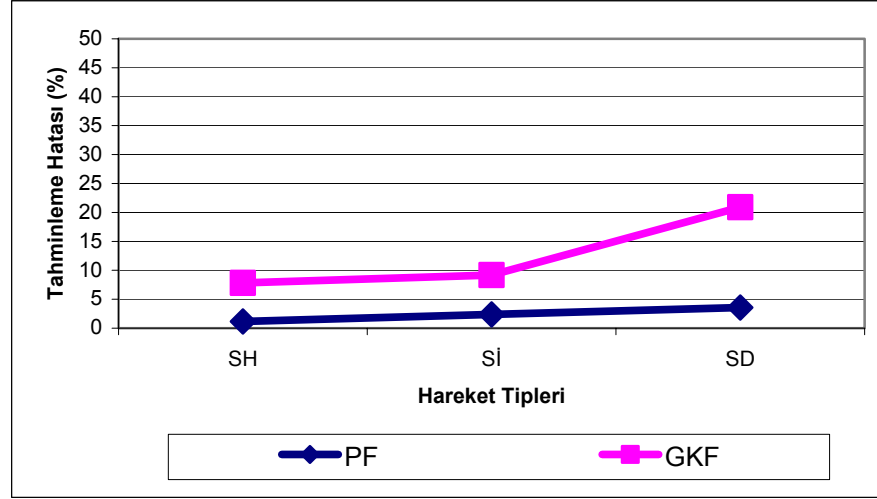
Hareket6’da, hedef x ve y yönlerinde 5 ve 10 m/s’lik hızlarla harekete geçmiştir. Daha sonra 9° ’lik bir açıyla dönüş yapıp, bir süre sabit hızla hareket etmiştir. Devamında tekrar 9° ’lik bir açıyla dönüş yaparak, x ve y yönlerinde 1 ve -1 m/s^2 ’lik ivmelenme hareketi yapmıştır. 9° ’lik bir dönüşün ardından, x ve y yönlerinde 2 ve 1 m/s^2 ’lik ivmelenme hareketi yaptıktan sonra -15° ’lik açıyla dönüş yaparak hareketini tamamlamıştır.

Şekil7.2’de hedef takibi için kullanmış olduğumuz rotaların altı tanesi gösterilmektedir. Şekilden de görülebildiği gibi, her bir rota için önermiş olduğumuz Parçacık Filtresi Tabanlı DKOİA Algoritması hedef takip edilebilirliğinde iyi sonuç vermektedir.



Şekil 7.3 PF ile GKF’nin Hedef Takibinde Farklı Hareketler için Tahminleme Hataları

Genişletilmiş Kalman ve Parçacık Filtrelerinin her ikisi de DKOİA algoritmasına entegre edilmiş, farklı hedef rotaları için elde edilen tahminleme hataları Şekil 7.3 ‘te belirtilmiştir. Sonuçlardan görülmektedir ki, her bir hedef rotası için Parçacık Filtresi, Genişletilmiş Kalman Filtresine göre çok iyi performans göstermektedir. Şekil 7.3’e göre, Parçacık Filtresindeki tahminleme hatası en kötü durumda % 2.97 iken, bu değer Genişletilmiş Kalman Filtresinde %15 gibi oranlara çıkabilmektedir.



Şekil 7.4 PF ile GKF'nin Hedef Takibinde Farklı Hareket Tipleri için Tahminleme Hataları

Şekil 7.4, Parçacık Filtresi ile Genişletilmiş Kalman Filtresini sabit hızlı, sabit ivmeli ve sabit dönüşlü farklı hareket tiplerine göre tahminleme hatalarını karşılaştırmaktadır. Her bir hareket tipi için Parçacık Filtresinin tahminleme hatası Genişletilmiş Kalman Filtresine göre oldukça düşüktür. Genişletilmiş Kalman Filtresinin özellikle dönüş modelindeki tahminleme hatasının büyüklüğü açıkça görülebilmektedir. Bunun nedeni, Genişletilmiş Kalman Filtresindeki yaklaşım olan, doğrusal olmayan modelin doğrusallaştırılması esnasındaki gerçekleşen bozulmalardır.

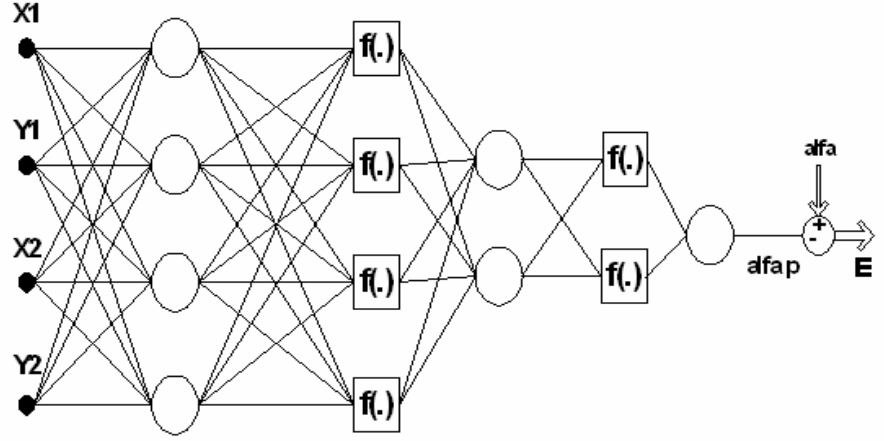
7.6 Yan Çalışmalar

Tez kapsamında tahminlemede kullanılması için Dağıtık Yapay Sinir Ağı ve Bezier Spline olmak üzere iki farklı yöntem daha incelenmiş fakat yüksek enerji kaybına neden olması ve iyi tahminleme performansı göstermemesinden dolayı kullanımı tercih edilmemiştir.

7.6.1 Dağıtık Yapay Sinir Ağı Modeli

Yapay sinir ağları (YSA), öğrenme yolu ile yeni bilgiler türetebilme, oluşturabilme ve keşfedebilme gibi yetenekleri, otomatik olarak gerçekleştirebilmek amacı ile geliştirilen bilgisayar sistemleridir (Öztemel, 2003). YSA'larda karmaşık işlevler bir çok nöronun eş zamanlı çalışması ile meydana getirilir.

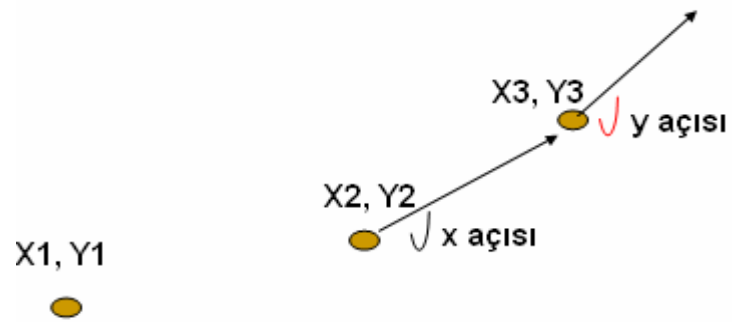
Şekil 7.5'te gösterilen, tasarlanmış olduğumuz ysa modeli 4 katmanlıdır ve 7 nöron içermektedir. YSA modelimiz geriye beslemeli bir yapıya sahiptir ve güncellemeler için gradyan yöntemini kullanmaktadır.



Şekil 7.5 Hedef Takibinde Tahminleme için Tasarlanan YSA Modeli

Telsiz duyarga ağı ortamı dağıtık bir yapıya sahip olduğu için önermiş olduğumuz modelin de dağıtık bir yapıya sahip olması gerekir. Yani hedef takip sistemimizdeki her bir lider/kök düğüm birbirlerinden bağımsız olarak dağıtık ysa modelini kullanarak hedefin bir sonraki konumu ile ilgili çıkarsama yapabilmelidir.

Şekil 7.6'da dağıtık yapay sinir ağı modelinin hedef takibinde nasıl kullanmayı planladığımız gösterilmektedir.



Şekil 7.6 Dağıtık YSA'nın Hedef Takibinde Kullanımı

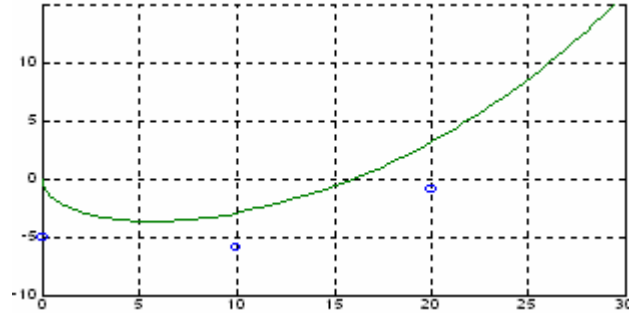
Şekil 7.6'da görülen $(X1, Y1)$, $(X2, Y2)$ ve $(X3, Y3)$ hedefin üç farklı zamanda hesaplanan konumlarıdır. Mevcut lider / kök düğüm, hedefin $(X1, Y1)$ ve $(X2, Y2)$ konum bilgilerini dağıtık yapay sinir ağı sistemine verir ve x açısı ile sistemi

eğitir. Daha sonra sisteme (X2,Y2) ve (X3,Y3) konum bilgilerini vererek y açısını hesaplamaya çalışır.

YSA'nın doğru çıkarsama yapabilmesi için, çok fazla veri ile önceden eğitilmiş olması gerekir ve ayrıca işlem yükü çok fazladır. Hem YSA modelinin eğitimi için yeterli bilginin bulunmaması, hem de duyargaların enerji sınırının olması nedeniyle, YSA'ların tahminleme işlemi için telsiz duyarga ağlarında kullanımı uygun değildir.

7.6.2 Bezier Spline Yöntemi

Tahminleme işlemi için kullanmayı planlamış olduğumuz başka bir yöntem de Bezier Spline yöntemidir. Bu yöntem ile, lider / kök düğümün, hedefin üç konum bilgisinden geçen eğriyi bulması ve bu eğrinin nasıl devam edeceğini tahminlemesi hedeflenmektedir. 0 ile 1 arasında değer alan u değerine tez çalışmasında 1.5 değeri verilerek eğrinin belirlenen noktalardan sonra da devamı sağlanır. Şekil 7.7'de Bezier Spline yöntemi ile noktalardan geçen eğrinin bulunması işlemi gösterilmektedir.



Şekil 7.7 Bezier Spline Yöntemi ile Noktalardan Geçen Eğrinin Bulunması

Bezier Spline yönteminde, $n+1$ kontrol noktası vardır (k , 0'dan n 'e) ve $P_k=(x_k,y_k,z_k)$ 'dır. Bu kontrol noktaları $P(u)$ konum vektörünü oluşturacak şekilde harmanlanır. $P(u)$, p_0 ile p_n arasındaki yaklaşık Bezier polinom fonksiyonudur (Uğur, 2006). Aşağıdaki formüller ile hesaplanır:

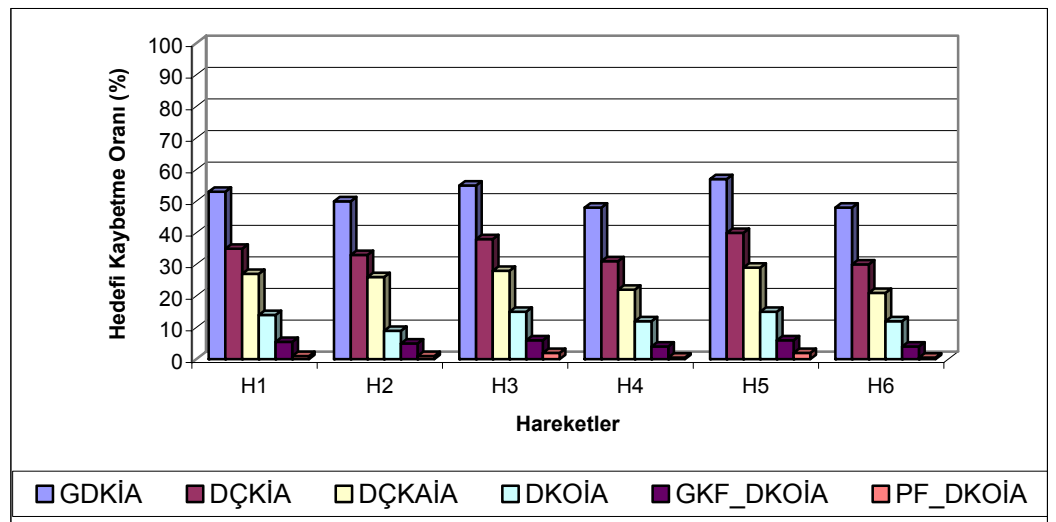
$$P(u) = \sum_{k=0}^n p_k BEZ_{k,n}(u) \quad 0 \leq u \leq 1 \quad (38)$$

$$BEZ_{k,n}(u) = C(n,k)u^k(1-u)^{n-k} \quad C(n,k) = \frac{n!}{k!(n-k)!}$$

Bezier Spline yöntemi, basit uygulanabilirliği ve işlem yükünün az olması gibi avantajları vardır. Fakat araştırmalarımızın sonucunda bu yöntemin daha çok belirlenen noktalardan geçebilecek eğrinin hesaplanmasında kullanılmakta olduğunu, eğrinin uzatılması için ekstrapolasyon gibi başka yöntemlerin kullanılmasının daha uygun olduğunu gördük.

8 SONUÇ ve TARTIŞMA

Tezdeki amacımız telsiz duyarga ağlarında çok hızlı hareket eden hedeflerin takibinde, hedefi kaybetme oranını minimuma düşürecek algoritmalar önermektir. Bunun için de tez kapsamında, kablosuz duyarga ağları için dört farklı dağıtık hedef takibi algoritmaları önerilmiştir. Bunlardan ilk üçü DÇKİA, DÇKAİA ve DKOİA çok hızlı, doğrusal hareket eden nesnelerin takibi için gerçekleştirilmiş ve simülasyon sonuçları alınmıştır. Önermiş olduğumuz bu üç algortmadan çok hızlı, doğrusal hareket eden hedefin takibinde en iyi performansı gösteren DKOİA algoritmasına Parçacık Filtreleri entegre edilerek, dördüncü algortmamız olan PF_DKOİA algoritması önerilmiştir. Bu algoritma ile doğrusal olmayan hareketler yapan hedefin takibinin gerçekleştirilmesi hedeflenmiştir. Simülasyon sonuçlarından PF_DKOİA algoritmasının manevra yapan hedefin takibinde çok iyi performans gösterdiği görülmüştür. Doğrusal olmayan hedef takibinde sıkça kullanılan bir başka yöntem Genişletilmiş Kalman Filtresi'dir. DKOİA algoritmasına, Genişletilmiş Kalman Filtreleri de eklenmiş (GKF_DKOİA) ve simülasyon sonuçları alınarak Parçacık Filtreleri ile karşılaştırılmıştır. Simülasyon sonuçlarından Genişletilmiş Kalman Filtresinin tahminlemedeki hata oranının, Parçacık Filtrelerine göre oldukça yüksek olduğu görülmüştür. Genişletilmiş Kalman Filtrelerindeki tahminleme hata oranı yüksekliği, özellikle dönüşlü hareket tiplerinde daha net görülebilmektedir. Bunun nedeni, Genişletilmiş Kalman Filtresinin, doğrusal olmayan hareket modellerini doğrusallaştırması esnasındaki bozulmalardır.



Şekil 8.1 Doğrusal Olmayan Hareketler için Önerilen Bütün Hedef Takip Algoritmalarının Hedefi Kaybetme Oranlarının Karşılaştırılması

Şekil 8.1’de Genel küme tabanlı yaklaşım GDKİA ile bütün önerilen algoritmalar DÇKİA, DÇKAİA, DKOİA, GKF_DKOİA ve PF_DKOİA, doğrusal olmayan hareketler yapan hedef için gerçekleştirilip, bu algoritmaların hedefi kaybetme oranları karşılaştırılmıştır. DÇKİA, DÇKAİA ve DKOİA’da doğrusal tahminleme yapılırken, DKOİA’ya GKF ve PF olmak üzere iki farklı tahminleme mekanizması entegre edilmiştir. Simülasyon sonuçları PF_DKOİA algoritmasının tahminlemedeki başarımının yüksek olması nedeniyle, hedefi kaybetme olasılığının oldukça düşük olduğunu göstermiştir. GKF_DKOİA algoritması, PF_DKOİA’ya göre daha düşük performans göstermesine rağmen, DÇKİA, DÇKAİA ve DKOİA algoritmalarına göre çok daha iyi sonuç vermektedir. Bunun nedeni, doğrusal olmayan hareketleri bir sonraki konumunu belirleyebilmesi için bir tahminleme alt yapısı içermesidir. PF_DKOİA’ya göre daha kötü sonuç vermesinin nedeni ise dönüşlerdeki tahminleme hatalarıdır. Hedefin aniden dönmesi ya da ivmelenmesi durumunda, doğrusal tahminleme yapan DÇKİA, DÇKAİA ve DKOİA algoritmaları hedefi kaybetme olasılığı için iyi sonuçlar vermemişlerdir. DKOİA algoritmasının, DÇKİA ve DÇKAİA algoritmasına göre daha iyi sonuç vermesinin nedeni, hedefin ani yön değiştirmesi yada ivmelenmesi durumunda, koni içinde oluşturulan kapsama ağaçlarından birinin hedefi yakalama olasılığının DÇKİA ve DÇKAİA algoritmalarına göre yüksek olmasıdır. Benzer sebeple, DÇKAİA algoritması, kümelendirilmiş kapsama ağacı içinde bulunan düğüm sayısının DÇKİA algoritmasına göre daha fazla olması nedeniyle, hedefi kaybetme olasılığı daha düşüktür. En düşük performanslı sonuç ise genel kümeleme yaklaşımı olan GDKİA’da alınmıştır. Hedefin manevralar yapıyor olması ve doğrusal tahminleme ile sadece $t+1$ anındaki kümenin oluşturuluyor olması nedeniyle, GDKİA hedefi takip etmede düşük performans göstermiştir.

Sonuç olarak, tez kapsamında önermiş olduğumuz bütün algoritmalar, çok hızlı hareket eden hedeflerin takibinde çok iyi performans göstermektedirler. Özellikle de literatürdeki genel yaklaşım ile karşılaştırdığımızda, genel yaklaşımda hedefi kaybetme oranı oldukça yüksek iken, önermiş olduğumuz yaklaşımlarda genel yaklaşıma göre çok daha iyi sonuçlar alınmıştır. Tez kapsamında, doğrusal hareket yapan hedeflerin takibine ek olarak, doğrusal olmayan hareketler yapan hedeflerin takibi için de algoritma önerilmiştir. Bunun için de literatürde var olan iki yaygın tahminleme algoritması, önerilen hedef takip sistemine entegre edilip, simülasyon sonuçlarının karşılaştırılması yapılmıştır. Simülasyon sonuçlarından Parçacık Filtrelerinin tahminlemedeki hata oranının, Genişletilmiş Kalman Filtrelerine göre daha düşük olduğu görülmüştür. Sonuç olarak, önermiş

olduğumuz PF_DKOİA algoritması, doğrusal olmayan hareketler yapan hedef takibinde oldukça iyi sonuç vermektedir. Çizelge 8.1, Çizelge 8.2 ve Çizelge 8.3 tez kapsamında önerilen ve gerçekleştirimi yapılan algoritmaların farklı hız ve hareket tipleri için genel bir karşılaştırmasını göstermektedir.

Çizelge 8.1. Doğrusal Hareketler için Algoritmaların Farklı Hız Değerleri için Karşılaştırılması

	30-40 m/s	50-60 m/s	70-80 m/s	90-100 m/s
GDKİA	Çok kötü	Çok kötü	Çok kötü	Çok kötü
GDKAİA	Kötü	Kötü	Kötü	Kötü
DÇKİA	Orta	Orta	Orta	Orta
DÇKAİA	İyi	İyi	İyi	İyi
DKOİA	Çok iyi	Çok iyi	Çok iyi	Çok iyi

Çizelge 8.1’de de görüldüğü gibi çok yüksek hızlarda hareket eden hedefin takibinde en iyi sonucu önermiş olduğumuz DKOİA’da alırken, en kötü sonuçlar ise mevcut yaklaşımlar olan GDKİA ve GDKAİA’da alınmaktadır.

Çizelge 8.2. Doğrusal Olmayan Hareketler için Algoritmaların Karşılaştırılması

	Çok Kötü	Kötü	Orta	İyi	Çok iyi
GDKİA	X				
GDKAİA	X				
DÇKİA		X			
DÇKAİA			X		
DKOİA			X		
GKF_DKOİA				X	
PF_DKOİA					X

Çizelge 8.2’de doğrusal olmayan hareketler için önermiş olduğumuz algoritmaların karşılaştırmaları yapılmıştır. Çizelgeden de görüldüğü gibi en iyi performansı PF_DKOİA gösterirken, en kötü performansı GDKİA göstermektedir.

Çizelge 8.3. Tahminleme Mekanizmalarının Farklı Hareket Türleri için Karşılaştırılması

	Sabit Hız	Sabit İvme	Sabit Dönüş
Genişletilmiş Kalman Filtreleri	Çok iyi	İyi	Orta
Parçacık Filtreleri	Çok iyi	Çok iyi	Çok iyi

Çizelge 8.3’de tez kapsamında kullanmış olduğumuz tahminleme mekanizmaları olan Genişletilmiş Kalman Filtreleri ile Parçacık Filtrelerinin tahminlemedeki hata

oranları bazında karşılaştırmaları yapılmıştır. Buna göre, her bir hareket türü için Parçacık Filtrelerinin çok iyi bir performans gösterdiği, Genişletilmiş Kalman Filtrelerinin ise özellikle dönüşlü hareketlerde orta düzeyde performans gösterdiği görülmektedir.

Tez kapsamında geliştirmiş olduğumuz sistem, alana giren çeşitli hedeflerin izlenmesi amacıyla kullanılabilir. Hedef takip uygulamaları, askeri uygulamalar başta olmak üzere, çevresel, güvenlik gibi bir çok alanda kullanılmaktadır. Askeri uygulamalarda özellikle savaş alanının ve düşman askerinin izlenmesi, sınır bölgelerine yaklaşan hedeflerin tespit edilmesi ve uyarılması, çevresel uygulamalarda belirli bir doğal ortamda yaşayan hayvanların takip edilmesi, sel felaketinin tespit edilmesi, güvenlik uygulamalarında araba hırsızlıklarının ve hız sınırını aşan araçların tespit edilmesi ve takibi gibi amaçlar için kullanılmaktadır. Tez kapsamında 30 m/s 'den 100m/s hızlara kadar çıkabilen ve manevralar yapan hedefin takibini başarıyla sağlayan bir sistem geliştirilmiştir. Bu sistemin askeri ve sivil alanlarda çok hızlı hareket eden hedeflerin tespiti ve takibi amacıyla için kullanılabileceği düşüncesindeyiz. Şu an geliştirmiş olduğumuz sistemin karada kullanılacağı varsayılmıştır. Ancak su altında kullanılabilen duyargalar da mevcuttur. Geliştirilen sistem bu tür duyargalara da uyarlanabilir.

Gelecek çalışmalarımızda önermiş olduğumuz hedef takip sistemini aşağıdaki amaçlar için geliştirmeyi planlamaktayız:

- Birden çok hedefin takibi
- Hedeflerin sınıflandırılması
- Yeni tahminleme mekanizmalarının sisteme entegre edilmesi
- Hedef takibinin üç boyutlu ortamlara taşınması
- Önerilen hedef takip sisteminin su altı duyargalar ile de gerçekleştirilmesi
- Simülasyonların gerçek ortamda test edilmesi.

9 KAYNAKLAR DİZİNİ

Abbasi, A. and Younis, M., 2007, A survey on clustering algorithms for wireless sensor networks. In the Journal of Computer Communications, Vol. 30, s. 2826-2841.

Akkaya, K. and Younis M., 2005, A survey on routing protocols for wireless sensor networks. In the Journal of Ad Hoc Networks, Vol. 3, No.3, s. 325-349.

Alaybeyoglu, A., Dagdeviren, O., Erciyes, K., Kantarci, A., 2009, Performance Evaluation of Cluster-based Target Tracking Protocols for Wireless Sensor Networks. In Proceedings of the 24th International Symposium on Computer and Information Sciences.

Altındış N., 2002, Bayes Karar Teorisi, Ders Notu.

Arulampalam, S., Maskell, S., Gordon, N., Clapp, T., 2001, A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. IEEE Transactions on Signal Processing, Vol. 50, s.174-188.

Brooks, R.R., Ramanathan P., Sayeed, A.M., 2003, Distributed Target Classification and Tracking in Sensor Networks. In Proceedings of the IEEE, Vol. 91, s.1163-1171, Issue: 8.

Brooks, R. R., Griffin, C., Friedlander, D. S., 2002, Self-organized Distributed Sensor Network Entity Tracking. International Journal of High Performance Computing Applications, Vol.16, No.3, s. 207-219.

Bulusu, N., Estrin, D., Girod, L., Heidemann, J., 2001, Scalable Coordination for Wireless Sensor Networks: Self-Configuring Localization Systems. In Proceedings of 6th International Symposium on Communication Theory and Application, Ambleside.

Chen,W., Hou, J., 2004, Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks. IEEE Transactions on Mobile Computing, s. 258-271.

KAYNAKLAR DİZİNİ (devam)

Canan, S., Akkaya, R., Ergintav, S., 2004, Genişletilmiş Kalman Filtresi ile Algılayıcı Tümlleştirilmesi ve Gezer Robotlara Uygulaması. Sinyal İşleme Kurultayı, Korumar Otel, Kuşadası, İzmir.

Evangelos B., Mazomenos, J., Reeve, S., White M., 2009, A Range-Only Tracking Algorithm for Wireless Sensor Networks. In Proceedings of the International Conference on Advanced Information Networking and Applications Workshops, s. 775-780.

Fall, K. and Varadhan, K., The ns manual, 2007. http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf. (Erişim Tarihi: 10 Ekim 2009)

Goshorn, R., Goshorn, J., Goshorn D., Aghajan, H., 2007, Architecture for Cluster Based Automated Surveillance Network for Detecting and Tracking Multiple Persons. IEEE International Conference on Distributed Smart Cameras, s. 219-226.

Gui, C., Mohapatra, P., 2004, Power conservation and quality of surveillance in target tracking sensor networks. In Proceedings of ACM International Conference on Mobile Computing and Networking, s. 129-143.

R. Gupta, S. R. Das, 2003, Tracking moving targets in a smart sensor network. In Proceedings of IEEE VTC Symposium., Vol. 5, s. 3035-3039.

Halvardsson, M. and Lindberg, P., 2004, Reliable group communication in a military ad hoc network. Technical report, Vaxjo University.

Heinzelman, W., Kulik, J., Balakrishnan, H., 1999, Adaptive protocols for information dissemination in wireless sensor networks. In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, s. 174-185.

Heinzelman, W., Chandrakasan, A., Balakrishnan, H., 2000, Energy efficient communication protocol for wireless sensor networks. In Proceeding of the Hawaii International Conference System Sciences.

KAYNAKLAR DİZİNİ (devam)

Hightower, J. and Borriello, G., 2001, A Survey and Taxonomy of Location Systems for Ubiquitous Computing Technical Report. University of Washington, Computer Science and Engineering, Seattle, WA.

Intanagonwiwat, C., Govindan, R., Estrin, D., 2000, Directed diffusion: a scalable and robust communication paradigm for sensor networks. In Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking, s. 56-67.

Jeong J., Hwang T., He Ti., Du D., 2007, MCTA: Target Tracking Algorithm based on Minimal Contour in Wireless Sensor Networks, IEEE INFOCOM , s. 2371-2375

Kambhampati S.,2008, Target/Object Tracking Using Particle Filtering, PHD Thesis.

Kuang X., Feng R., Shao H., 2008, A lightweight target tracking scheme using wireless sensor network, Measurement Science and Technology, vol. 19, .

Lana A., 2001, Kalman Filtresi ve Olasılıksal Veri İlişkilendirme Yöntemlerini Kullanan Çoklu Algoritmaları, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 77s.

Li D., Wong K. D., Hu Y. H., Sayeed A. M., 2002, Detection, Classification and Tracking of Targets in Distributed Sensor Networks, IEEE Signal Processing Magazine, s. 17-29.

Lindsey S., Raghavendra C.S., 2002, PEGASIS: power efficient gathering in sensor information systems. In Proceedings of the IEEE Aerospace Conference, vol.3, s.1125-1130.

Mechitov K., Sundresh S., Kwon Y., Agha G., 2003, Cooperative Tracking with Binary-Detection Sensory Networks, ACM Sensys 03, s. 332-333.

Mühlich M., 2003, Particle Filters an overview, Talk at Filter Worskshop, Bucharest, Romania.

KAYNAKLAR DİZİNİ (devam)

Nasipuri A., Li K., 2002., A Directionality based Location Discovery Scheme for Wireless Sensor Networks, WSNA'02, s. 105-111.

Olule E., Wang G., Guo M., Dong M., 2007, RARE: An Energy-Efficient Target Tracking Protocol for Wireless Sensor Networks. In the Proceedings of the International Conference on Parallel Processing Workshops (ICPPW 07), s. 76.

Özfidan, Ö., Bayazit, U., Cirpan, H. A., 2007, Cluster Based Sensor Scheduling in a Target Tracking Application with Particle Filtering. In Proceedings of the IEEE International Symposium on Industrial Electronics.

Öztemel E., 2003, Yapay Sinir Ağları, Papatya Yayıncılık, İstanbul.

Öztürk, F., 2008, Polinom İnterpolasyonu.

<http://80.251.40.59/science.ankara.edu.tr/ozturk/Dersler/ist310/Ders3/interpolasyon.doc> (Erişim Tarihi: 10 Ekim 2009)

Ray, S., Ungrangsi R., Pellegrini F., Trachtenberg A., Starobinski D., Robust Location Detection in Emergency Sensor Networks, 2002, INFOCOM'03, vol. 2, s. 207-219.

Ren S., Li Q., Wang H., Zhang X., 2005, Design and Analysis of Wave Sensing Scheduling Protocols for Object Tracking Applications, DCOSS 05, s. 228-243.

Ren S., Li Q., Wang H., Chen X., Zhang X., 2005, Analyzing object detection quality under probabilistic coverage in sensor networks, IWQoS'05, s. 107-122.

Rodoplu V. and Ming T.H., 1999, Minimum energy mobile wireless networks, IEEE Journal of Selected Areas in Communications , vol. 17, no. 8, s. 1333–1344.

Schurgers C., Srivastava M.B., 2001, Energy efficient routing in wireless sensor networks. In The MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, s. 357-361.

Serper, Ö., 1992, Uygulamalı İstatistik II, Filiz Yayınevi, s. 233

KAYNAKLAR DİZİNİ (devam)

Simon G., Ledezczi A., Maroti, M., 2004, Sensor Network-Based Countersniper System, SenSys04, ACM Press, s. 1-12, 2004.

Suganya, S., 2008, A Cluster-Based Approach for Collaborative Target Tracking in Wireless Sensor Networks , ICETET'08, s. 276-281.

Tasdemir, Ç., Kent, S., 2006, Radarla Hedef İzleme, 3.URSI Ulusal Bilimsel Kongresi.

Thorstensen, B., Syversen, T, Walseth T., Bjornvold,T., 2004, Electronic Shepherd: A Low-Cost, Low-Bandwidth, Wireless Network System. In the Proceeding of Second International. Conference on Mobile Conference on Mobile Systems, Applications and Services, ACM Press, s. 245-255.

Tseng Y., Kuo S., Lee H., Huang C., 2003, Location Tracking in a Wireless Sensor network by Mobile Agents and its Data Fusion Strategies, International Workshop on Information Processing in Sensor Networks (IPSN 03), s. 625-641.

Uğur A., 2006, Bezier Eğrileri ve Yüzeyleri, yzgrafik.ege.edu.tr/~ugur/07_08.../3DViewing_Illumination.ppt (Erişim Tarihi: 10 Ekim 2009)

Wang X, Ma JJ, Wang S, Bi DW, 2007, Cluster-based dynamic energy management for collaborative target tracking in wireless sensor networks, International Journal of Sensors, vol. 7, s. 1193-1215.

Wang, Z., Li, H., Shen, X., Sun, X., Wang, Z., 2008, Tracking and Predicting Moving Targets in Hierarchical Sensor Networks, IEEE International Conference on Networking, Sensing and Control (ICNSC 08), s. 1169-1173.

Wattenhofer R., 2006, Sensor Networks: Distributed Algorithms Reloaded-or Revolutions, SIROCCO '06, s. 24-28.

Welch G.and Bishop G., 2001, An Introduction to the Kalman Filter, SIGGRAPH – Course.

KAYNAKLAR DİZİNİ (devam)

Xu Y., Heidemann J., Estrin D., 2001, Geography-informed energy conservation for ad hoc routing In Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'01), s. 20-84.

Xu Y., Lee W., 2007, Compressing Moving Object Trajectory in Wireless Sensor Networks, International Journal of Distributed Sensor Networks, Volume [3](#), Issue 2 , s. 151 – 174.

Xu, Y., Lee, W., 2003, On Localized Prediction for Power Efficient Object Tracking in Sensor Networks. In Proceedings of the 23rd International Conference on Distributed Computing Systems, s.434.

Xu, Y., Winter, J., Lee, W., 2004, Prediction-based Strategies for Energy Saving in Object Tracking in Sensor Networks, Mobile Data Management (MDM'04), s. 346-357.

Yang H., Sikdar B., 2003, A Protocol for Tracking Mobile Targets Using Sensor Networks. In Proceedings of the IEEE Workshop on Sensor Network Protocols and Applications, s. 7181.

Yu, Y., Estrin, D., Govindan, R., 2001, Geographical and energy-aware routing: a recursive data dissemination protocol for wireless sensor networks, UCLA Computer Science Department Technical Report, UCLA-CSD.

Yang, W., Fu, Z., Kim, J. and Park M., 2007, An Adaptive Dynamic Cluster-Based Protocol for Target Tracking in Wireless Sensor Networks, WAIM 07, s. 157-167.

Yang, L. Feng, C., Rozenblit, J. W., Qiao, H., 2006, Adaptive Tracking in Distributed Wireless Sensor Networks, Engineering of Computer Based Systems, s. 111.

Yang, N., Tian, W. and Jin, Z., 2006, An interacting multiple model particle filter for manoeuvring target location, Measurement Science and Technology, s.1307-1311.

KAYNAKLAR DİZİNİ (devam)

Younis, O., Fahmy, S., 2004, Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach, INFOCOM'04, s. 629-640.

Yükselen M.A., 2008, Eğri Uydurma ve İnterpolasyon,

<http://www3.itu.edu.tr/~yukselen/HM504/02%20E%F0ri%20uydurma%20ve%20interpolasyon.pdf> (Erişim Tarihi: 10 Ekim 2009)

Zhang P., Sadler, C.M., Lyon, S.A. and Martonosi, M., 2004, Hardware Design Experiences in ZebraNet, SenSys04, ACM, s. 227-238.

Zhang W., Cao, G., 2004, DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks, IEEE Transactions on Wireless Communications, s.1689-1701.

EKLER

EK 1 Simülasyon Ortamı

Tez kapsamında algoritmaların gerçekleştirimleri için, literatürde çok yaygınca kullanılan ns2 (Fall and Varadhan, 2006) simülasyon ortamı tercih edilmiştir. Ns2 simülasyon ortamı, kullanımının bedava olması, Linux ve Windows gibi farklı platformlarda çalıştırılabilmesi, duyarga ağlarının karakteristikleri sunmada birçok esneklik sunması gibi çok sayıda avantajlara sahiptir. Ns2 ortamında, duyarga ağı birçok protokol ve karakteristik kümeleriyle yapılandırılabilir. Ns2'nun kablosuz modeli ayrıca düğüm hareketleri, enerji sınırlamaları da içerir ve senaryolara uygun olarak kodun değiştirilebilmesini mümkün kılar (Halvardson and Lindberg, 2004).

Ns2, ağ ortamının benzetimini gerçekleştirebilmek amacıyla, Tcl/OTcl ve C/C++ dillerinde yazılmış çok sayıda kütüphane içeren bir bilgisayar programıdır. Bu kütüphaneler TCP, yönlendirme, çoğa yayım (multicast), MAC protokolleri ve mimariler gibi çeşitli protokoller içermektedir. Tcl/OTcl ile C/C++ dillerinin birleşiminden oluşan ns2'da, Tcl dili genellikle ağ ortamını modellemek ve komut girmek için kullanılırken, C++ dili, Tcl ile oluşturulan senaryoda girilen komutlar doğrultusunda işlem yapmak için kullanılır.

Node sınıfından türetilmiş olan MobileNode, kanal üzerinden veri iletimi, alımı ve hareketlilik gibi bir çok fonksiyona sahip temel bir Node nesnesidir. İşlemler çoğunlukla bu nesne üzerinden yapılmaktadır.

Ortamdaki duyargaların hareketliliği setdest komutu ile sağlanmaktadır. Setdest komutunun çalışma formatı:

setdest -v <version> -n <number of nodes> -m <minimum speed (m/s)> -M <maximum speed (m/s)> -t <simulation time (s)> -p <pause time> -x <width of surface area> -y <height of surface area> > scenario file

Bu komutta kullanılan parametrelerin anlamları:

- -v: Setdest komutunun versiyonunu belirtir. Tez kapsamında gerçekleştirmiş olduğumuz uygulamalarda 2. versiyonu kullanmış bulunmaktayız.

- -n: Hareketli olmasını istediğimiz düğüm sayısını belirtir.
- -m: Hareket edecek düğümün minimum hızı belirtir. Çalışmamızda kullanmış olduğumuz minimum hız değeri 30 m/s olarak belirlenmiştir.
- -M: Hareket edecek düğümün maksimum hızını belirtir. Çalışmamızda maksimum 100m/s hızında hareket eden hedef takibi gerçekleştirilmiştir.
- -t: Simülasyon süresini belirtir.
- -p: Duraklama süresini belirtir.
- -x: Duyarga ağ ortamının genişliğini belirtir.
- -y: Duyarga ağ ortamının yüksekliğini belirtir.
- > scenario file: Hareket senaryosunu tutan dosyadır.

EK 2 Deneyler Hakkında Detaylı Bilgi

Simülasyon ortamı için ns2 sürüm 2.31 kullanılmıştır. Deneyler, ortama rasgele dağıtılmış ve IEEE 802.11 ve MAC standartları ile yapılandırılmış düğümler ile gerçekleştirilmiştir. Simülasyon süresi 100 saniye olarak belirlenmiştir.

Deneylerde hem doğrusal hem de doğrusal olmayan hareket modelleri üzerinden ölçümler yapılmıştır. Doğrusal hareket modelleri için ns2 simülatörü tarafından desteklenen Random Waypoint hareket modeli kullanılmıştır. Bu hareket modeline göre düğüm bir başlangıç noktasından V_{max} ve V_{min} arasında belirlemiş olduğu sabit bir hızla rasgele bir uç koordinata doğru ilerler. Hız değeri $[0, V_{max}]$ aralığından seçilmektedir. Büyük V_{max} değerleri için düğümler daha hızlı ilerlemektedir. Çünkü V_{max} değeri arttıkça $V_{avg} = (0 + V_{max})/2$ değeri de artmaktadır. Düğüm uç noktaya ulaştığında T_{pause} da belirtilen süre kadar bekler. Bu hareket modelinde düğüm zigzag doğrular çizerek ilerler. Doğrusal olmayan hareket modellerinin üretimi için özel bir yazılım geliştirilmiştir. Geliştirmiş olduğumuz yazılım ile, hedefin sabit hızlı, ivmeli veya dönüşlü hareketleri rasgele belirlenen süre ve değerler ile oluşturulmaktadır.

Deneylerde simülasyon ortamını modellemek için ns2'nun desteklemiş olduğu Tcl dosyası, hedef takip algoritmasında tanımlı işlemlerin gerçekleştirimi için de C++ dosyası kullanılmıştır. Tcl dosyasında, kaç adet düğümün kullanılacağı, ortamın X ve Y koordinatları, düğümler arasındaki iletişim alt yapısı, simülasyon süresi gibi bilgiler tanımlanmaktadır. C++ dosyasında ise, her bir algoritma için tasarlanmış olduğumuz sonlu durum makinesine göre kodlamalar gerçekleştirilmiştir. Sonlu durum makinesinde belirlenen durum ve mesaj tanımlamaları yapılmış, hangi durumda hangi mesajın gelmesi halinde gerçekleştirilecek işlemler kodlanmıştır.

Her bir ölçüm sonucu, deneylerin ortalaması alınarak belirlenmiştir. Doğrusal hareket eden deneylerde her bir hız aralığı için setdest komutu ile 70'er adet rasgele farklı rotalarda hareket modelleri oluşturulup, elde edilen ortalama hedef kayıp oranları hesaplanmıştır. Doğrusal olmayan hareket modellerindeki Parçacık ve Genişletilmiş Kalman Filtrelerinin tahminlemedeki hata oranı karşılaştırılması işleminde 30 tane rasgele doğrusal olmayan hareket modeli kullanılmış ve ortalamaları alınmıştır.

Algoritmaların gerçekleştirimindeki işlem sırasını özetleyecek olursak, öncelikle önermiş olduğumuz algoritmaların sonlu durum makineleri tasarlanmıştır. Bunu takiben farklı hızlarda rasgele doğrusal ve doğrusal olmayan hareket modelleri

oluřturulmuřtur. Daha sonra simlasyon ortamını modelleyen Tcl dosyası oluřturulmuřtur. Son olarak, sonlu durum makinesinde belirtilen durum, mesaj ve eylemler C++ ile kodlanarak lm sonuları elde edilmiřtir.

ÖZGEÇMİŞ

Kişisel Bilgiler

Doğum Tarihi: 3 Kasım 1983

Doğum Yeri: Manisa- Alaşehir

Eğitim Bilgileri

Doktora:

Ege Üniversitesi Bilgisayar Mühendisliği
Celal Bayar Üniversitesi adına OYP programı kapsamında.

Tez konusu: Telsiz Duyarga Ağlarında Hedef Takibi

Tez danışmanları: Doç. Dr. Aylin Kantarcı ve Prof. Dr. Kayhan Erciyes

Lisans:

Sakarya Üniversitesi, Bilgisayar Mühendisliği

Genel Not Ortalaması: 3.86 / 4.00

Mühendislik Fakültesi Birincisi

Lise:

Manisa Turgutlu Halil Kale Fen Lisesi

Genel Not Ortalaması: 4.82 / 5.00

Ortaokul:

Manisa Alaşehir Şehitleri Anadolu Lisesi

Genel Not Ortalaması: 93 / 100

İlgi Alanları:

- Dağıtık sistem tasarımı, entegrasyonu ve analizi
- Telsiz Duyarga Ağları
- Telsiz Duyarga Ağlarında Hedef Takibi
- Telsiz Haberleşme ve Ağ oluşturma

Yayınlar:

Uluslararası Hakemli Dergiler (SCI-Expanded)

- Aysegül Alaybeyoğlu, Kayhan Erciyes, Orhan Dagdeviren, Aylin Kantarcı, "Tracking Fast Moving Targets in Wireless Sensor Networks", IETE Technical Review, 2009 (Kabul edildi)

- Aysegul Alaybeyoglu, Aylin Kantarci, Kayhan Erciyes, Orhan Dagdeviren, "Cluster Based Target Tracking in Wireless Sensor Networks", The Computer Journal, 2009 (Gönderildi)
- “Tracking Algorithms for Linearly and NonLinearly Fast Moving Targets in Wireless Sensor Networks”, Journal of Network and Computer Application, Elsevier (Gönderilecek)

Uluslar arası Bildiri

- Aysegul Alaybeyoglu, Orhan Dagdeviren, Kayhan Erciyes, Aylin Kantarci, "Performance Evaluation of Cluster-based Target Tracking Protocols for Wireless Sensor Networks", ISCIS 2009, IEEE
- Aysegul Alaybeyoglu, Aylin Kantarci, Kayhan Erciyes, “A Dynamic Distributed Tree Based Tracking Algorithm for Wireless Sensor Networks”, CONECO 2010, LNCS (Gönderildi)

Ulusal Bildiri

- Ayşegül Alaybeyoğlu, Orhan Dağdeviren, Aylin Kantarcı, Kayhan Erciyes, "Telsiz Duyarga Ağlarında Hızlı Hareket Eden Hedefler için Küme Tabanlı Hedef İzleme Algoritması", IEEE SIU 2009, Antalya
- Ayşegül Alaybeyoğlu, Aylin Kantarcı, Kayhan Erciyes, "Telsiz Duyarga Ağlarında Hedef İzleme Senaryoları", Akademik Bilişim 2009, Urfa
- Ayşegül Alaybeyoğlu, Aylin Kantarcı, Kayhan Erciyes, “Çizge Teorisi, Dağıtık Algoritmalar ve Telsiz Duyarga Ağları”, Akademik Bilişim 2010, Muğla (Gönderildi)

