



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



**OTONOM ROBOTLARDA, DURUMSAL FARKINDALIK TEMELLİ
DİNAMİK KARAR VERME MODELİ VE YENİ BİR VERİ SETİ İLE
ADAPTİF SLAM UYGULAMASI**

Mümine YILDIZ

DOKTORA TEZİ

Mekatronik Mühendisliği Anabilim Dalı

AĞUSTOS - 2024

KONYA

Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Mümine YILDIZ tarafından hazırlanan ”*OTONOM ROBOTLARDA, DURUMSAL FARKINDALIK TEMELLİ DİNAMİK KARAR VERME MODELİ VE YENİ BİR VERİ SETİ İLE ADAPTİF SLAM UYGULAMASI*” adlı tez çalışması 30/07/2024 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Doç. Dr. Sinan GÜLTEKİN

Danışman

Prof. Dr. Mehmet KARALI

Üye

Doç. Dr. Barış GÖKÇE

Üye

Doç. Dr. Mustafa KAYA

Üye

Doç. Dr. Abdulsamed TABAK

Yukarıdaki sonucu onaylarım.

Prof. Dr. Havvanur UÇBEYİAY

FBE Müdürü

Bu tez çalışması NEÜ BAP birimi tarafından 191419006 nolu proje ile desteklenmiştir.

Dr. Öğr. Üyesi Özgür DÜNDAR bu tez çalışmasının ikinci danışmanıdır.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Mümine YILDIZ

Tarih: 29/08/2024

ÖZET

DOKTORA TEZİ

OTONOM ROBOTLARDA, DURUMSAL FARKINDALIK TEMELLİ DİNAMİK KARAR VERME MODELİ VE YENİ BİR VERİ SETİ İLE ADAPTİF SLAM UYGULAMASI

Mümine YILDIZ

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü

Mekatronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Mehmet KARALI

2024, 133 Sayfa

Jüri

Prof. Dr. Mehmet KARALI

Doç. Dr. Sinan GÜLTEKİN

Doç. Dr. Barış GÖKÇE

Doç. Dr. Mustafa KAYA

Doç. Dr. Abdulsamed TABAK

Otonom robotlar için çevredeki engelleri algılama ve doğru konumlandırma, çözülmesi gereken temel sorunlar arasında yer almaktadır. GPS olmayan ortamlarda, SLAM (Eş Zamanlı Konumlama ve Haritalama) algoritmaları, bu soruna en etkili çözümlerden biri olarak öne çıkmaktadır. SLAM algoritmaları, gerçek dünya koşullarında önemli başarılar elde etmiş olsa da, çevresel zorluklar ve değişken koşullar, bu algoritmaların etkinliğini sınırlayabilmektedir. Bu nedenle, daha kapsamlı ve etkili bir SLAM teknolojisinin geliştirilmesi gerekmektedir.

Bu çalışma, SLAM algoritmalarına daha bütüncül bir yaklaşım getiren ve insanların durumsal farkındalık ve dinamik karar verme süreçlerinden esinlenerek tasarlanan "Adaptif SLAM Karar Modeli"ni sunmaktadır. Bu model, robotların çevresel engelleri algılama, doğru

konumlandırma ve deęişen kořullara adaptif tepkiler verme yeteneklerini geliřtirmektedir. Sistem, üretilen tepkilerin sonuçlarını deęerlendirerek kendini geliřtiren bir öęrenme sürecine sahiptir. Böylece, SLAM algoritmalarının zorlu çevre kořullarında daha saęlam ve güvenilir performans sergilemesi saęlanmaktadır.

Modelin etkinlięini test etmek amacıyla Konya-Dataset ¹ oluřturulmuřtur. Bu veri seti, 2D Lidar, RGB-D ve Stereo kamera, IMU ve Leddar sensörleri gibi sensörlerle toplanan verilerden oluřmaktadır. Ayrıca, durumsal farkındalıęı artırmak amacıyla bir ıřık sensörü de sisteme dahil edilmiřtir. Veri seti, yansıtıcı yüzeyler, farklı aydınlatma ortamları, çok katlı yapılar, özniteliksiz sahneler ve hareket bozulmaları gibi zorluklarla dolu çeřitli kapalı alanlarda toplanmıřtır. Bu sayede, SLAM algoritmalarının sınırları zorlanmıř ve algoritmaların hataya düřtüęü durumlar tespit edilmiřtir. Tespitlerimiz SLAM topluluęunun odaklanması gereken noktalar olarak ayrıca bir katkı saęlamıřtır.

Adaptif SLAM Karar Modeli, ORB-SLAM3 algoritması ile yapılan testlerde tespit edilen bozulmaları gidermek amacıyla uygulanmıřtır. Durum farkındalıęının kazandırıldıęı ve sensör füzyonunun adaptif bir karar süreci ile yönetilerek elde edilen yörüngeler, Geniřletilmiř Kalman Filtresi ve ORB-SLAM3 algoritmalarıyla elde edilen yörüngelerle karřılařtırılmıřtır. Sonuçlar, Adaptif SLAM Karar Modeli'nin, zorlu çevre kořulları karřısında daha saęlam ve güvenilir bir performans sergiledięini göstermektedir.

Bu model, SLAM algoritmalarını sadece bir haritalama aracı olarak deęil, çevresel zorlukları ve görevle ilgili anlamlandırmayı ięeren çok katmanlı bir sistem olarak ele almaktadır. Modelin yapay zeka ve derin öęrenme gibi yöntemlerle daha da geliřtirilmesi, gelecekte daha yüksek performanslı ve güvenilir SLAM çözümlerine ulařılmasını saęlayacaktır. Bu çalıřma, adaptif SLAM teknolojisinin geliřtirilmesinde saęlam bir temel sunmakta ve otonom sistemlerin performansını artırmak için önemli bir katkı saęlamaktadır.

Anahtar Kelimeler: Adaptif SLAM, Durumsal Farkındalık, Sensör Füzyonu, SLAM Veri Seti

¹Konya Dataset řu adreste mevcuttur: <https://github.com/Konya-Dataset/Konya-Dateset>

ABSTRACT

Ph.D THESIS

ADAPTIVE SLAM APPLICATION IN AUTONOMOUS ROBOTS WITH A SITUATIONAL AWARENESS BASED DYNAMIC DECISION MAKING MODEL AND A NEW DATASET

Mümine YILDIZ

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE / DOCTOR OF PHILOSOPHY
IN MATHEMATICS**

Advisor: Prof. Dr. Mehmet KARALI

2024, 133 Pages

Jury

Prof. Dr. Mehmet KARALI

Doç. Dr. Sinan GÜLTEKİN

Doç. Dr. Barış GÖKÇE

Doç. Dr. Mustafa KAYA

Doç. Dr. Abdulsamed TABAK

One of the fundamental challenges in autonomous robotics is detecting environmental obstacles and achieving accurate localization. In the absence of GPS, SLAM (Simultaneous Localization and Mapping) algorithms are among the most effective solutions to this problem. Although SLAM algorithms have achieved significant success in real-world conditions, environmental challenges and varying conditions can limit their effectiveness. Therefore, there is a need to develop a more comprehensive and effective SLAM technology.

This study presents the "Adaptive SLAM Decision Model," which introduces a more holistic approach to SLAM algorithms, inspired by human situational awareness and dynamic

decision-making processes. This model enhances the robot's ability to detect environmental obstacles, achieve accurate localization, and adaptively respond to changing conditions. The system undergoes a learning process that evaluates the outcomes of the generated responses, allowing it to improve over time. Consequently, the SLAM algorithms are enabled to perform more robustly and reliably under challenging environmental conditions.

To test the effectiveness of the model, the Konya-Dataset² was created. This dataset comprises data collected using sensors such as 2D Lidar, RGB-D and Stereo cameras, IMU, and Leddar sensors. Additionally, a light sensor was incorporated into the system to enhance situational awareness. The dataset was collected in various indoor areas with challenges such as reflective surfaces, different lighting environments, multi-story structures, featureless scenes and motion distortions. Through this dataset, the limits of SLAM algorithms were tested, and the situations where the algorithms failed were identified. These findings also contribute to highlighting the areas where the SLAM community should focus its efforts.

The Adaptive SLAM Decision Model was applied to address the errors and failures identified during tests conducted with the ORB-SLAM3 algorithm. The trajectories obtained by providing situational awareness and managing sensor fusion with an adaptive decision process were compared with the trajectories obtained with the Extended Kalman Filter and ORB-SLAM3 algorithms. The results demonstrate that the Adaptive SLAM Decision Model exhibits more robust and reliable performance in the face of challenging environmental conditions.

This model treats SLAM algorithms not merely as a mapping tool but as a multi-layered system that incorporates environmental challenges and task-related comprehension. Further development of the model through methods such as artificial intelligence and deep learning is expected to lead to more high-performance and reliable SLAM solutions in the future. This study provides a solid foundation for the advancement of adaptive SLAM technology and makes a significant contribution to enhancing the performance of autonomous systems. **Keyword:** Adaptive SLAM, Sensor Fusion, Situational Awareness, SLAM Dataset

²Konya Dataset is available at: <https://github.com/Konya-Dataset/Konya-Dateset>

ÖNSÖZ

Gerek akademik gerek manevi yönden desteklerini esirgemeyen kıymetli danışmanım Prof. Dr. Mehmet Karalı'ya şükranlarımı sunarım. Bu süreç boyunca tecrübelerini benimle paylaşan ve sabırla dinleyen hocalarım Dr. Öğr. Üyesi Özgür Dündar, Doç. Dr. Barış Gökçe, Doç. Dr. Sinan Gültekin ve Doç. Dr. Üyesi Emrehan Yavşan'a teşekkür ederim. Ayrıca, birçok problemi birlikte çözdüğümüz Furkan Ateş'e, ortak projeler yürüttüğüm ve hayaller paylaştığım Yasin Elhan ve İbrahim Aydın'a içten teşekkürlerimi iletirim. Çalışmalarımın son döneminde yanımda olan Hasan Karayaka, Arzu Coşgun ve Mehmet Fatih Koç'a da teşekkür ederim.

Bu süreçte yanımda olan ailemin her bir ferdine, özellikle varlığıyla destek olan annem Neslihan'a minnettarım. Kızlarım Halime Dua ve Elif Sueda'ya, bu çalışmayı ithaf ederim.

Mümine YILDIZ

KONYA-2024

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	vi
ÖNSÖZ	viii
ŞEKİLLER LİSTESİ	xii
ÇİZELGELER LİSTESİ	xiii
SİMGELER VE KISALTMALAR	xiv
1. GİRİŞ	1
1.1. Araştırma Problemleri	2
1.1.1. Dinamik Ortamlarda SLAM Güvenilirliği ve Sağlamlığı	2
1.1.2. Durumsal Farkındalık Kavramının SLAM Teknolojilerinde Kullanımı	3
1.1.3. SLAM Veri Setlerinin Güncelliği ve Çeşitliliği	4
1.2. Tezin Katkıları	5
1.3. Tez Genel Planı	6
2. KAYNAK ARAŞTIRMASI	8
2.1. Otonom Robotlarda Konumlama ve Haritalama	9
2.2. Bilişsel Robotik ve Durumsal Farkındalık	20
2.3. SLAM Veri setleri	28
3. EŞ ZAMANLI HARİTALAMA VE KONUMLAMA (SLAM)	36
3.1. SLAM Algoritmaları Teori ve Yaklaşımları	36
3.2. Robot İşletim Sistemi (ROS) ve SLAM	40
4. SLAM ALGORİTMALARI İÇİN DURUMSAL FARKINDALIK TEMELLİ DİNAMİK KARAR VERME MODELİ	46
4.1. Karar Verme Modelleri	46
4.1.1. Dinamik Karar Verme Modeli (DDM)	47
4.1.1.1. Durumsal Farkındalık	48
4.1.1.2. Karar Verme	51
4.1.1.3. Bellek Yapıları	52
4.1.2. Örneğe Dayalı Öğrenme Teorisi (IBLT)	53

4.2. Adaptif SLAM Karar Modeli	55
4.2.1. Çalışma Belleği	55
4.2.1.1. Durumsal Farkındalık	56
4.2.1.2. Analiz	59
4.2.1.3. Karar Verme	60
4.2.1.4. Eylem	60
4.2.2. Uzun Süreli Bellek.....	60
4.2.2.1. Deneyim	61
4.2.2.2. Kurallar	63
4.2.2.3. Stratejiler.....	63
4.2.2.4. Reçeteler	64
4.3. Tartışma	65
5. KONYA VERİ SETİ	67
5.1. Sensör Platformu ve Sensörler	67
5.2. Sensor Zaman Senkronizasyon	71
5.3. Veri Dizileri	71
5.4. Veri Seti ile SLAM Algoritma Testleri	75
5.5. Tartışma	78
6. ADAPTİF SLAM UYGULAMASI	83
6.1. Durumsal Farkındalık	83
6.1.1. Algılama.....	83
6.1.2. Kavrama	83
6.1.3. Kestirim	86
6.2. Reçeteler	86
6.2.1. Kalman Filtresi.....	87
6.2.2. Genişletilmiş Kalman Filtresi	96
6.2.3. Sistem modeli	97
6.2.4. Reçeteler	103
6.3. Analiz	104
6.4. Karar Verme	105
6.5. Test Sonuçları	106
7. SONUÇ VE ÖNERİLER	117
7.1. Sonuçlar	117

7.2. Öneriler	117
KAYNAKLAR	119

ŞEKİLLER LİSTESİ

<u>Şekil</u>	<u>Sayfa</u>
2.1 Dinamik Karar Verme Modeli (Endsley, 1995)	22
2.2 IBLT modeli (Gonzalez vd., 2003)	23
2.3 Robot Durum Farkındalığı Modeli (Ben Ghezala vd., 2014)	25
2.4 Bilişsel ve İşbirlikçi Süreç (Haag ve Huschbeck)	26
2.5 S-Graphs (Bavle vd., 2022)	27
2.6 S-Graphs+ (Bavle vd., 2023b)	28
4.1 Dinamik Karar Verme Modeli (Endsley, 1995)	47
4.2 IBLT modeli (Gonzalez vd., 2003)	53
4.3 Adaptif SLAM Karar Modeli	55
5.1 Tekerlekli Sensör Platformu	68
5.2 Elde Taşınan Sensör Platformu	69
5.3 Zaman içerisinde gerçekleşen değişimlere ait görseller	73
5.4 Aynaların bulunduğu sahne	74
5.5 Aynaların bulunduğu sahneden elde edilen harita	75
5.6 Parlama ve özniteliksiz sahne nedeniyle yörünge kopması	76
5.7 Aşırı karanlık sahne nedeniyle yörünge kopması	76
5.8 Akademik building2.1 ORB-SLAM3 ve Cartographer yörünge çıktıları .	77
5.9 Cartographer Akademik Building2.1 haritasında kayma hatası	78
6.1 Sadeleştirilmiş Adaptif SLAM Karar Modeli	83
6.2 Robot modeli ve sensör bağlantıları	84
6.3 Kalman Algoritması	88
6.4 Genişletilmiş Kalman Algoritması	97
6.5 Home1.0 Yörünge Çizgisi	109
6.6 Home1.0 Yörünge ve Kopma Noktaları	110
6.7 ORB-SLAM3 yörünge APE hata sonuçları	111
6.8 ORB-SLAM3 yörünge RPE hata sonuçları	112
6.9 EKF yörünge APE hata sonuçları	113
6.10 EKF yörünge RPE hata sonuçları	114
6.11 Model ile elde edilen yörünge APE hata sonuçları	115
6.12 Model ile elde edilen yörünge RPE hata sonuçları	116

ÇİZELGELER LİSTESİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1 Popüler Veri Setleri Özellikleri	33
2.2 Popüler Veri Setleri Özellikleri	34
2.3 Kapalı Alan Veri Setlerinin Konya-Veri Seti ile Karşılaştırma Tablosu ..	35
5.1 Sensör Tipleri ve Özellikleri	70
5.2 Home1 veri seti dizi özellikleri	80
5.3 Home2 veri seti dizi özellikleri	80
5.4 Home2 ayna özel veri seti dizi özellikleri	81
5.5 Academic_building1 veri seti dizi özellikleri	81
5.6 Academic_building1 çok katlı veri seti dizi özellikleri	81
5.7 Academic_building2 veri seti dizi özellikleri	82
5.8 Farklı alanlarda toplanan veri seti dizi özellikleri	82

SİMGELER VE KISALTMALAR

Simgeler

- 3D: Üç Boyutlu
2D: İki Boyutlu
 z_k : Sensör Ölçümleri
 \hat{x}_k : Tahmin
 \hat{x}_{k+1}^- : Durum Kestirimi
 K_k : Kalman Kazancı
 R : Ölçüm gürültüsü matrisi
 Q : Süreç gürültüsü matrisi
 D : Desen matrisi
 K : Kestirim dizisi

Kısaltmalar

- | | |
|---------------------------------------|-------------------------------------|
| İHA | :İnsansız Hava Aracı |
| SLAM | :Eş Zamanlı Konumlama ve Haritalama |
| SA (Situation Awareness) | :Durumsal Farkındalık |
| DDM (Dynamic Decision Making) | :Dinamik Karar Verme |
| IBLT (Instance based learning theory) | :Örneğe dayalı öğrenme teorisi |
| MDE (Model Driven Engineering) | :Model Tabanlı Mühendislik |
| Lidar (Light Detection and Ranging) | :Işın Hızıyla Uzaklık Ölçümü |
| RGB-D | :Renk ve Derinlik |
| GPS | :Küresel Konumlandırma Sistemi |
| IMU | :İnertiyal Ölçüm Ünitesi |
| RANSAC | :Rastgele Örnek Konsensüsü |
| KF(Kalman Filter) | :Kalman Filtresi |
| EKF(Extended Kalman Filter) | :Genişletilmiş Kalman Filtresi |
| UKF (Unscented Kalman Filter) | :Kokusuz Kalman Filtresi |
| IF (Information Filter) | :Bilgi Filtresi |
| PF Particle Filter) | :Parçacık Filtresi |
| CNN (Convolutional Neural Network) | :Evrimsel Sinir Ağı |
| ROS(Robot Operating System) | :Robot İşletim Sistemi |

ORB-SLAM (Omnidirectional RGB-D SLAM)

: Omnidireksiyonel RGB-D SLAM

RTAB (Real-Time Appearance-Based)

:Gerçek Zamanlı Görünüm Tabanlı

CPU

:Merkezi İşlem Birimi

6-DoF

:Altı Serbestlik Derecesi

FOV(Field of View)

:Görüş Alanı

VINS (Visual-Inertial Navigation System)

: Görsel Ataletsel Navigasyon Sistemi

1. GİRİŞ

Otonom robotlar, insan müdahalesi olmaksızın çevrelerinde bağımsız olarak hareket edebilen ve belirli görevleri yerine getirebilen cihazlar olarak tanımlanır. Bu robotlar, sensörler, yapay zeka algoritmaları ve gelişmiş kontrol mekanizmaları aracılığıyla çevrelerini algılar, yorumlar ve bu bilgiler doğrultusunda kararlar alırlar. Bu süreç, robotların algılama, konum belirleme, biliş ve hareket kontrolü gibi birçok unsura bağlı olması gerektiği anlamına gelir. Otonom robotların temel hedeflerinden biri, sensör verilerini dikkate alarak, bilinen veya bilinmeyen bir ortamda bir noktadan diğerine güvenli bir şekilde hareket edebilmektir.

Otonom robotların temel unsurları; haritalama, konumlandırma ve yol planlamayı içerir. Haritalama, robotların belirli bir ortamda doğru ve güvenilir bir şekilde konumlandırma yapabilmesi için çevrenin haritasını oluşturma sürecidir. Bu süreç, robotların ortamda meydana gelen değişiklikleri fark etmelerini ve buna uygun hareket etmelerini sağlar. Konumlandırma, robotun bulunduğu ortam içindeki tam yerini belirleme sürecidir (Panzieri vd., 2002). Robotların görevlerini yerine getirebilmeleri için başlangıçta buldukları konumu doğru bir şekilde bilmeleri hayati önem taşır. Yol planlama ise, otonom bir robotun belirlenmiş bir harita üzerinde en uygun rotayı hesaplaması sürecidir (Nilwong vd., 2019).

Çoğu durumda, otonom robotlar başlangıç noktasından hedefine doğru düz bir rota izleyemezler. Bu nedenle, robotun hareketini sağlamak için doğru ve yeterli konum bilgisi kritik öneme sahiptir. Konum belirleme yöntemleri, bu bağlamda navigasyon sürecinin temel taşları olarak kabul edilir. Ayrıca, hareket eden bir robotun başarılı bir şekilde navigasyon yapabilmesi; kontrol yetenekleri, yol planlaması, engellerden kaçınma ve hedefe güvenli bir mesafede ulaşma gibi ek beceriler gerektirir (Zghair ve Al-Araji, 2021). Bu bağlamda, Eş Zamanlı Yerelleştirme ve Haritalama (SLAM) en yaygın kullanılan yöntemlerden biridir. SLAM, bir robotun bilinmeyen bir ortamda hem kendi konumunu belirlemesini hem de bu ortamın haritasını oluşturmasını sağlar. SLAM algoritmaları, robotun sensörlerinden (Lidar, kamera, GPS, IMU) gelen verileri kullanarak çevrenin bir haritasını çıkarır ve aynı zamanda robotun bu harita içindeki konumunu sürekli

olarak günceller. Bu döngüsel süreç, robotun hareket ettikçe hem harita hem de konum bilgisinin sürekli olarak iyileştirilmesini sağlar.

Otonom robotların gelişimi, sensör teknolojileri, yapay zeka ve kontrol algoritmalarındaki ilerlemelerle hızla devam etmektedir. Ancak, önemli gelişmelere rağmen, otonom robotların yapılandırılmamış ortamlarda ve uzun süreli kullanımlarında konum kaybı ve harita bozulması gibi çözülmesi gereken önemli sorunlar bulunmaktadır. Bu tez çalışmasının ana odağı olan SLAM algoritmalarının yapılandırılmamış ortamlardaki performansını artırmaya yönelik temel yaklaşımlar, "Araştırma Problemleri" başlığı altında detaylı bir şekilde açıklanacaktır.

1.1. Araştırma Problemleri

Bu bölümde, SLAM algoritmalarına yönelik temel araştırma problemleri tartışılmıştır.

1.1.1. Dinamik Ortamlarda SLAM Güvenilirliği ve Sağlamlığı

SLAM teknolojisindeki ilerlemeler, SLAM algoritmalarının gerçek dünya koşullarında uygulanabilirliğini artırmış, böylece bu algoritmalar yapılandırılmış ortamlar yerine daha dinamik ve değişken şartlarda kullanılmaya başlanmıştır. Ancak bu geçiş, doğruluk ve sağlamlık açısından yeni zorlukları da beraberinde getirmiştir. Robotun hareketi veya çevresindeki değişiklikler, SLAM algoritmalarında harita oluşturamama, harita bozulmaları ve konum kaybı gibi başarısızlıklara neden olabilmektedir (Cadena vd., 2016). Gerçek dünya koşullarında karşılaşılabilecek zorluklar arasında dinamik engeller, değişen aydınlatma koşulları, zamanla değişen nesnelere, çok katlı yapılar, asansörler, merdivenler ve farklı yüzey özellikleri (örneğin aynalar) bulunmaktadır. Bu zorluklar karşısında SLAM algoritmalarında ortaya çıkan hatalar, otonom robotların görevlerini yerine getirmesini doğrudan etkileyebilir.

İdeal bir SLAM çözümü, başarısızlığa karşı korumalı olmalı ve yaklaşmakta olan başarısızlıkları önceden tespit edebilmelidir. Bu, sistemin uygun çalışmayı sürdürebilmesi için kurtarma mekanizmalarını devreye sokabilmesini gerektirir. SLAM algoritmalarının sağlamlığını ve güvenilirliğini artırmak amacıyla sensör

füzyonu, öğrenme algoritmaları ve optimizasyon tekniklerine dayalı çeşitli yaklaşımlar geliştirilmiştir. Ancak, gerçek dünya senaryolarında tam anlamıyla sağlam ve güvenilir bir çözüme henüz ulaşılamamıştır.

1.1.2. Durumsal Farkındalık Kavramının SLAM Teknolojilerinde Kullanımı

Durumsal farkındalık, psikoloji, askeriye, havacılık ve eğitim gibi birçok alanda derinlemesine incelenmiş temel bir insan yeteneğidir. Bireylerin veya sistemlerin çevrelerindeki olayları ve koşulları anlama, doğru bir şekilde yorumlama ve gelecekteki olası gelişmeleri öngörebilme yeteneği olarak tanımlanır. Endsley (1995) tarafından geliştirilen Durumsal Farkındalık Temelli Dinamik Karar Verme Modeli, robot-işbirlikli projelerde operatörlerin farkındalığını artırmak amacıyla sıkça kullanılmıştır. Son yıllarda, otonomi çalışmalarındaki ilerlemelerle birlikte, durumsal farkındalık ve dinamik karar verme görevleri robotlara aktarılmaya çalışılmaktadır.

Mobil robotların karmaşık görevleri verimli ve güvenli bir şekilde gerçekleştirme yetenekleri, çevreleri hakkındaki bilgi seviyeleri, yani durumsal farkındalıkları ile sınırlıdır. Bu nedenle, robotların karar verme sistemleri, çevrelerinin sürekli olarak yeni gözlemlerini edinmeli, temel unsurlarını anlamalı, karmaşık akıl yürütme yapmalı ve bu bilgiler doğrultusunda eylemler gerçekleştirmelidir. Gelişmiş akıl yürütme, karar verme ve icra becerileri, robotların bilinmeyen ortamlarda otonom olarak hareket etmesini sağlar. Durumsal farkındalık, robotların karar verme süreçlerine katkıda bulunur; örneğin, bir robot çevresindeki bilgileri kullanarak engellerden kaçınma, en verimli yolu bulma veya belirli görevleri yerine getirme konusunda kararlar alabilir. Durumsal farkındalık ayrıca robotların geçmiş deneyimlerinden öğrenmelerine ve çevrelerine daha iyi adapte olmalarına olanak tanır; bu da özellikle uzun süreli görevlerde veya sürekli değişen koşullarda önemli bir avantaj sağlar.

Ancak durumsal farkındalık, robotik alanda henüz tam anlamıyla ele alınmamıştır. Bu alanda daha çok algılama, mekansal algı, sensör füzyonu, durum tahmini ve SLAM gibi tekil konulara odaklanılmıştır. SLAM algoritmaları genellikle çevrenin anlamlandırılması aşamasında bir çözüm olarak kullanılsa da, bu yaklaşım durumsal farkındalık modelinin yalnızca bir kısmını kapsamaktadır.

SLAM teknolojilerinin nihai hedefi, her türlü durumda sağlam ve güvenilir bir sistem geliştirmek için daha bütüncül bir yaklaşımdır. Dinamik Karar Verme (DDM) modelinin SLAM algoritmalarına uyarlanması, SLAM'ın problemlere karşı dayanıklılığını artırabileceğini ve öğrenme ile kendini geliştirmesine katkıda bulunacağını düşünmekteyiz.

1.1.3. SLAM Veri Setlerinin Güncelliği ve Çeşitliliği

SLAM algoritmaları, kullanılan sensör teknolojilerine ve yaklaşımlara bağlı olarak gelişim göstermektedir. SLAM algoritmalarının geliştirilmesi ve test edilmesi için kullanılan veri setlerinin, güncel çalışmaları destekleyebilecek, teknolojik donanım ve senaryo beklentilerini karşılayabilecek özelliklere sahip olması gerekmektedir. Ayrıca, SLAM algoritmalarının gerçek dünya uygulamalarında kullanımını desteklemek amacıyla, farklı ortam ve senaryolar içerisinde karşılaşılabilecek zorlukları içermesi gerekmektedir (Zhang vd., 2022). Bu zorluklar, algoritmaların dayanıklılığını artırmak için önemlidir. Kapalı ortamlar ve açık alanlarda karşılaşılabilecek zorluklar farklı karakterlere sahiptir. Örneğin, açık alanlarda mevsimsel değişiklikler ön plana çıkarken, kapalı alanlarda sahne ve aydınlatma özellikleri daha belirleyicidir.

Açık alan ve kapalı alan veri setleri incelendiğinde, otonom sürüşe yönelik veri setleri çeşitlilik açısından yeterli olabilir, ancak kapalı alanlarda çalışacak mobil robotlar için yeterli çeşitlilikte veri seti bulunmamaktadır. Kapalı alan veri setleri arasında, 2D Lidar-görsel füzyon çalışmalarını destekleyen ve aynı zamanda güncel zorlukları içeren veri setleri oldukça sınırlıdır.

Son çalışmalarda, algoritmaların başarısızlık modları üzerine yapılan detaylı analizler büyük talep görmektedir. Başarısızlıklara neden olan zorlukların tespit edilmesi, etkilerinin nicel olarak değerlendirilmesi ve bu tür zorlukları içeren daha çeşitli veri setlerine olan ihtiyacı göstermektedir.

1.2. Tezin Katkıları

Bu tez çalışması, otonom robotların performansını artırmayı hedefleyen SLAM algoritmalarının, gerçek dünya koşullarında uzun süreli kullanım performansını değerlendirmeyi amaçlamaktadır. Bu değerlendirme sırasında, SLAM algoritmalarının zorlandığı durumlar belirlenmiş ve bu zorlukları içeren, kapalı ortamlar için uygun bir veri seti oluşturulmuştur. Bu veri seti, yapılandırılmamış ortamlarda gerçek sensör verileriyle toplanmış olup, sensör füzyonuna elverişli bir şekilde tasarlanmıştır.

Tez kapsamında, Lidar tabanlı Cartographer SLAM algoritması ve Görsel-SLAM tabanlı ORB-SLAM3 algoritmaları bu veri seti ile test edilmiştir. Testler, algoritmaların çeşitli zorluklar karşısında başarısız olabildiğini ortaya koymuştur. Örneğin, her iki algoritma da çok katlı haritalama görevlerinde başarısız olmuş; sarsıntı, özniteliksiz ortamlar, karanlık veya aşırı aydınlatmalı ortamlar gibi durumlarda hatalar gözlemlenmiştir. Ayrıca, yansıtıcı yüzeyler karşısında Cartographer algoritması, aynadaki yansımayı haritaya yanlış yansıtırken, ORB-SLAM3 algoritması yanlış bir yörünge çıkarmıştır. Bu tür hatalar, gerçek dünya senaryolarında sıkça karşılaşılabilecek zorlukları temsil eder ve otonom robotların güvenilirliği açısından kritik öneme sahiptir. Bu bağlamda, zorlukların tespiti ve araştırmacılara sunulan bu veri seti, topluluk için önemli bir katkı sağlamaktadır.

SLAM algoritmalarının gerçek dünya koşullarında uzun süreli kullanımı için sensör füzyonu, öğrenme algoritmaları ve optimizasyon teknikleri gibi disiplinler arası birçok çalışma bulunmaktadır. Ancak, bu çalışmalar genellikle belirli bir problemi hedef alır ve tüm şartlara uyum sağlama yeteneğinden yoksundur. Bu tezde, SLAM algoritmalarının zorluklarla başa çıkabilmesi için durumsal farkındalık temelli bir dinamik karar verme modeli geliştirilmiştir. Model, SLAM algoritmalarını çevredeki engellerin algılanması noktasından, kendi içinde karar verebilen ve öğrenebilen bir seviyeye taşımaktadır.

Geliştirilen bu model, SLAM algoritmalarına durumsal farkındalık kazandırarak, robotların sadece engelleri değil, aynı zamanda zorlukları ve çevredeki önemli bileşenleri algılayıp anlamlandırmalarını sağlamaktadır. Bu da robotların görev doğrultusunda adaptif kararlar alarak zaman içinde öğrenme

yeteneğine kavuşmalarını sağlar. Durumsal farkındalığın SLAM teknolojilerine adaptasyonu ile, robotlar görevlerini daha etkili, güvenli ve bağımsız bir şekilde yerine getirebilecek, çevrelerindeki nesnelere, engelleri ve olumsuz şartları doğru bir şekilde tanımlayabileceklerdir. Bu, karmaşık bir çevreyi insanlar gibi anlama ve algılama yeteneğine sahip bir sistemin oluşmasına katkıda bulunur.

Önerilen modelin etkinliği, oluşturulan veri setiyle yapılan testler ve model çerçevesinde geliştirilmiş karanlık algılama algoritmaları ile test edilmiştir. Elde edilen sonuçlar, modelin hafıza yapılandırmasının farklı öğrenme teknikleri ile desteklenerek daha sağlam sonuçlar verebileceğini göstermektedir. Topluluğun, bu model çerçevesinde yapacağı çalışmalarla, insan algılama, anlamlandırma ve çıkarım yapma yeteneklerini taklit eden, adaptif kararlar alabilen bir sisteme erişilebileceği öngörülmektedir. Ayrıca, farklı disiplinlerdeki çalışmaların bu çerçevede ilerlemesi, araştırmalara hız kazandıracaktır.

1.3. Tez Genel Planı

Tezinizin genel planı şu şekilde düzenlenmiştir:

Giriş Bölümü: Tezin temel problemleri ve ana başlıklar altında bu problemlerin kısa açıklamaları yapılmış, tezin genel çerçevesi belirlenmiştir. Ayrıca, tezin temel katkıları özetlenmiştir.

Kaynak Araştırmaları: Her bir problem için detaylı bir literatür değerlendirmesi yapılmış, ilgili alanlardaki mevcut çalışmalar ve yaklaşımlar ele alınmıştır.

SLAM Algoritmaları: Çalışmanızın temelini oluşturan SLAM algoritmalarına dair kapsamlı bilgiler verilmiştir. Bu bölümde, SLAM algoritmalarının teorik ve uygulama boyutları ele alınmıştır.

Dinamik Karar Verme Modeli ve Durumsal Farkındalık: SLAM algoritmalarının etkinliğini artıracak Dinamik Karar Verme Modeli ve Durumsal Farkındalık konularındaki çalışmalar detaylandırılmıştır. Bu bölümde, bu modellerin SLAM algoritmalarına nasıl entegre edilebileceği ve sağladıkları avantajlar açıklanmıştır.

Veri Seti Detayları ve Test Sonuçları: Oluşturulan veri setinin özellikleri, farklı algoritmalarla yapılan test sonuçları ve bulgular detaylandırılmıştır. Veri setinin

kapsamı ve test süreçleri hakkında bilgiler verilmiştir.

Adaptif SLAM Uygulamaları ve Testleri: Dinamik Karar Verme Modelinin uygulamaları ve testleri sunulmuş, adaptif SLAM uygulamaları ile elde edilen sonuçlar değerlendirilmiştir. Bu bölümde, modelin gerçek dünya senaryolarında nasıl performans gösterdiği üzerinde durulmuştur. Model çerçevesinde hazırlanan algoritma yörünge çıktıları, Cartographer ve ORB-SLAM3 sonuçları ile EKF füzyon çıktıları karşılaştırılmış ve sonuçlar tartışılmıştır.

Öneriler ve Gelecek Çalışmalar: Tez çalışmasından elde edilen sonuçlara dayanarak öneriler sunulmuş ve gelecekte yapılabilecek çalışmalar hakkında bilgiler verilmiştir.

2. KAYNAK ARAŞTIRMASI

II. Dünya Savaşı sırasında, bağımsız hareket edebilen zeki robotlar kavramı, bilgisayar bilimi ve elektoniğin mekanik mühendislikteki ilerlemelerle birleşmesiyle (mekatronik) ortaya çıkmaya başlamıştır. Bu, mobil robotikte araştırma ve geliştirme üzerine odaklanan bir endüstrinin başlangıç noktasıdır. Bilgisayar ve elektoniğin gelişimine bağlı olarak, robotların özerklik ve belli bir düzeyde zeka sergilemeye başladığı 1960 yılları ise ikinci bir başlangıç noktasıdır (Considine ve Considine, 2012). Bu gelişmeler, otonom robotlar kavramına zemin hazırlamış ve bu robotların kendi başlarına öğrenme, adaptasyon ve karar verme yeteneği kazanmalarını mümkün kılmıştır. Otonom robotlar, karmaşık ve dinamik ortamlarda çalışabilir; çevresel değişikliklere ve beklenmedik durumlara adapte olabilen gelişmiş algoritmalar ve yapay zeka kullanarak tecrübelerine göre davranışlarını değiştirebilirler (Nitzan, 1985).

Otonom robotlar, çevrelerini algılamak ve verilen görevleri yerine getirmek için sensör teknolojilerini kullanırlar. Bir otonom robotun çalışma süreci genellikle üç ana adımdan oluşur: algılama (hissetme), planlama ve yorumlama (işleme) ve hareket (eylem). Bu üç adım sayesinde, otonom robotlar, görevlerini yerine getirmek için yol alma süreçlerini tanımlayan navigasyon işlemlerini gerçekleştirir. Navigasyon becerileri ile çevrelerini sürekli olarak algılayarak, navigasyon süreçlerinde karşılaştıkları dinamik değişikliklere hızlı bir şekilde tepki verebilirler ve bu sayede karşılaştıkları engellere uygun stratejiler geliştirerek hedefine güvenli bir şekilde ulaşabilirler(Ohya vd., 1998). Otonom robotların başarılı bir navigasyon gerçekleştirebilmesi için planlama, konumlandırma ve haritalama işlemleri hayati öneme sahiptir. Planlama, robotun en uygun rotayı belirlemesini sağlarken; konumlandırma, robotun çevresine göre nerede olduğunu hassas bir şekilde belirler. Haritalama ise robotun çevresindeki alanı algılayıp bir harita oluşturmaya yardımcı olur, böylece bilinmeyen ortamlarda güvenle hareket edebilir. Bu süreçler, özellikle SLAM (Simultaneous Localization and Mapping) algoritmalarında kritik bir rol oynar ve bu algoritmaların performansı, robotun durumsal farkındalığı ve çevresini doğru bir şekilde haritalandırmasına bağlıdır. Eğer bu üç işlemten biri

düzgün bir şekilde gerçekleştirilemezse, robotun çevresini doğru algılaması ve güvenli bir şekilde hedefe ulaşması mümkün olmaz, bu da gerçek dünya uygulamalarında hareket kabiliyetini engeller.

Son yıllarda, otonom robotların gezinme ve görevlerini yerine getirebilmesi için robotik ve yapay zeka arasında, robotların insan bilişine benzer şekilde özerk kararlar alabilmesini sağlamak amacıyla artan bir ilişki kurulmuştur. Bu amaçla, araştırmacılar yapay sinir ağlarının otonom robotlarla entegrasyonunu aktif olarak araştırmaktadır (Sharma vd., 2023). Bu araştırmalar, bilişsel robotik alanında önemli bir ilerleme sağlamakta olup, robotların durumsal farkındalık kazanmalarına ve çevrelerine daha duyarlı hale gelmelerine olanak tanımaktadır. Bu zeki ve kendi kararlarını alabilen robotlar, insan yeteneklerini taklit etme potansiyeline sahiptir. Aynı zamanda, SLAM veri setleri üzerine yapılan çalışmalar, otonom robotların dinamik ortamlarda etkili bir şekilde konumlandırılmasını ve haritalanmasını sağlamak amacıyla önemli bir referans noktası oluşturmuştur.

2.1. Otonom Robotlarda Konumlama ve Haritalama

SLAM, robotların veya otonom araçların gerçek zamanlı olarak kendi konumlarını bir harita üzerinde belirleyebilme ve aynı zamanda çevrenin haritasını oluşturma yeteneğine sahip olma sürecidir. SLAM algoritmaları, robotun çevresel algılayıcılardan (kameralar, Lidar, ultrasonik sensörler vb.) gelen verileri kullanarak önceden belirlenmiş bir referans noktasına veya haritaya ihtiyaç duymadan bir ortamı keşfetmesini sağlar. SLAM konusundaki çalışmalar, Smith ve Cheeseman (1986) tarafından başlatılmış ve 1990'lerde Leonard ve Durrant-Whyte (1991) ve Dellaert vd. (1999) gibi birçok önemli çalışma ile popüler hale gelmiştir. Başlangıçta, robotlar genellikle statik ve basit çevrelerde çalışmak üzere programlanmış ve çevreleri hakkında önemli miktarda önceden belirlenmiş bilgiye ihtiyaç duymuşlardır.

1990'lar ve 2000'ler boyunca, SLAM algoritmaları büyük bir gelişim göstermiştir. Yıllar içinde, farklı sensörler (kamera, lazer, radar vb.) kullanılarak yeni yöntemler ortaya çıkmış ve yeni veri temsilleri ile yeni harita türleri oluşturulmuştur. Ayrıca, SLAM alanında çeşitli tahmin teknikleri de geliştirilmiştir. Gelişen bilgisayar işlem kapasitesi, daha iyi algılayıcı teknolojileri ve algoritmik

yenilikler, robotların karmaşık ve dinamik çevrelerde daha etkili bir şekilde navigasyon yapmalarını sağlamıştır. Extended Kalman Filter (EKF) SLAM, Graph-based SLAM ve FastSLAM gibi çeşitli yaklaşımlar, bu dönemdeki önemli ilerlemeler arasında yer almaktadır (Bresson vd., 2017).

Sensörler, çevre modelleme, tanıma, planlama ve hareket süreçlerinin entegre bir parçası olarak, robotların karmaşık bir çevreyi anlamalarını ve görevlerini güvenilir bir şekilde yerine getirmelerini sağlar (Sharma vd., 2023). Yeni sensör teknolojilerinin ortaya çıkışı, SLAM ve mobil robotikteki ilerlemeleri sık sık tetiklemiştir. 2D lazer mesafe ölçerlerin devreye alınması, güçlü SLAM sistemlerinin geliştirilmesine önayak olmuş, 3D Lidarlar ise otonom araçlar gibi modern uygulamaların anahtar bileşeni haline gelmiştir. Son on yılda, görüntü sensörleri üzerine yoğun araştırmalar yapılmış, bu da artırılmış gerçeklik ve görüntü tabanlı navigasyon gibi etkileyici uygulamalara katkıda bulunmuştur. Robotik algılama alanında Lidar'lar ve geleneksel görüntü sensörleri öne çıksa da, derinlik ve olay tabanlı kameralar gibi yeni sensör türleri de SLAM için kullanım potansiyeli göstermektedir ve bu sensörler artık daha yaygın olarak kullanılmaktadır (Chen vd., 2022).

Lidar tabanlı SLAM sistemleri, bir robotun veya aracın çevresini yüksek hassasiyetle algılamasını ve haritalandırmasını sağlayan önemli bir teknolojidir. Lidar sensörleri, çevreyi taramak ve mesafeleri ölçmek için lazer ışınları kullanır. Bu sensörler, 360 derece dönebilen bir platform üzerine monte edilerek çevrenin 2D veya 3D noktalar bulutu halinde yüksek doğrulukla haritalandırılmasını sağlar. Lidar tabanlı SLAM, özellikle karmaşık ve dinamik ortamlarda, diğer sensör teknolojilerine göre daha yüksek doğruluk ve güvenilirlik sunar. Bu sistemler, otonom araçlar, robotlar ve insansız hava araçları gibi uygulamalarda, gerçek zamanlı olarak çevreyi modelleme, konumlandırma ve engellerden kaçınma gibi kritik görevleri yerine getirirken, yüksek çözünürlüklü çevre modelleri oluşturarak güvenli ve verimli navigasyon sağlar.

Lidar sensörlerine dayanan başlıca SLAM algoritmaları, Gmapping (Grisetti vd., 2007), Hector (Kohlbrecher vd., 2011) ve Cartographer algoritmalarıdır (Hess vd., 2016). Gmapping, 2D Lidar verileri kullanarak düşük maliyetli ve etkili bir haritalama çözümü sunar. Bu algoritma, hücresel haritalama yöntemleri kullanarak robotun çevresindeki ortamı grid tabanlı bir harita üzerinde temsil eder. Hector SLAM, özellikle yüksek hızlı hareket eden robotlar için optimize edilmiş bir SLAM

algoritmasıdır. Dikkat çeken bir özelliği ise Hector SLAM'ın kendi odometrisini lazer verilerinden üretme yeteneğidir. Algoritma, Lidar taramalarından elde edilen veri akışını kullanarak robotun hareketini ve yönelimini dinamik olarak hesaplar, böylece GPS veya diğer dış kaynaklardan bağımsız olarak yüksek doğrulukta konum belirleme ve haritalama sağlar. Cartographer, Lidar tabanlı SLAM algoritmaları arasında özellikle büyük ve karmaşık ortamlarda etkili haritalama yapabilmesi ile öne çıkar. Bu algoritmanın dayanıklılığını artıran önemli bir özellik, çevreyi tek bir harita olarak değil, alt haritalar (submaps) şeklinde temsil etmesidir (Hess vd., 2016). Hwang ve Song, veri ilişkilendirme sonuçlarını elde etmek için en yakın komşu veri ilişkilendirme yöntemini kullanan ve ardından mobil robotun duruşunu ve çevre haritasını tahmin etmek için genişletilmiş bir Kalman filtresi kullanan Extended Kalman filtresi tabanlı bir SLAM yöntemidir (Hwang ve Song, 2011). CoreSLAM (Turnage, 2016), hafif ve hızlı bir 2D SLAM algoritmasıdır ve 2D Lidar sensörleri ile çalışmak üzere tasarlanmıştır. CoreSLAM, küçük robotlar ve düşük işlem gücüne sahip cihazlar gibi sınırlı donanım kaynaklarına sahip sistemlerde kullanılmak üzere optimize edilmiştir. CoreSLAM, tarama eşleştirme için bir Monte Carlo algoritması kullanır ve gelen lazer verilerini lazer telemetre cihazından toplanan verilerle birleştirerek düşük kaliteli tanınabilir haritalar çıkarır.

İlk 3D lazer SLAM yöntemi, Carnegie Mellon Üniversitesi'nden Zhang (Zhang vd., 2016; Zhang ve Singh, 2017) tarafından önerilen Lidar odometri ve haritalama algoritmasıdır. Bu yöntemde, 3D Lidar daha yüksek doğruluk ve gerçek zamanlı performansla bir lazer odometresi elde etmek ve bir çevre haritası oluşturmak için kullanılmıştır. Zhang vd. (2014) tarafından geliştirilen LOAM, mükemmel performansıyla bir kilometre taşıdır. İlk olarak, her karedeki 3D Lidar taramasından kenar ve düzlem noktaları çıkarılır ve seçilir. İkinci olarak, hareket bozulmasını düzeltmek için bitişik kareler arasında özellik noktaları eşleştirilir. Eşleştirme sonuçları, karşılıkların hesaplanması için iteratif olarak optimize edilir. Algoritma, Lidar verilerini iki farklı zaman ölçeğinde işler: kısa zaman dilimlerinde robotun hareketini hassas bir şekilde takip ederken, daha uzun zaman dilimlerinde daha geniş ve küresel bir harita oluşturur. Bu iki süreç, algoritmanın hem hızlı hem de doğru bir şekilde çalışmasını sağlar, bu da LOAM'ı özellikle 3D çevresel algılamanın kritik olduğu uygulamalarda (örneğin, otonom araçlar) oldukça etkili kılar.

2D lazer sensörleri kullanarak haritalama yapılabilirse de, bu sensörlerin belirli

dezavantajları vardır. Çünkü 2D lazer sensörü sadece derinlik bilgisine sahiptir ve bir düzlemi tarar, bu nedenle robotun koridorda hareketi sırasında sensörden dönen derinlik bilgisi zamanla değişmediğinde, uygulanan algoritma başarısız olur. Aynı şekilde saf dönüş, agrasif hareketler gibi hareket modeline bağlı zorluklar karşısında başarısız olma eğilimindedir. 3D Lidar sensörlerin gelişmesiyle birlikte, 3D Lidar sensör şemaları, otomatik sürüş ve dış mekan düşük hızlı insansız araçlar tarafından giderek benimsenmiştir. Ancak, 3D Lidar sensörlerin yüksek maliyeti nedeniyle, 2D Lidar'lar hala iç mekan sahnelerinde SLAM için ağırlıklı olarak kullanılmaktadır. Diğer ucuz bir sensör olan kameraların SLAM algoritmalarında kullanımı, robotik ve otonom sistemler için önemli bir dönüm noktası olmuştur.

İlk SLAM algoritmaları genellikle Lidar gibi mesafe ölçüm sensörlerine dayanıyordu, ancak kameraların eklenmesiyle birlikte, çevresel algılama ve haritalama kabiliyetleri büyük ölçüde gelişti. Kameralar, hem geniş bir görüş alanı sunarak çevrenin daha ayrıntılı bir şekilde algılanmasını sağladı hem de görsel özelliklerin (örneğin, köşeler, kenarlar ve yüzeyler) takibi ile robotların çevrelerini daha hassas ve doğru bir şekilde modellemesine imkan tanıdı. Ayrıca, kameralar ile elde edilen görüntüler, renk, doku ve şekil gibi ek bilgi katmanları sunar, bu da daha karmaşık ortamlarda daha güvenilir ve hassas haritalama yapılmasına olanak tanır.

MonoSLAM (Davison vd., 2007), tek gözlü görselin gerçek zamanlı olarak geri kazanımını için ilk SLAM sistemidir; ön uçta seyrek özellikleri izlemek için genişletilmiş bir Kalman filtresi kullanır, böylece kameranın mevcut durumunu ve tüm özellik noktalarının durum değişkenlerini günceller. Bu çalışma, kameraların SLAM algoritmalarındaki potansiyelini göstermiş ve ardından gelen araştırmalara ilham kaynağı olmuştur. Klein ve Murray (2007), kamera poz izleme ve özellik noktası haritası oluşturmanın paralel olarak gerçekleştirildiği gerçek zamanlı çalışan bir paralel izleme ve haritalama (PTAM) algoritması sunulmuştur. Bu algoritmayla tasarlanan sistemler, daha yüksek çerçeve hızlarında izlenebilen, doğruluk ve dayanıklılık açısından model tabanlı sistemlerle karşılaştırıldığında üstün performans gösteren, binlerce yol işaretini içeren detaylı haritalar üretilmiştir. De Croce vd. (2019), stereo PTAM için dağıtılmış bir mimari olan dağıtılmış stereo PTAM (DS-PTAM) adlı bir stereo SLAM sistemini önermiştir. Bu sistemde, giriş bilgileri iki kamera tarafından alınır ve izleme ve haritalama işlemleri dağıtılmış bir sistemde gerçekleştirilir. DS-PTAM sistemi, konumlandırma ve haritalama modüllerinin uzaktaki baz istasyonlarında çalıştırılmasına izin verir, böylece araç

işlemcisi üzerindeki yükü azaltır. Ancak, kamera da kendi dezavantajlarına sahiptir ve geniş bir aralıkta dönerken özellikleri kaybetme ve faklı aydınlatma durumları, özniteliksiz sahneler gibi zorlu şartlarda etkisiz hale gelmektedir. Görsel-SLAM ayrıca doğru bir derinlik elde etmek için çok fazla bilgiye ihtiyaç duyar, bu da yanlış konumlandırma ve izleme hatasının yanı sıra hatalı haritanın yeniden yapılandırılmasına yol açabilir. Bu durumlarda, lazer üstün özellikler sergiler. Son yıllarda, her sensörün kendi sınırlamaları olduğu için, araştırmacılar çoklu sensör füzyonuna dayalı birçok SLAM haritalama algoritması geliştirmiştir.

Lidar ve görsel tabanlı SLAM algoritmaları gözden geçirildiğinde öne çıkan problemler şunlardır: İlk olarak, mekan dar ve dokusal özelliklerden yoksun olduğunda, stabilite ve doğruluk garanti edilemez, bu da büyük kümülatif hatalara ve kötü haritalama sonuçlarına yol açar. İkinci olarak, hareket bozulmasının düzeltilmesi hala zordur (Qiu vd., 2019). Bu koşullar altında, sağlam ve doğru bir konumlandırma ve haritalama yöntemi gereklidir. Görsel-SLAM ve Lidar-SLAM'ın gelişimi oldukça iyi bir seviyededir ve birçok gerçek dünya sahnesinde kullanılabilir, ancak yine de göz ardı edilemeyecek dezavantajları vardır. Görsel-SLAM, düşük dokulu veya karanlık sahnelerde zayıf sağlamlığa sahiptir. Lidar-SLAM hem iç hem de dış mekanlarda iyi sonuçlar elde edebilir, ancak birçok Lidar sistemi çok yakın mesafelerdeki nesnelere algılamakta güçlük çeker. Lidar-SLAM, oldukça dinamik veya seyrek özelliklere sahip sahneler için uygun değildir. Aynı zamanda Lidar tabanlı SLAM sistemi, yüksek ölçüm doğruluğu ve aydınlatma koşullarına karşı duyarsızlık avantajlarına sahiptir. Bu iki sensör güçlü tamamlayıcı avantajlara sahiptir ve iki sensörü birleştirerek verimli ve doğru haritalama ve navigasyonu sağlayan yöntemlerin belirlenmesi önemlidir.

Değişen çevre ve yoğun hareketle başa çıkmak için çoklu sensör füzyonuna dayalı çalışmalar önerilmiştir. Tipik olarak, Lidar ve diğer sensör türleri (IMU veya görsel sensörler) sıkı bir şekilde birleştirilerek kısıtlamalar sağlanır ve pozlar ortak olarak tahmin edilir. Bu yöntemler ile zorlu ortamlarda daha küçük kümülatif hata ve daha yüksek doğruluk elde etmiştir.

Karto SLAM, Lidar verilerini IMU verileriyle birleştirerek sensör füzyonu gerçekleştirir. Lidar, çevrenin bir anlık görüntüsünü sunarken, IMU, robotun hareketine dair sürekli bilgi sağlar. Karto SLAM, haritayı grafik olarak temsil eder ve her düğüm, robotun yolundaki bir köşe noktası ve bir sensör ölçüm veri kümesini temsil eder (Konolige vd., 2010). Başarılı sensör füzyonları, her bir sensörün gücüne

dayanır, ancak füzyon verilerinin işlenmesinde gelişmiş algoritmalar ve makine öğrenme teknikleri olmadan algılama ve hissedebilme potansiyellerini tam olarak ortaya çıkarmak zor olacaktır (Ahmed vd., 2019). Shamwell vd. (2019), RGB-derinlik (RGB-D) görüntülerini atalet ölçüm birimlerinden alınan mutlak yörünge tahmini ile birleştirmek için denetimsiz bir sinir ağı yaklaşımını benimsemiştir. Ağ, bir Atalet ölçüm biriminin (IMU) ölçümlerini entegre etmeyi öğrenir ve hipotez yörüngeleri üretir, bu yörüngeler ölçeklenmiş görüntü projeksiyon hatasının Jacobian matrisi ile çevrimiçi olarak düzeltir. Westfechtel vd. (2019), Lidar taraması ve kamera verilerini senkronize etmek için senkronize Lidar odometri ve haritalama algoritmasını kullandılar ve 3D lazer verilerini 2D görüntü verilerine projekte etmek için bir sinir ağı algoritması kullanarak görüntünün anlamsal bilgisini topladılar ve 3D RGB-SLAM oluşturdular.

Zhang ve Singh (2015) tarafından bir Görsel-Lidar odometri ve haritalama mimarisi yayınlandı. Görsel-Lidar odometrisinin genel çerçevesi, agresif hareket veya görsel özelliklerin geçici eksikliği altında sağlamlığı korumak için önerilmiştir. Görsel odometri ön ego hareketini tahmin ederken, Lidar odometrisi aynı anda nokta bulutu kayıt sonuçlarını iyileştirir. Mur-Artal vd. (2015) tarafından sunulan klasik bir çalışma, çeşitli iç ve dış mekan ortamları için gerçek zamanlı özellik tabanlı monoküler SLAM sistemidir. Bu çalışmadaki fikir, yüksek doğruluğa sahip kompakt haritalar oluşturmak için özellik noktalarını ve ana kareleri seçme stratejisine dayanır. ORB-SLAM hesaplamaları hızlandırılmış segment testi ve döndürülen ikili sağlam bağımsız temel özellikler, ayrıca görsel hız ölçüm ve döngü algılama için bir ORB sözlüğü içerir.

Zhang vd. (2016), dejenerasyon sorunlarına odaklanan bir kamera-Lidar füzyon şeması tanımladılar. Dejenerasyon, görsel sensörler için düşük dokusal özellikler ve menzil sensörleri için kötü geometrik yapılar altında meydana gelir. Bu çalışmada, optimizasyon tabanlı problemler için dejenerasyonu hafifletmek amacıyla çevrimiçi bir yöntem tanımlanmıştır. Başka bir çalışmada (Zhang ve Singh, 2018), multi-line Lidar, kamera ve IMU sensörlerini birleştiren Lidar-Görsel-Atalet odometri algoritmasını önermiştir. Bu algoritma, kamera görüntülerinin ve IMU ölçümlerinin kısıtlamalarını kullanarak yüksek frekanslı duruş çıktısını sağlar; ardından lazer nokta bulutları arasındaki özellik eşlemesini kullanarak düşük frekanslı duruş çıktısını ve haritalama çıktısını sağlar; son olarak 3D nokta bulutu haritası oluşturulur. Bu çalışma, pozları kaba ve ince olarak hesaplayan ardışık, çok

katmanlı bir hat kullanarak sensor dejenerasyonunu otomatik yeniden yapılandırma ile çözebildiğini kanıtlamıştır.

Shan vd. (2020), yüksek doğrulukta, gerçek zamanlı mobil robot yörünge tahmini ve harita oluşturma sağlayan sıkı bağlı bir lazer radar odometresi çerçevesi olan düzeltme ve haritalama yoluyla Lidar odometrisi önerdiler. Düzeltme ve haritalama yoluyla Lidar odometrisi, bir sistemin yerel aralıktaki gerçek zamanlı performansını etkili bir şekilde iyileştirmiştir. Liu ve Shen (2017), yükseklik tahmincisi başlatma yöntemi olarak spline tabanlı monoküler Görsel-Atalet navigasyon sistemi önerdiler. Bu yöntem, daha erken yakınsamayı başarmak için gereken parametre tahminlerinin sayısını açıkça azaltmıştır. Qin vd. (2018) tarafından geliştirilen monoküler Görsel-Atalet sistemi (VINS), kamera ve IMU kullanır. Ana yenilikçi bölümü, IMU ön entegrasyonu ve sıkı bağlanmış özellik gözlem yöntemidir. Bir döngü düzeltme modülü temel alınarak konumlandırma hatası minimize edilmiştir. Böylece kamerayı yeniden konumlandırabilmekte ve mevcut haritadan kameranın mutlak duruşunu elde edebilmektedir. Ayrıca, kaymayı düzeltmek ve küresel tutarlılığı sağlamak için dört serbestlik dereceli bir duruş grafiği optimizasyonu yapmaktadır.

Zuo vd. (2019), Lidar, IMU ve Görsel odometri bilgilerini birleştirmek için EKF kullanmışlardır, ancak farklı sensörler için farklı hız ve özellikler göz önünde bulundurulmamıştır. Çizgiler, noktalardan daha sağlam geometrik kısıtlamalar sağladığından, Zou vd. (2020) tarafından bir RGB-D SLAM sistemi önerilmiştir. Bu çalışma, düşük dokulu sahnelerde nokta ve çizgi özelliklerini birleştirerek tahmin hatasını çözmektedir ve Mu vd. (2020), 2D Lidar nokta bulutu ve RGB-D kamera nokta bulutuna yönelik ilgili stratejiyi tasarlamak için Kalman filtresi (UKF) kullanmıştır. Deneyler, tasarlanan çoklu sensör SLAM çerçevesinin etkili olduğunu göstermiştir. Geneva vd. (2018), lazer ve görselin eşzamansız füzyonunu gerçekleştirmek için faktör grafik tabanlı bir optimizasyon çerçevesi tasarlamışlardır. Önerilen sensör füzyon sistemi, gerçek dünya deneysel veri setinde performans iyileşmesi sağlamıştır. Ancak, bu yöntemler farklı sahnelerde lazer ve görsel arasındaki güven farkını dikkate almamıştır. Örneğin, lidar ve görsel sensörler, dönme ve düz çizgide hareket sırasında farklı duyarlılıklara sahiptir.

Ye vd. (2019) tarafından sunulan LIO, IMU ve Lidar ile sıkı bir şekilde bağlanmış önemli bir çalışmadır. Bu çalışmada, başlangıç pozunu IMU ile tahmin edilip ardından hareket bozulması düzeltilmiştir. Daha sonra, daha doğru bir poz elde

etmek için çoklu karelerin pozlarını kısıtlayan bir yerel harita oluşturulmuştur. Sonuç olarak, karmaşık ortam veya agresif hareketler altında konumlandırma doğruluğu büyük ölçüde artırılmıştır. Ancak, VINS gibi, düzlemde düzgün hareket ettiğinde, IMU iyi kısıtlanamaz ve sistemde rastgele hatalar birikir. İç mekan sahnelerinde düzlem ve kenar özelliklerini birleştiren bir düzlem-kenar-SLAM yaklaşımı (Sun vd., 2021), dokusuz veya karanlık sahnelerde iyi çalışmakta ve sensör gürültüsüne karşı dayanıklılık göstermektedir.

Rui Huang (2021) karmaşık ortamlara daha iyi uyum sağlamak ve dejenerasyonu engellemek için değişken çevreler ve yoğun hareket altında yüksek konumlandırma doğruluğu ve sağlamlık sağlayan uyarlanabilir bir algoritma önermiştir. Algoritma iki bölümden oluşmaktadır: bozulmayı baskılama için kendi kendine uyarlanabilir modül ve bozulma düzeltme modülü. Algoritma, çeşitli ortamlarda daha yüksek yerelleştirme ve haritalama doğruluğu elde etmek için SLAM sistemindeki gürültünün etkisinin azaltılması, sistem yoğun hareket halinde değilse, hesaplama yükünü azaltmak için yerel harita optimizasyonunun iptali ve yerel haritada bir hareket distorsiyon düzeltmesi gibi farklı stratejiler uygulamıştır. Böylece uzun mesafe testlerinde veya ani hareket değişikliklerinde optimize edilmiş bir pozisyon elde edilmiştir. Deneysel sonuçlar, önerilen algoritmanın çeşitli sahnelere uyum sağladığını ve yüksek yerelleştirme ve haritalama doğruluğunu koruduğunu kanıtlamıştır. Özellik noktası eşleştirmeye dayalı algoritma çerçeveleri, yaygın olarak kullanılmaktadır. Ancak, karmaşık ve değişken iç mekan ortamlarında, özellik noktası eşleştirmeye dayalı SLAM şu anda iki ana sorunla karşı karşıyadır: SLAM sistemine dinamik nesnelerin neden olduğu müdahaleler nedeniyle poz tahmininin doğruluğunun azalması ve zayıf dokulu sahnelerde özellik noktalarının eksikliği nedeniyle izleme kaybı yaşanmasıdır. Bu sorunları ele almak için, (Chang vd., 2022) ORB-SLAM3'e dayalı sağlam ve gerçek zamanlı bir RGB-D SLAM algoritması sunmuştur.

Xing vd. (2022) çalışmalarında, eşzamansız füzyonun yanı sıra sensörlerin özelliklerinden kaynaklanan güven farkını da dikkate almıştır. 2D lazer ve dürbün kameralarına dayalı olarak, ağırlıklı eşzamansız bir füzyon algoritması önerilmiştir. Algoritma, farklı sensörlerin frekans edinme özelliklerini dikkate alarak ve farklı çalışma durumlarında farklı güven seviyelerine sahip olmalarını sağlamaktadır. Bu, sensörlerin farklı örnekleme hızlarında çalışmasını sağlamakla beraber, belirli bir sahneden kaynaklanan performans kaybını da etkili bir şekilde önlemiştir. Deneysel,

önerilen algoritmanın doğruluk ve dayanıklılığa sahip olduğunu göstermiştir. Bu çalışma aynı zamanda sabit bir füzyon yaklaşımındansa adaptif bir yaklaşımın daha etkili olduğunu ortaya koymuştur.

Li vd. (2020) çalışmalarında, derin öğrenmeye dayalı SLAM'ı ayrıntılı olarak özetlemiş ve geleneksel yöntemlerin eksikliklerine dikkat çekmiştir. Li vd. (2019), dönme açısı büyük olduğunda doğruluğun önemli ölçüde azaldığı problemi çözmek için bir 2D Lidar ve bir atalet ölçüm birimi ile donatılmış bir mobil robota tekrarlayan bir evrişimli sinir ağı (RCNN) uygulamıştır.

Günümüzde, SLAM algoritmaları yapay zeka ve derin öğrenme tekniklerini entegre ederek sürekli değişen çevreleri algılamak ve yorumlamak için daha da gelişmektedir. Bu, otonom sistemlerin, daha önceki algoritmaların başa çıkamadığı belirsiz ve karmaşık ortamlarda bile güvenilir ve doğru bir şekilde çalışabilmesini sağlayarak SLAM teknolojisinin sınırlarını zorlamaktadır. Özellikle 2016'dan bu yana, derin öğrenme alanındaki artışla birlikte araştırmacılar, derin öğrenme yöntemlerinin bilgisayarla görmede uygulanmasının geleneksel yöntemlerle çözülmesi zor olan sorunları büyük ölçüde azaltabileceğini keşfetmiştir.

SLAM algoritmalarına dair mevcut problemler, beklentiler ve yeni araştırma sorularını topluluğun önde gelen araştırmacılarının (Cadena vd., 2016; Sousa vd., 2023; Zhang vd., 2023) görüşleri ve deneyimlerimizle birlikte ele aldığımızda belli bazı başlıklar oluşmaktadır.

Başarısızlığa Karşı Korunmalı (Failsafe) SLAM ve Kurtarma: SLAM arka ucunda kaydedilen ilerlemelere rağmen, mevcut SLAM çözümleri hala aykırı değerler karşısında savunmasızdır. Bu durum, neredeyse tüm sağlam SLAM tekniklerinin, konveks olmayan maliyetlerin yinelemeli optimizasyonuna dayanmasından kaynaklanmaktadır. Bunun iki sonucu vardır: birincisi, aykırı değer reddi sonucu, optimizasyona verilen başlangıç tahmininin kalitesine bağlıdır; ikincisi, sistem doğası gereği kırılmalıdır: tek bir aykırı değer dahil edilmesi, tahminin kalitesini düşürür ve bu da daha sonra aykırı değerlerin ayrımını zorlaştırır. Bu tür hatalar, doğru olmayan bir doğrusallaştırma noktasına yol açar ve bu noktadan kurtulmak, özellikle artımlı bir kurulumda, kolay değildir.

İdeal bir SLAM çözümü, hatasız ve hatanın farkında olmalıdır; yani sistem, yaklaşan bir hatanın (örneğin, aykırı değerler veya bozulmalar nedeniyle) farkında olmalı ve doğru işlemi yeniden sağlama mekanizmaları sunmalıdır.

Donanım Arızasına Karşı Sağlamlık: Donanım arızalarını ele almak,

SLAM'ın kapsamı dışında görünsede, bu arızalar SLAM sistemini etkiler ve bu sistem, sensör ve hareket arızalarını tespit etme ve hafifletme konusunda önemli bir rol oynayabilir. Bir sensörün doğruluğu, arızalanma, norm dışı koşullar veya yaşlanma nedeniyle bozulduğunda, sensör ölçümlerinin kalitesi (örn. gürültü, önyargı) arka uçta kullanılan gürültü modeliyle uyuşmaz, bu da kötü tahminlere yol açar. Bu, farklı araştırma sorularını doğurur: bozulmuş sensör işletimini nasıl tespit edebiliriz? Sensör gürültü istatistiklerini (örneğin, kovaryanslar, önyargılar) buna göre nasıl ayarlayabiliriz? daha genel olarak, farklı sensörlerden gelen çelişkili bilgileri nasıl çözebiliriz? Bu, sensör verilerinin yanlış yorumlanmasının insan hayatını riske atabileceği güvenlik açısından kritik uygulamalarda (örn. otonom araçlar) önemlidir.

Zamanla Değişen ve Deforme Olabilen Haritalar: Ana akım SLAM yöntemleri, rijit ve statik dünya varsayımıyla geliştirilmiştir; ancak gerçek dünya, hem dinamikler hem de nesnelere doğası gereği deformasyonu nedeniyle rijit değildir. İdeal bir SLAM çözümü, ortamın dinamiklerini, rijit olmama durumunu da içerecek şekilde mantıklı bir şekilde ele almalı, uzun zaman dilimlerinde her tür arazide haritaları oluşturmalı ve bunu gerçek zamanlı olarak yapabilmelidir. SLAM topluluğunda, Newcombe vd. (2015) küçük ölçekli yeniden yapılandırma için rijit olmayan durumu ele almıştır. Ancak, büyük ölçekli rijit olmayan harita problemi için hala çalışmalara ihtiyaç vardır.

Otomatik Parametre Ayarlama: SLAM sistemleri (özellikle veri ilişkilendirme modülleri) belirli bir senaryo için düzgün çalışabilmeleri için kapsamlı parametre ayarlaması gerektirir. Bu parametreler, özellik eşleştirmeyi kontrol eden eşikler, RANSAC parametreleri ve grafiğe ne zaman yeni faktörler ekleneceği veya eşleşmeleri aramak için bir döngü kapanış algoritmasının ne zaman tetikleneceği gibi kriterileri içerir. SLAM'in keyfi senaryolarda kullanıma hazır çalışması gerekiyorsa, ilgili parametrelerin otomatik ayarlanması için yöntemler dikkate alınmalıdır.

Harita Temsili: Uzun vadeli operasyon sırasında haritanın nasıl depolanacağı yeterince araştırılmamış bir sorudur. Bellek sıkı bir kısıtlama olmasa bile, örneğin veriler bulutta depolandığında, nokta bulutları veya hacimsel haritalar gibi ham temsiller bellek açısından israf olur; benzer şekilde, görsel tabanlı SLAM için özellik tanımlayıcılarını depolamak zahmetlidir. Sıkıştırılmış harita temsilleri (Lynen vd., 2015) ve bellek açısından verimli yoğun yeniden yapılandırma

(Klingensmith vd., 2015) için bazı ilk çözümler önerilmiştir.

Öğrenme, Unutma ve Hatırlama: Uzun vadeli haritalama için ilgili bir açık soru, haritaya dahil edilen bilgilerin ne sıklıkla güncellenmesi gerektiği ve bu bilgilerin ne zaman güncelliğini yitirip atılabileceğine karar verilmesidir. Unutmanın uygun olduğu durumlar var mıdır? Hangi durumda, ne unutulabilir ve neyi muhafaza etmek esastır? Haritanın bazı bölümleri bırakılabilir mi ve gerektiğinde hatırlanabilir mi? Dinamik ortamlarda, zorluk iki yönlüdür. İlk olarak, SLAM sistemi değişiklikleri algılamalı, göz ardı etmeli veya izlemelidir. İkinci zorluk, SLAM sisteminin kalıcı veya yarı kalıcı değişiklikleri modellemesi ve haritayı ne zaman ve nasıl güncelleyeceğini anlamasıdır. Dinamiklerle başa çıkan mevcut SLAM sistemleri, aynı yerin birden fazla (zamana bağlı) haritasını tutar (Dayoub vd., 2011) veya tek bir temsili, zamana bağlı bir parametre ile parametrize eder (Krajník vd., 2014).

Dayanıklı Dağıtılmış Haritalama: Tek robot vakasında aykırı değerlerin reddi için yaklaşımlar önerilmişken, çoklu robot SLAM literatüründe aykırı değerler probleminde pek değinilmemektedir. Yanlış ölçümlerle başa çıkmak iki nedenden dolayı özellikle zordur. İlk olarak, robotlar ortak bir referans çerçevesini paylaşmayabilir, bu da yanlış döngü kapanışlarını tespit edip reddetmeyi zorlaştırır. İkinci olarak, dağıtılmış yapılandırmada robotlar, çok kısıtlı ve yerel bilgilerden aykırı değerleri tespit etmek zorundadır. Bu sorunu ele almaya yönelik erken bir girişim (Fox vd., 2006)'dir, bu çalışmada robotlar, bilgi birleştirmeden önce bir buluşma stratejisi kullanarak konum hipotezlerini aktif olarak doğrular. Indelman vd. (2016), yanlış ölçümler karşısında ortak bir referans çerçevesi oluşturmak için olasılıksal bir yaklaşım önerir.

Kaynak Kısıtlı Platformlar: Mevcut SLAM algoritmalarını robotik platformların ciddi hesaplama kısıtlamalarına sahip olduğu durumlara uyarlamak, nispeten az araştırılmış bir başka sorundur. Bu sorun, platform boyutu küçüldüğünde büyük önem taşır, örneğin, cep telefonları, mikro hava araçları veya robot böcekler (Wood vd., 2013). Birçok SLAM algoritması bu tür platformlarda çalıştırmak için yüksek hesaplama maliyeti gerektirir. Benzer sorunlar, çoklu robot ortamlarında da ortaya çıkar: Dar bant genişliği kısıtlamaları ve iletişim kesintileri karşısında çoklu robot ekiplerinin güvenilir operasyonunu nasıl garanti edebiliriz? Cieslewski vd. (2015)'nin "versiyon kontrolü" yaklaşımı bu yönde ilk çalışmadır.

SLAM sistemi, yeni veriler geldikçe konum tahminini ve haritayı sürekli

olarak güncellemelidir. Aynı zamanda, çevreyle ilgili öğrendiği bilgileri entegre ederek ve geçmiş deneyimlerinden yararlanarak kendini iyileştirebilen adaptif bir öğrenme mekanizmasına sahip olmalıdır. Bu sınırlamaların üstesinden gelmek için, robotik, bilgisayar bilimi, yapay zeka ve elektronik mühendisliği gibi çeşitli disiplinler arasında entegre ve yenilikçi çözümler geliştirilmesi gerekmektedir Aynı zamanda, kullanılan algoritmaların ve donanımın sürekli iyileştirilmesi, değişen çevreyle başa çıkmak için dinamik stratejiye sahip yeni bir yöntemin olması, SLAM sistemlerinin daha da güçlü ve kullanışlı hale gelmesini sağlayacaktır.

2.2. Bilişsel Robotik ve Durumsal Farkındalık

Otonom robotların tarihçesi, 20. yüzyılın ortalarına dayanır. 1980'lerde, robotik alanında yapay zekâ ve makine öğrenmesi gibi teknolojilerin entegrasyonu yapılmaya başlanmıştır. Yapay zekanın algısal zekadan bilişsel zekaya geçişiyle birlikte, robotlar yeni bir bilişsel keşif aşamasına girmiştir. Bilişsel robotlar, insan benzeri düşünme, mantık ve bilişsel öğrenme yeteneklerine sahiptir. Çevreleriyle etkileşime girerek dinamik tepkiler verebilirler. Bu, çok disiplinli teknolojilerin işbirliğini gerektiren karmaşık bir sistemdir. Mevcut araştırmalar genellikle algılama, bilgi edinme, yol planlama ve eylem kontrolüne odaklanmaktadır. Bilişsel robotik üzerindeki çalışmalar, bilişsel özerklik ve davranış kontrolü alanında bazı ilerlemeler kaydetmiştir (Cui vd., 2022a). Ancak, akıllı bilişsel hesaplama ve eylem kontrolünü bütüncül olarak ele alan çalışmalar henüz yeterince gelişmemiştir.

Bilişsel robotlar, yalnızca hafıza ve planlama yapan mekanik otomatik robotlardan farklıdır; aynı zamanda algılama, eylem ve öğrenme ile akıl yürütme yeteneklerine sahip yüksek seviyeli akıllı sistemlerdir. Bir bilişsel robot, işbirliği yeteneğine sahip birçok akıllı bileşenden oluşur ve kapsamlı bilişsel yetenekler sağlar. Çevre, insan geri bildirim ve eylemlerin sonuçları gibi faktörlerden etkilenir. Bu robotların çekirdek teknolojileri, haritalama ve konumlama, görüntü sınıflandırma, hedef nesne tespiti, konuşma tanıma, doğal dil anlama ve pekiştirmeli öğrenme gibi alanları kapsar.

Bilişsel robotlar, karmaşık ve değişken ilişkilerin bulunduğu çok disiplinli sistemlerdir. Bu nedenle, sistem teorisi rehberliğinde, bileşenler arasındaki etkileşimler ve kısıtlamalar dikkatlice ele alınmalıdır. Bilişsel robotun bileşenleri

arasındaki etki faktörleri optimize edilmeli ve genel etkiyle birleştirilmelidir. Enerji ve bilgi alışverişi tam olarak dikkate alınmalı ve dış çevrenin bu alışveriş üzerindeki etkisi göz önünde bulundurulmalıdır. Oluşturulacak model ve mimariler, robotların uzay tanımlamasını, koordinat dönüşümünü, görev yönetimini, pozisyon ve kontrolünü analiz ederek karmaşık sorunları çözmek için sistematik rehberlik sağlamalıdır (Scharfe-Scherf vd., 2022).

Otonom robotların çevresel bilgilere dayanarak hızlı ve doğru kararlar almaları gerekmektedir. Engel önleme, yol planlama ve trafik katılımı gibi görevler, karmaşık ortamlarında güvenli ve verimli operasyonlar sağlamak için kritik öneme sahiptir.

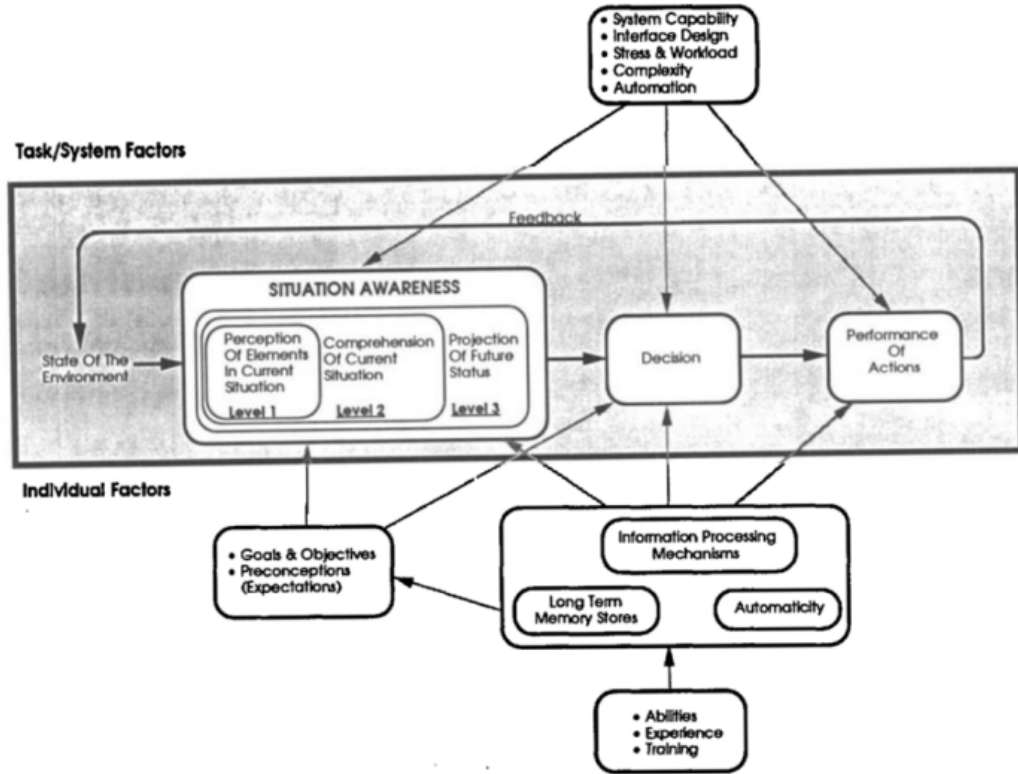
Bu bağlamda, araştırmacılar ve uygulayıcılar, otonom sistemlerde algılama, karar verme ve kontrolle ilgili zorlukları ele almak için çeşitli algoritma modelleri ve teknolojik yöntemleri aktif olarak araştırmaktadır. Geliştirilen algoritmalar, genellikle açık çevrimli sistemler veya olasılıksal yaklaşımlar kullanarak sistemin çeşitli etkenleri (örneğin çevre etkisi) dikkate almasına olanak tanır. Zamanla, robotların bilişsel öğrenme süreçlerine geçişiyle, algıdan davranışa ve davranıştan algıya kapalı bir döngü oluşturulmuştur (Gonzalez, 2017). Bu süreçte, algının davranışa hizmet etmesi kadar, davranışın da algıya katkıda bulunması önemlidir.

Bilişsel zeka, bilişsel robotların temel bilimidir ve akıllı bilimlerin gelişiminde ileri bir aşamadır (Cui vd., 2022b). Bu sistemler, insan bilişsel modelini temel alarak, analiz, karar verme, planlama, çıkarım ve rota planlama gibi işlemleri gerçekleştirir. Nihai hedef, insanın temel zekasını ve yeteneklerini taklit etmektir. Bilişsel çerçeve, bellek pekiştirme, özellik işleme, anlama hesaplaması, çıkarım tahmini, ilişki önerisi, karar yargısı ve geri bildirim öğrenmesini içerir. Bu araştırma çabaları, otonom teknolojilerin gelişimini teşvik etmeyi ve pratik uygulama senaryoları için daha iyi çözümler sağlamayı amaçlamaktadır. Bu doğrultuda, daha doğru, verimli ve güvenilir algoritma modellerinin geliştirilmesi, otonom sürüş sistemlerinin performansını ve güvenilirliğini artırmaya yönelik önemli bir adımdır.

Durumsal farkındalık, bir kişinin çevresindeki unsurları algılaması, bu unsurların anlamlarını kavraması ve gelecekteki durumlarını projekte etmesi olarak tanımlanır (Endsley, 1995). Endsley'in Dinamik Karar Verme Modeli, bu durumsal farkındalığın karar verme süreçlerinde ne kadar kritik olduğunu vurgular. Model, bir karar vericinin çevresel bilgileri nasıl algıladığını, bu bilgiyi nasıl dönüştürdüğünü ve karar verme sürecinde nasıl kullanıldığını detaylandırır.

Bu modelde, makine çevreyi algılar, ancak operatör bu bilgileri anlamlandırır ve gelecekteki durumları tahmin eder. Karar verici, çevreden aldığı bilgiyi alternatifler oluşturmak, tercihler belirlemek ve seçenekleri değerlendirmek için kullanır. Bu süreç, kapalı döngü olarak işler: eylemler çevrede değişikliklere neden olur ve geri bildirim bu değişikliklerin sonuçlarını sağlar. Bu geri bildirim, önceki kararları pekiştirmek veya değiştirmek için kullanılır, böylece öğrenme ve karar verme süreci sürekli olarak gelişir.

Dinamik karar verme, deneyime dayalı ve geri bildirime bağımlı bir öğrenme sürecidir. Alternatifler zaman içinde ortaya çıkar ve karar verme süreci bu değişen koşullara bağlı olarak evrilir. Bu etkileşimli ve kapalı döngü karar verme modeli, bilişsel psikolojide bir uyarının sunulması ile seçim arasında gerçekleşenleri ele alan etkili bir yaklaşımdır (Gonzalez, 2012).



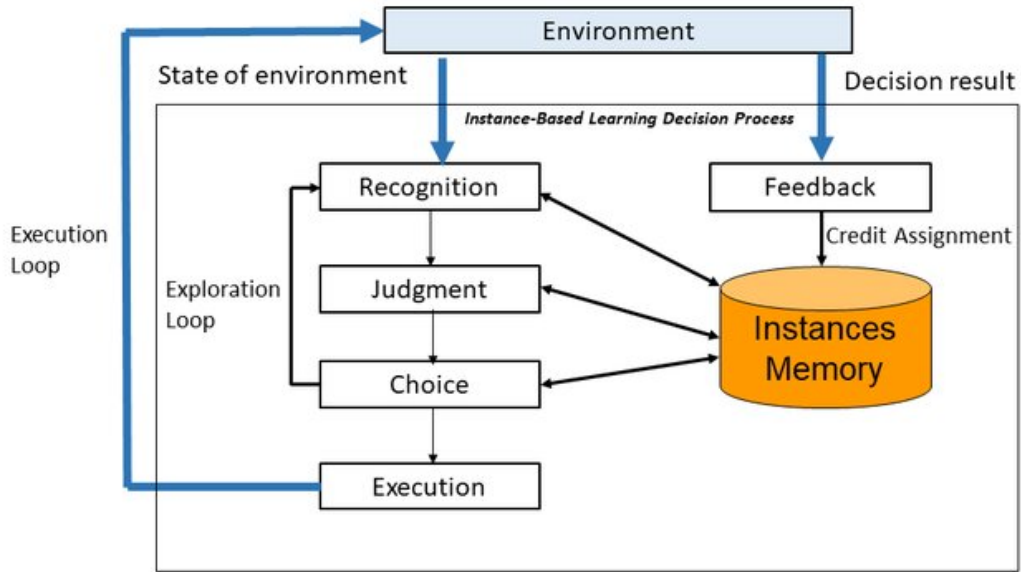
Şekil 2.1. Dinamik Karar Verme Modeli (Endsley, 1995)

Örneğe Dayalı Öğrenme Teorisi (IBLT) (Gonzalez vd., 2003, 2005), dinamik görevlerde karar verme davranışını yeniden üretmek için geliştirilmiştir. Bu teori, insan ve çevre etkileşimleriyle geri bildirim döngüsü süreci aracılığıyla deneyimlenen olayların bellekte "örnekler" olarak saklanmasını içerir.

IBLT, dinamik görevlerde öğrenmeyi, karar alma olaylarını deneyimleyerek hafızada üç öğeden oluşan temsiller olarak depolayarak karakterize eder: Durum (S),

bir dizi nitelik veya ipucu ile tanımlanır; Karar (D), Durum S’de alınan eylemi temsil eder; ve Fayda veya Değer (U), durum S’de karar D’nin beklenen veya alınan değeridir. Bu süreç, tanıma (geçmiş örneklerin benzerlik temelli geri çağırılması), yargılama (bir durumda kararın beklenen faydasının değerlendirilmesi), seçim (optimal mevcut alternatifi seçme kararı), yürütme (seçilen kararın uygulanması) ve geri bildirim (karar örneklerinin değerinin güncellenmesi) adımlarını içerir.

IBLT, geçmiş benzer durumlardan elde edilen deneyimlere başvurarak ve bu geçmişte işe yarayan kararları uygulayarak dinamik görevlerde karar verme olasılığı sunar. Teorinin en önemli katkısı, deneyimlerin nasıl inşa edileceği, geri çağırılacağı, değerlendirileceği ve etkileşimler sırasında nasıl pekiştirileceği konularında sağladığı ayrıntılı öğrenme süreçleri ve mekanizmalarıdır. IBLT’nin basitliği ve geniş kapsamlı tahmin yeteneği, teorinin önemini vurgular (Gonzalez vd., 2011; Gonzalez, 2012, 2017). Bu model, sadece dinamik karar alma süreçlerinde kullanılan yapıları ve süreçleri tanımlamakla kalmaz, aynı zamanda bu kararların neden alındığını ve alternatif karar yollarını açıklar.



Şekil 2.2. IBLT modeli (Gonzalez vd., 2003)

Araştırmacılar, tek seferlik kararlar yerine tekrarlanan kararları içeren ve olası sonuçları gözlemlenen sonuçlardan tahmin etmeyi içeren deneysel paradigmaları tercih ederler. Bu tür paradigmalar, alternatiflerin, sonuçların ve olasılıkların bilinmediği gerçek dünya durumlarında karar vermenin doğal süreçlerini temsil eder.

Modeller, insanların bağlama ilişkin neden-sonuç bilgilerini öğrendiğini

kabul eder ve genellikle yapısal bilgi veya kuralları ana öğrenme biçimi olarak sunmazlar. Ayrıca, tüm modeller, tanıdık desenlerin geçmiş deneyimlerden tanınmasının önemini vurgular; yani, kararlar, benzer geçmiş durumlardan elde edilen çözümlerle deneyimden alınır.

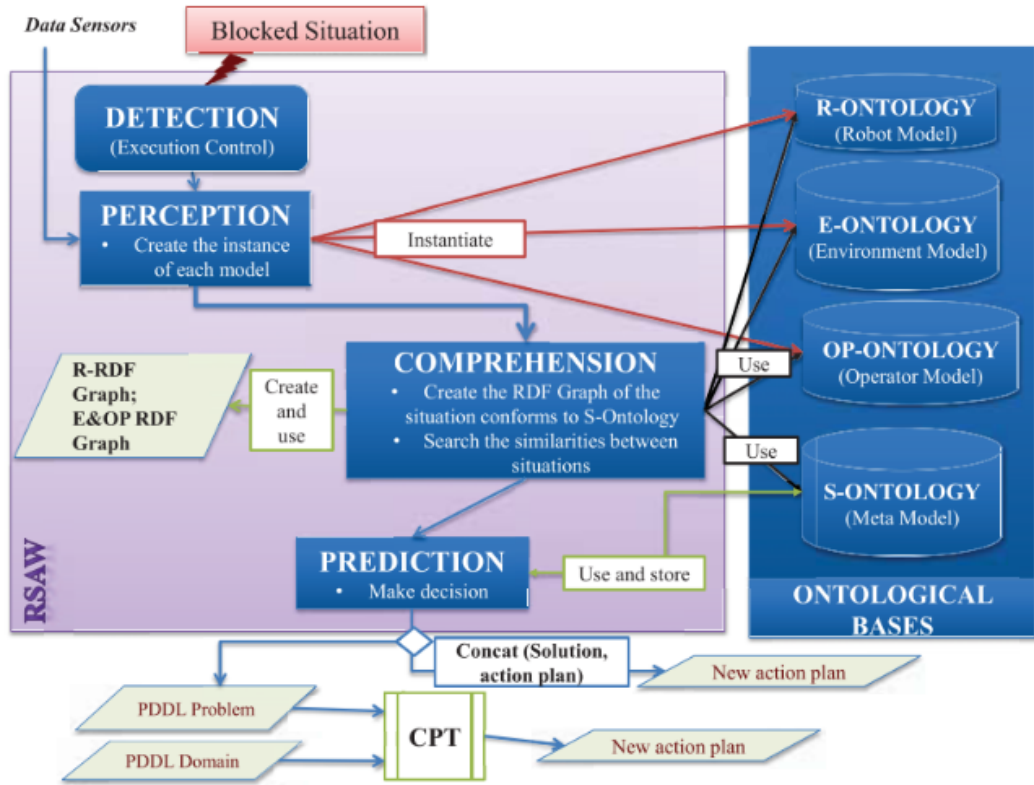
Dinamik karar verme süreci, önceki kararların sonuçları, eylemsizlik veya zamanla spontane olarak meydana gelen değişiklikler sonucunda zaman içinde evrilen koşullarda birden fazla, tekrarlayan veya ardışık seçimler yapmayı içerir (Edwards, 1962). Bir karar durumuyla karşılaşıldığında, insanların bellekten benzer SDU'ları (benzer durumlarla olan SDU'lar) geri çağırdığı düşünülmektedir. Tipik bir durumda, eylemin beklenen faydası, bellekten geri çağrılan benzer örneklerin faydalarının birleştirilmesiyle hesaplanır (Blending). Ancak, atipik durumlarda, insanlar sezgisel yöntemlere başvururlar. Bir kararın beklenen faydasının değerlendirilmesi Yargılama adımıyla yapılır. Alternatif eylemler sıralı olarak değerlendirilir ve her değerlendirmeden sonra daha fazla alternatifin değerlendirilip değerlendirilmeyeceğine karar verilir (Gereklilik). Bu gereklilik, karar vericinin tercihlerine veya dışsal faktörlere bağlı olarak subjektif olarak belirlenebilir. En yüksek faydaya sahip alternatif seçilir (Seçim) ve uygulanır (Yürütme), böylece ortam değiştirilir ve bellekte hangi SDU'nun uygulandığı kaydedilir. Bir karar verildikten sonra, kararın sonucu, karar örneklerinin fayda değerini geri bildirim olarak değiştirmek için kullanılır (Geri Bildirim).

Durumsal farkındalık, insan-robot etkileşimlerinde operatör müdahalesini azaltmak ve robotların engellenmiş durumları aşmasını sağlamak için önemli bir kavramdır (Scharfe-Scherf vd., 2022; Cossitt vd., 2022). Bu alandaki çalışmalar, dikkat ve çalışma belleği gibi bireysel ve çevresel faktörlerin, operatörlerin çevreden bilgi edinme ve yorumlama süreçlerini nasıl sınırladığını araştırmaktadır. Zihinsel modeller ve hedefe yönelik davranışlar ise bu sınırlamaları aşmak için önemli mekanizmalar olarak kabul edilmektedir.

Durumsal farkındalık ve dinamik karar verme modeli, robotikte henüz geniş çapta ele alınmamıştır. Robotik çalışmalar, algılama, mekansal algı, sensör füzyonu, durum tahmini ve SLAM gibi alt bölümler üzerine odaklanmıştır. Ancak, durumsal farkındalığın geliştirilme süreci şu ana kadar yalnızca insanlar için tanımlanmış, robotlar için özgün bir tanım yapılmamıştır.

Bu bağlamda, RSAW (Robot Durum Farkındalığı) sistemi geliştirilmiştir (Ben Ghezala vd., 2014). RSAW, robotların operatör müdahalesi olmadan

engellenmiş durumları anlamasını ve aşmasını hedefler. RSAW, durumsal farkındalık gibi bir eylem gerçekleştirmeden önce zorunlu bir aşama olarak kendini sunmaz; yalnızca robot engellendiğinde devreye girer. Ayrıca, RSAW, işin operatör ve makine arasında bölünmesini yeniden düzenleyerek operatör müdahalesini azaltmayı amaçlar. Sistemin, bilgi temsili ve engellenmiş durumları aşmak için bir süreç tanımlaması, robotların operasyonel verimliliğini artırmayı hedefler.

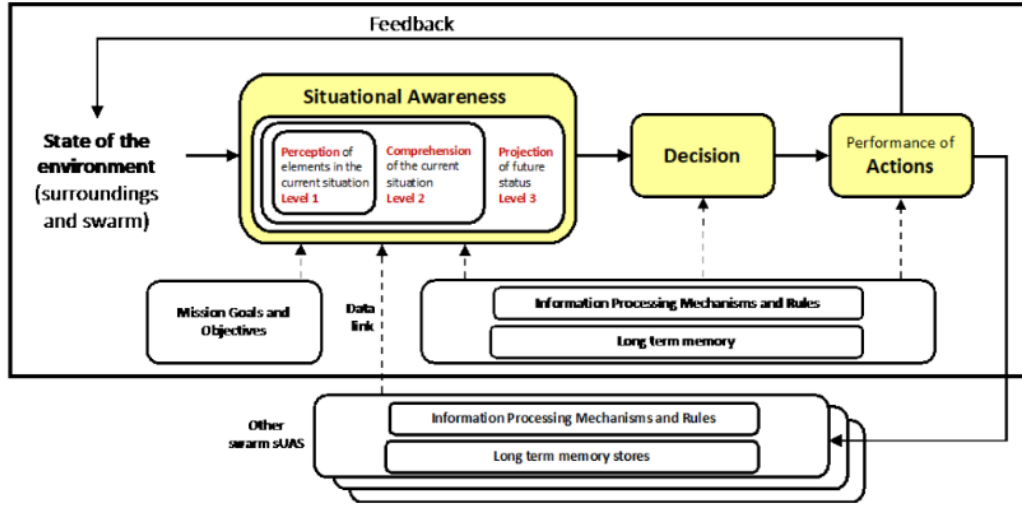


Şekil 2.3. Robot Durum Farkındalığı Modeli (Ben Ghezala vd., 2014)

RSAW (Robot Durum Farkındalığı) sistemi, engellenmiş durumları tespit etmek ve çözmek için geliştirilmiştir (Ben Ghezala vd., 2014). Sistem, dört aşamalı bir süreç içerir: (1) Tespit (Detection): Robot bir anomali tespit eder; (2) Algılama (Perception): İlgili ontolojilerde bilgi oluşturur; (3) Anlama (Comprehension): Durum grafikleri oluşturup analiz ederek mevcut durumu anlar; (4) Tahmin (Prediction): Uygun bir karar alarak kendini uyumlar. Şekil 2.3, bu sürecin görsel temsiline dair ayrıntıları sunmaktadır.

Sürü İHA'ları (İnsansız Hava Araçları) için uyarlanmış dinamik karar verme modeli, bilişsel ve işbirlikçi unsurları merkeze alır (Haag ve Huschbeck). Bu model, sürü içindeki bireylerin çevrenin bir modelini oluşturmasını ve bilgi paylaşımını sağlayarak, grup içindeki etkileşimlerin daha iyi çarpışmadan kaçınma kararları ve

etkili navigasyon rotaları belirlemesine katkıda bulunur. Bireysel İHA'lar, görevler sırasında edinilen deneyimlerle bilgi değiş tokuşunda bulunarak, yeni İHA'ların bu deneyimlerden öğrenmesine olanak tanır. Grup içindeki etkileşimler, çevre hakkında daha iyi ve daha ayrıntılı bilgi edinmeye yol açar ve böylece daha iyi çarpışmadan kaçınma kararları alınabilir ve gerekli navigasyon performansını destekleyen rotalar belirlenebilir.

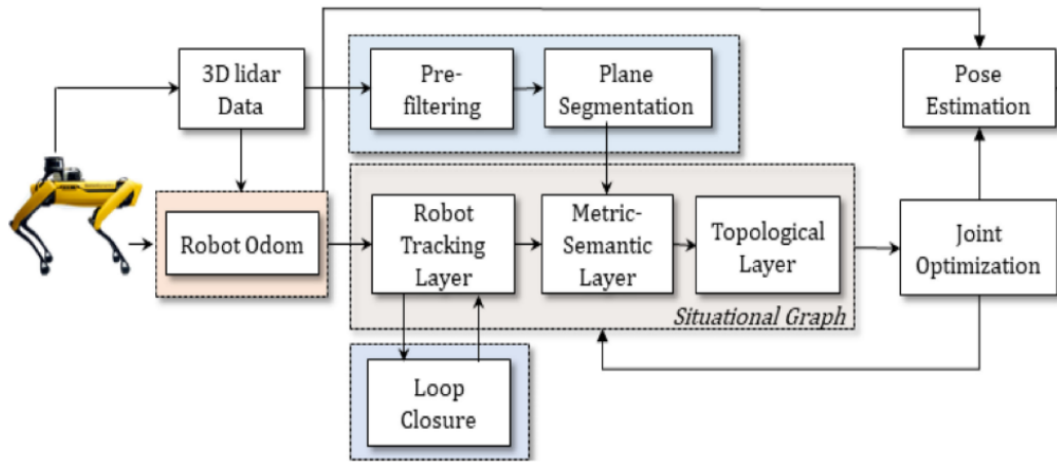


Şekil 2.4. Bilişsel ve İşbirlikçi Süreç (Haag ve Huschbeck)

Şekil 2.4, sürülerin bilişsel ve işbirlikçi navigasyonu için önerilen yöntemin temel blok diyagramını göstermektedir. Bu blok diyagramı, Endsley'in dinamik karar modeline benzer bir yapı sergiler. Endsley'in modelinde, insan faktörleri ve durumsal farkındalık üzerine odaklanılmıştır, ancak bu kavramlar, sürü üyeleri ve sürü bütünü için uyarlanmış durum farkındalık modellerine kolayca çevrilmiştir. Sürü üyeleri, görev hedeflerine ulaşırken ve gerekli navigasyon performansını sağlarken çevrelerine yönelik farkındalıklarını artırmaktadır.

Otonom robotlarda, etraflarındaki durumu anlamak, çevrelerinin doğru görev odaklı haritalarını tahmin etmek ve kendilerini bu haritalara yerleştirmek, önemli bir işlem basamağıdır. Semantik SLAM, bir robotun çevresindeki geometrik varlıklara semantik kavramlar atmasıdır. Son yıllarda yapılan çalışmalar, çevrenin temel unsurlarını zengin semantik açıklamalarla tanımlayarak, robotların durum farkındalığını ve görev planlamasını iyileştirmiştir (Cadena vd., 2016). Semantik haritalamada farklı yaklaşımlarla semantik kavramların sayısında ve türlerinde, ve bu kavramların çevrenin farklı kısımlarıyla nasıl ilişkilendirildiği farklılık gösterir. Yüksek seviyeli görev tanımlamaları, kalabalık, tekrarlayan veya dinamik

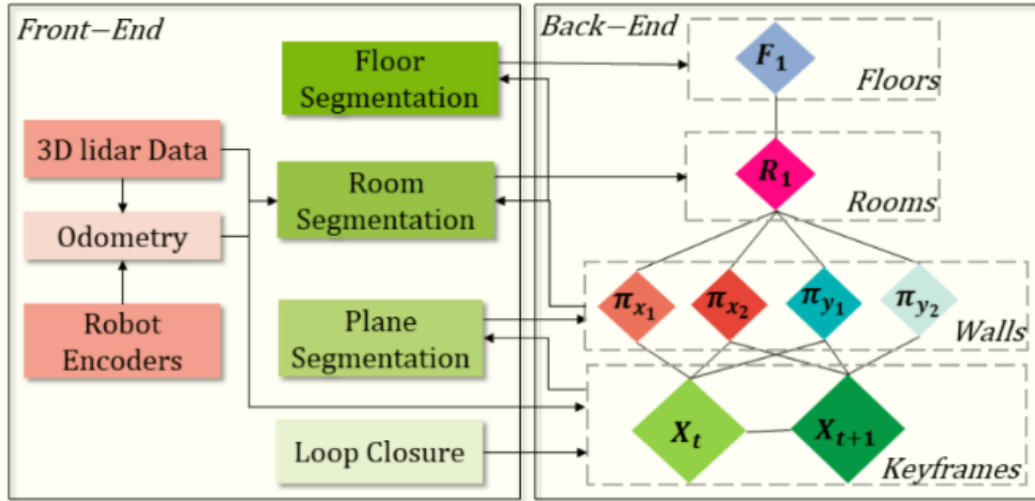
ortamlarda yerleştirme ve haritalama performansı için unsurları anlamlandırmak ve harita ile ilişkilendirmek gerekir. Ancak, bazı semantik SLAM yöntemleri çevredeki kilit semantik unsurları kullanmasına rağmen, bu unsurlar arasındaki ilişkileri modellemekte yetersiz kalmaktadır (Bowman vd., 2017; Chen vd., 2019). Diğer yaklaşımlar, sahne grafiklerini kullanarak çevrenin ve semantik unsurlarının yapısal ve topolojik kısıtlamalarla birlikte temsilini sağlamaktadır (Armeni vd., 2019; Wu vd., 2021; Bavle vd., 2022). Ancak bu tür sahne grafiklerini, robot pozlarını ve sahne grafiğini aynı anda optimize eden SLAM yöntemleriyle bağlama noktasında hala bir boşluk bulunmaktadır ve sahne grafiklerinin, navigasyon gibi karmaşık otonom görevleri ele almada yetersizdir. Bu boşluğu doldurmak için, Bavle vd. (2023a) robotik durum farkındalığını artıran üç katmanlı bir model sunmuştur. Algılama katmanı, çevresel algı için multimodal sensörler içerir. Anlama katmanı, robotun durum farkındalığını iyileştiren ve çevresel unsurları modelleyen yöntemleri içerir. Projeksiyon katmanı, robotun gelecekteki durumunu tahmin etmek için tahmin modelleri ekler. Bu model, robotların çevresel durumları anlamasını ve dinamik görevleri etkin bir şekilde yerine getirmesini sağlar.



Şekil 2.5. S-Graphs (Bavle vd., 2022)

İlk olarak Bavle vd. (2022) çalışmasında sunduğu S-Graphs Şekil 2.5 adındaki modelde lazer taramaları ve odometri verisini kullanarak yol takibi, metrik sematik çıkarım ve topolojik çıkarımlarla ortamın oda koridor gibi çıkarımlarını yaparak birirlerine göre ilişkilendirmiştir. Model sahnenin mevcut anda algılanan yönlerini, anlayışını, önceki kayıtlarla entegrasyonunu veya muhtemelen standartlaştırılmış bir ontolojiden dış kaynakları ve varlıkların modelleri aracılığıyla geleceği tahmin etmeyi vurgulayan bir bilgi grafiği olarak tanımlanır. Yazar daha

sonra ön uç ve arka uç arasında katmanlı çıkarımların olduğu S-Graphs+ modelini Şekil 2.6 sunmuştur (Bavle vd., 2023b). SLAM algoritmalarına durumsal farkındalığın kazanıldırılması açısından önemli bir çalışmadır. Fakat kazandırılan farkındalık duvar oda ve koridor ile sınırlıdır. Topolojik katman, belirli bir zamanda haritalanan düzlemlerin belirli bir nesneye veya yapısal bileşene ait olup olmadığını değerlendirir ve geometrilerini daha fazla sınırlar. Böylelikle algılanan duvar yapılarından oda ve koridor çıkarımları yaparak topolojik olarak birbirine bağlar.



Şekil 2.6. S-Graphs+ (Bavle vd., 2023b)

SLAM çalışmalarında durumsal farkındalık kavramı genellikle çevrenin algılanıp anlamlandırılması olarak dar bir kapsamda ele alınmaktadır. Ancak, durumsal farkındalık, anlamlandırmanın ötesinde, yakın geleceğe dair kestirimler de sağlayan daha üst seviye bir kavramdır. Akai vd. (2018) ve Tong vd. (2017) gibi çalışmalarda hata tespiti üzerine odaklanılmıştır, fakat bu çalışmalar genellikle durumsal farkındalık kavramını kapsamlı bir şekilde ele almamaktadır.

SLAM algoritmalarının mevcut durumları sadece algılama ve anlamlandırma düzeyinde ele alması, çevresel zorlukları yönetme ve tepki üretme kapasitesini sınırlamaktadır. Otonom robotların etkili ve güvenilir bir şekilde çalışabilmesi için SLAM algoritmalarının çevresel zorlukları ele alacak ve adaptif tepkiler üretecek şekilde tasarlanması gerekmektedir. Bu bağlamda, Dinamik Karar Modelleri (Dynamic Decision-Making (DDM)), SLAM sistemlerinin bütüncül olarak tasarlanmasına ve daha sağlam bir otonomi sağlamasına yardımcı olabilecek önemli bir yaklaşım olarak öne çıkmaktadır.

2.3. SLAM Veri setleri

SLAM algoritmalarının somut deneylerinin maliyetli donanımlar ve karmaşık prosedürler içerir. Sensör entegrasyonları, kalibrasyonlar ve harita çıkarım süreçleri gibi işlemler, yüksek maliyetler ve uzun süreli çalışmalar gerektirir. Genel olarak, her algoritma taslağının saha deneyi şeklinde test edilmesi neredeyse imkansızdır. Bu nedenle, SLAM algoritmalarının gelişimi ve değerlendirilmesi için veri setleri kritik bir araç haline gelmiştir (Liu vd., 2021).

SLAM algoritmalarının gelişimiyle birlikte, veri setleri de teknolojik ilerlemelere paralel olarak farklı sensör modellerini içermektedir. Sensör kurulumuna göre, SLAM yöntemleri Lidar tabanlı, Görsel tabanlı, Görüntü-Lidar füzyonu ve RGB-D tabanlı olarak sınıflandırılabilir. Lidar tabanlı yöntemler (2D (Hess vd., 2016) ve 3D (Zhang vd., 2014)), yüksek doğruluklu menzil ölçümleri sunarak yaygın olarak kullanılmakta ve olgun bir çözüm olarak kabul edilmektedir. Ancak, Lidar'ın yüksek maliyeti bu yöntemlerin sınırlamalarından biridir.

Görsel sensörler ve bilgisayar donanımındaki gelişmelerle birlikte, görüntü tabanlı SLAM yöntemleri büyük ilgi görmüştür (Caruso vd., 2015; Mur-Artal vd., 2015; Vidal vd., 2018). Bu yöntemler, monoküler, stereo, balıkgözü ve olay tabanlı olmak üzere çeşitli alt sınıflara ayrılmakta ve farklı uygulama senaryoları için uygun çözümler sunmaktadır. Her ne kadar Lidar tabanlı yöntemler kadar sağlam veya doğru olmasalar da (Shan ve Englot, 2018), görüntü yöntemlerinin büyük bir potansiyele sahip olduğu ve sürekli olarak geliştiği kanıtlanmıştır (Campos vd., 2021).

Son yıllarda, görüntü ve Lidar modalitelerinin avantajlarını birleştiren Görüntü-Lidar füzyonu önemli bir konu haline gelmiştir (Zhang ve Singh, 2015). RGB-D kameralar, renkli görüntülerin yanı sıra derinlik kanalı da toplar ve bu da 3D algılamayı daha kolay hale getirir. Microsoft Kinect (Zhang, 2012) ile tanıtılan RGB-D sensörleri, düşük maliyetleri ve tamamlayıcı doğaları nedeniyle SLAM araştırmalarında yaygın olarak benimsenmiştir (Newcombe vd., 2011; Whelan vd., 2015).

Veri setleri, sensör türleri, toplama platformları ve ortam özellikleri gibi önemli niteliklere göre farklı SLAM yaklaşımlarının ihtiyaçlarını karşılamaktadır. Literatürde en çok atıf alan veri setleri Tablo 2.1 ve Tablo 2.2'de özetlenmiştir.

En popüler veri setleri genellikle araba üzerinde toplanmış, maliyeti yüksek

ve yoğun veri içeren sensörlerle dış ortamlarda elde edilmiştir (Pandey vd., 2011; Geiger vd., 2013; Maddern vd., 2017). Açık ve kapalı alan ortamları, sensörlerin ölçüm yetenekleri ve karşılaşılan zorluklar açısından farklı karakteristiklere sahiptir. Açık alanlarda mevsimsel şartlar, aydınlatma koşulları ve dinamik engeller öne çıkarken; kapalı alanlarda ise aydınlatma şartları, sahne özellikleri ve çok katlı yapılar gibi zorluklar ön plandadır.

Mevcut veri setleri incelendiğinde, Lidar tabanlı veri setlerinin çoğunlukla (Yin ve Berger, 2018; Liu vd., 2021) açık alan ortamlarında toplandığı ve kapalı alanlar için yeterli çeşitliliğe sahip füzyon veri setlerinin bulunmadığı görülmüştür. Açık alan veri setlerinin çoğu (Ros vd., 2016; Burri vd., 2016; Caruso vd., 2015) görsel sistemlere dayanmakta ve Lidar-Görsel füzyonuna olanak tanımamaktadır. Son yıllarda SLAM veri setlerinde 3D Lidar ve görüntüleme sistemlerine yönelik bir eğilim gözlemlenmiştir (Zhang vd., 2022; Yin vd., 2022). 3D Lidar sistemleri, yüksek doğruluk ve ayrıntılı veri sağlama yetenekleri ile SLAM algoritmalarının performansını artırmakta ve farklı ortam koşullarında güvenilir sonuçlar elde edilmesine olanak tanımaktadır. Ancak, 2D Lidar sistemlerinin daha düşük maliyeti, daha az veri boyutu ve düşük enerji tüketimi gibi avantajları bulunmaktadır. 2D Lidar ve Görsel füzyonuna dair yapılacak çalışmalar, bu avantajları değerlendirerek önemli bir araştırma alanı olarak devam etmektedir.

SLAM teknolojisi ile paralel olarak gelişen veri setleri, son yıllarda gerçek dünya senaryoları, zorlu koşullar ve çeşitli sensör kombinasyonlarına odaklanmaktadır (Zhang vd., 2022). Kapalı alanlarda, servis, hizmet, güvenlik ve arama-kurtarma görevlerinde kullanılacak robotlar için kapalı alan veri setleri üzerinde yoğunlaşmıştır. Bu veri setleri, robotların karşılaşılabileceği farklı sahne özelliklerini (Tong vd., 2013; Chebroly vd., 2017a), değişken ışık koşullarını (Smith vd., 2009; Zhu vd., 2018; McCormac vd., 2017), dinamik engelleri (Blanco-Claraco vd., 2014b; Jeong vd., 2019) ve hızlı manevraları (Sturm vd., 2012; McCormac vd., 2017; Burri vd., 2016) içermekte ve SLAM algoritmalarını sınyarak robotların gerçek dünyayı daha iyi anlamasını ve doğru kararlar almasını sağlamaktadır.

Kapalı alanlarda toplanmış 2D Lidar ve Görsel füzyona uygun veri setleri şunları içerir: Ceriani vd. (2009); Pronobis ve Caputo (2009); Carlevaris-Bianco vd. (2016); Ruiz-Sarmiento vd. (2017); Shi vd. (2020). Bu veri setlerinden NCLT, OpenLORIS Lidar sensörü olarak 3D ve 2D Lidarı birlikte barındırmaktadır ve sensör konfigürasyonları daha çok 3D Lidar yönündedir. Veri setlerinden yalnızca Rawseeds,

Robot@Home ve COLD 2D Lidar ve visual sistemlere dayanmaktadır. Rawseed veri setinde temel gerçekliğinin SLAM algoritmalarını test etmek için önemine vurgu yapan ekip kapalı alan ve açık alan ortamlarda dinamik engellerin bulunduğu veri setini toplulukla paylaşmıştır. COLD mobil robotlar üzerinde temelde vizyon tabanlı topolojik yerleştirme ve semantik çalışmalar için oluşturulmuştur. 2D Lidar taramalarında sağlayan veri setinde dinamik engeller, aydınlatma şartlarındaki değişimlerin yanı sıra zaman içerisinde değişen nesne ve mobilyalar gibi zorluklar bulunmaktadır. NCLT veri seti kapalı alan ve açık alan ortamlarda toplanmış kapsamlı bir şekilde zorlu şartları bünyesinde bulunduran bir veri setidir. Fakat bu zorlu şartların çoğunluğu açık alan ortamlara aittir ve temel olarak sistem 3D Lidar ve çok yönlü görüntülere dayanmaktadır. Robot@Home veri seti dört RGB-D kamera ve bir 2D Lidar ile toplanan veri seti farklı aydınlatma şartları ve zayıf yüzey özelliklerini bulunduran temelde semantik slam için oluşturulmuş oda etiketlerini bulunduran bir veri setidir. Openloris veri setinde, Lifelong SLAM tanımlaması yapılmıştır ve servis robotlarının zaman içerisinde sahne, nesne ve aydınlatma şartlarında değişikliklerle karşılaşabileceğine vurgu yaparak farklı zamanlarda toplanmış farklı zorlukları barındıran veri setini toplum ile paylaşmıştır. Önceki veri setlerine kıyasla, bu veri seti, uzun süreli robot navigasyonu için özelleştirilmiş dinamik ve günlük değişen senaryolarla gerçek dünya şartlarını yansıtmaya çalışmıştır. Veri seti, hizmet robotları için oldukça tipik uygulama senaryoları olan ofis, koridor, ev, kafe ve market olmak üzere 5 sahnede toplanmıştır.

Her bir veri seti, SLAM algoritmalarının gerçek dünya koşullarındaki performansını değerlendirmek için çeşitli zorluklar ve özellikler sunmaktadır. Özellikle 2D Lidar ve Görsel füzyonu üzerine yapılan çalışmalar, bu alanın önemini vurgulamakta ve gelecekteki araştırmalar için önemli bir temel oluşturmaktadır.

Çalışmamızla eş zamanlı olarak yapılmış NUFR-M3F (Kaveti vd., 2023) ve ExhibitionDataset (Yuan vd., 2023) çalışmalarında, kapalı alan ortamlar için farklı zorluklar ele alınmıştır. Her iki veri setinde temel olarak 3D Lidar ve kamera sensörlerine dayanmaktadır. NUFR-M3F veri setinde çok katlı simetrik yapılar sahip bir bina içerisinde asansörler ile geçişlerin olduğu veri toplanmıştır, fakat merdivenler dahil edilmemiştir. ExhibitionDataset’de ise karmaşık ve sürekli değişen yapılarına vurgu yapılan bir sergi alanından veri toplanmıştır.

Füzyon çalışmalarında, 2016 öncesi 2D Lidar kullanımının yaygın olduğu, ancak daha sonrasında kullanım oranının azaldığı gözlemlenmiştir. 2D Lidar

sensörlerinin kapalı ortamlarda avantaj sağladığı ve varlığını devam ettirdiği görülmektedir, ancak güncel zorlukları barındıran veri setlerinin sınırlı olduğu anlaşılmaktadır.

Bu çalışmada, çeşitli zorlukları ve ortamları kapsayan, visual-2D Lidar füzyonuna uygun bir veri seti sunuyoruz. Kapalı alan ortamlarda maliyet, ağırlık ve güç tüketimi açısından avantajlı olan 2D Lidar ve RGB-D kameraları temel sensörler olarak kullanarak, ayrıca ışık sensörü ile ortamın aydınlatma durumu hakkında bilgi sağlamayı hedefledik. Kapalı alan ortamlarda visual ve 2D Lidar sensörlerini barındıran datasetleri Tablo 2.3 de karşılaştırılmıştır.

Tablolar içerisinde zorlu şartlar , Liu vd. (2021) referans alınarak, zorluklar harflerle ifade edilmiştir. “M” , hareket modeli tarafından oluşturulan, hızlı hareket, saf döndürme, hareket bulanıklığı ve Lidar bozulmaları “F”, zayıf, tekrarlayan ve özellik ve yapının müdahalesi gibi sahne türünden kaynaklanan zorluklar “S”, sahnede karışıklığa veya belirsizliğe neden olan karmaşık veya karışık yapılar “D”, dinamik objeler “T”, değişken, kötü ve değişen aydınlatma “W”, çeşitli ve olumsuz hava koşulları “T”, farklı zaman aralıkları “SE”, çoklu mevsimler “C”, karmaşık geometriyi “L”, büyük ölçekli dizileri (her birinde 1k'dan fazla çerçeve) Referansta isimlendirilen zorluklara ek olarak “TR” yansıtıcı ve saydam yüzeyler “VM” bina içi katlar arası hareket, asansör ve merdivenler “LL” uzun süreli çalışmaları ve zaman içerisindeki ortam değişiklikleri için LifeLong (Biber ve Duckett, 2009) teriminden esinlenerek kullanılmıştır.

Çizelge 2.1. Popüler Veri Setleri Özellikleri

Dataset Name	Year	Platform	Sensors			Collection Scene	Challenging
			İmu	Camera	Lidar		
New College (Smith vd., 2009)	2009	Wheeled Robot	Y	2 x gray (stereo) 5 x color (panora)	2 x Sick-2D	Outdoor-Campus	I
Rawseeds (Ceriani vd., 2009)	2009	Wheeled Robot	Y	3 x gray (stereo) 1 x color 1 x omni-color	2 x Hokuyo-2D 2 x Sick-2D	Indoor-Building Outdoor-Campus	D
DARPA (Huang vd., 2010)	2010	Car	N/A	4 x color (stereo) 1 x WA-color	1 2 x Sick-2D 1 x Velodyne-64	Outdoor-Urban	D, I
Marulan (Peynot vd., 2010)	2010	UGV	N/A	1 x color 1 x IR-thermal	4 x Sick-2D	Outdoor-Field	W, T
Ford Campus (Pandey vd., 2011)	2011	Car	Y	6 x color (omni)	1 x Velodyne-64 2 x Riegl-2D	Outdoor-Campus	-
Canadian Planetary (Tong vd., 2013)	2013	UGV	Y	3 x color (stereo)	1 x Sick-2D	Indoor/Outdoor	F,S
Institut Pascal (Korrapati vd., 2013)	2013	Car	Y	4 x gray (2 for stereo) 1 x omni-color 1 x fisheye-color 1 x webcam-color	2 x Sick-2D	Outdoor	-

Çizelge 2.2. Popüler Veri Setleri Özellikleri

Dataset Name	Year	Platform	Sensors			Collection Scene	Challenging
			İmu	Camera	Lidar		
Malaga Urban (Blanco-Claraco vd., 2014a)	2014	Car	Y	2 x color (stereo)	2 x Sick-2D 3 x Hokuyo-2D	Outdoor-Urban	D
NCLT (Carlevaris-Bianco vd., 2016)	2016	Wheeled Robot	Y	6 x color (omni)	1 x Velodyne-32 1 x Hokuyo-2D	Indoor-Building Outdoor-Campus	D, T, I, W, SE
Sugar Beets (Chebrolu vd., 2017b)	2017	Field Robot		1 x RGB/NIR 1 x RGB-D	2 x Velodyne-16 1 x Nippon-2D	Outdoor-Ag.	F, S, T
Oxford RobotCar (Maddern vd., 2017)	2017	Car	N/A	3 x color (stereo) 3 x fisheye-color	2 x Sick-2D 1 x Sick-4	Outdoor-Urban	D, I, T, W, SE
Robot@Home (Ruiz-Sarmiento vd., 2017)	2017	Wheeled Robot	N/A	4 x RGB-D	1 x Hokuyo-2D	Indoor-Rooms	I, F
Complex Urban (Jeong vd., 2019)	2019	Car	Y	2 x color (stereo)	2 x Velodyne-16 2 x Sick-2D	Outdoor-Urban, campus	D
Oxford Radar (Barnes vd., 2020)	2020	Car	N/A	3 x color (stereo) 3 x fisheye-color	1 x Radar 2 x Sick-2D 2 x Velodyne-32	Outdoor-Urban	I, D, T, W

Çizelge 2.3. Kapalı Alan Veri Setlerinin Konya-Veri Seti ile Karşılaştırma Tablosu

Dataset Name	Year	Collection Scene	Sensors		Confidence	Challenging	
			İmu	Camera			Lidar
Rawseeds (Ceriani vd., 2009)	2009	Indoor-Building Outdoor-Campus	Y	3 x gray (stereo) 1 x color 1 x omni-color	2 x Hokuyo-2D 2 x Sick-2D	X	D
COLD (Pronobis ve Caputo, 2009)	2009	Indoor-Laboratories	Y	2 x color 1 x omnicolor	1 x Sick-2D	X	I,D, T, LL
NCLT (Carlevaris-Bianco vd., 2016)	2016	Indoor-Building Outdoor-Campus	Y	6 x omnicolor	1 x Velodyne-32 1 x Hokuyo-2D	X	D, T, I, W,SE
Robot@Home (Ruiz-Sarmiento vd., 2017)	2017	Indoor-Rooms	N/A	4 x RGB-D	1 x Hokuyo-2D	X	I, F
OpenLORIS (Shi vd., 2020)	2020	Indoor-Multiple	Y	1 x RGB-D 2 x fisheye-RGB (stereo)	1 x Hokuyo-2D 1 x Robosense-16	X	I, T, D, TR, LL
Konya-Dataset	2023	Indoor-Multiple	Y	1 x RGB-D 1 x infrared (stereo)	1 x Slamtec-2D 1 x Leddar Vu8	✓	D, I, T, F, S, M, C, TR, MS, LL

3. EŞ ZAMANLI HARİTALAMA VE KONUMLAMA (SLAM)

Otonom robotların konumlama ve haritalama problemlerinin çözümüne yönelik ilk çalışmalarda, farklı teorilere dayanan iki yaklaşım öne çıkmıştır: İlki, haritanın başlangıçta bilindiği ve sisteme yüklendiği konumlama çalışmalarıdır; ikincisi ise, konumun bilindiği bir ortamda haritanın çıkarılmasına odaklanan çalışmalardır. Ancak, bu çalışmaların ötesinde, haritalama ve konumlandırma işlemlerinin eş zamanlı olarak yürütüldüğü SLAM yöntemi, robotik alandaki gelişmelerin önemli bir basamağı haline gelmiştir.

Olasılıksal robotik, "Belirsizliğini modelleyerek buna uygun hareket edebilen bir robot, bu yeteneğe sahip olmayan bir robota göre daha üstündür" görüşüyle temellenmiş bir yaklaşımdır ve SLAM bu alanın en önemli gelişmelerinden biri olarak kabul edilmektedir (Thrun, 2002). Olasılıksal yaklaşımlar, sensör sınırlamaları, sensör gürültüsü ve çevre dinamikleri gibi zorluklarla karşı karşıya kaldıklarında daha dayanıklıdır. Bu tür yaklaşımlar, karmaşık ve yapılandırılmamış ortamlara uyum sağlama konusunda daha başarılı olup, belirsizliğin önemli rol oynadığı dinamik ortamlarda daha iyi ölçeklenebilirlik sunar.

Aslında, olasılıksal algoritmalar, mobil robotların karşılaştığı bazı zorlu tahmin problemlerine bilinen tek etkili çözümü sunmaktadır. Örneğin, bir robotun kendi lokalizasyon hatasından kurtulması gereken kaçırılma problemi ya da GPS gibi global bir konumlandırma cihazı olmaksızın çok büyük ortamların doğru haritalarını oluşturma problemi, bu tür algoritmaların etkili bir şekilde ele aldığı durumlardır. Ek olarak, olasılıksal algoritmalar, birçok klasik planlama algoritmasının gerektirdiği yüksek model doğruluğu gereksinimlerini hafifleterek, programcayı doğru modeller oluşturma yükünden büyük ölçüde kurtarır.

Olasılıksal bir bakış açısıyla, robotik öğrenme, uzun vadeli bir tahmin problemidir. Bu nedenle, olasılıksal algoritmalar, robotik öğrenmenin birçok türü için sağlam bir metodoloji sunar. Son olarak, bu algoritmalar, gerçek dünyada algılama ve eylem içeren hemen her problem için geniş bir uygulama alanına sahiptir.

3.1. SLAM Algoritmaları Teori ve Yaklaşımları

SLAM problemleri, robotun çevre haritasına erişiminin olmadığı ve kendi konumlarını bilmediği durumlarda ortaya çıkar. SLAM, robotun çevresinin haritasını çıkarırken, aynı anda bu haritaya göre kendini konumlandırır. SLAM, robotik problemler arasında çözülmesi en zor olanlardan biridir. Konumlandırma probleminden daha zordur, çünkü harita bilinmemekte ve yol boyunca tahmin edilmek zorundadır. Aynı şekilde, bilinen konumlarla haritalama probleminden de daha zordur, çünkü konumlar bilinmemekte ve hareket boyunca sürekli olarak tahmin edilmelidir.

Robotun durumu, uygulama türüne bağlı olarak farklı şekillerde tanımlanabilir: 2D pozisyon ve yönlendirme, 6D pozisyon, hız, ivme vb. x_t , t zamanındaki araç durumu tahminini ve m çevre haritasını temsil eder. Bu değişkenleri tahmin edebilmek için, $t - 1$ ve t arasındaki hareketin bir tahminini temsil eden ve genellikle tekerlek kodlayıcıları veya yer değiştirme hakkında bilgi verebilecek diğer sensörlerden gelen kontrol girdileri u_t avantajından yararlanmak mümkündür.

Robota sadece $z_{1:t}$ ölçümleri ve $u_{1:t}$ kontrollerinin verildiği bir sistemde, olasılıksal olarak, SLAM probleminin iki ana formu vardır ve her ikisi de pratik açıdan önem taşır. Birincisi, çevrimiçi SLAM problemi olarak bilinir. Bu problem, robotun anlık konumu ve haritası üzerindeki dağılımı tahmin etmeyi içerir:

$$p(x_t, m \mid z_{1:t}, u_{1:t}) \quad (3.1)$$

Burada x_t zaman t 'deki konum, m harita, $z_{1:t}$ ve $u_{1:t}$ ise sırasıyla ölçümleri ve kontrolleri temsil eder. Bu problem çevrimiçi SLAM problemi olarak adlandırılır çünkü sadece zaman t 'deki sürekli değişkenlerin tahminini içerir. Çoğu çevrimiçi SLAM algoritması artımsaldır: Geçmiş ölçümler ve kontroller bir kez işlendikten sonra atılır.

İkinci SLAM problemi Tam SLAM problemi olarak adlandırılır. Tam SLAM probleminde, sadece mevcut konum x_t değil, tüm yol $x_{1:t}$ boyunca izlenen konumlar ve harita üzerindeki son dağılım hesaplanmaya çalışılır:

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \quad (3.2)$$

Çevrimiçi ve tam (bazı kaynaklarda global olarak da geçmektedir) SLAM problemleri arasındaki formülasyon farkı, kullanılacak algoritmaların türlerine etki eder. Özellikle çevrimiçi SLAM problemi, Tam SLAM probleminden geçmiş konumları entegre etmenin sonucudur:

$$p(x_t, m \mid z_{1:t}, u_{1:t}) = \int \int \cdots \int p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) dx_1 dx_2 \cdots dx_{t-1} \quad (3.3)$$

SLAM probleminin ikinci önemli özelliği, tahmin probleminin doğası ile ilgilidir. SLAM problemleri, sürekli ve ayrık bileşenlere sahiptir. Sürekli tahmin problemi, nesnelerin haritadaki konumları ve robotun pozisyonel değişkenleri ile ilgilidir. Nesneler, özellik tabanlı bir gösterimde yer işaretleri veya telemetreler tarafından tespit edilen nesne yamaları olabilir. Ayrık doğa ise yazışmalarla ilgilidir: Bir nesne tespit edildiğinde, SLAM algoritması bu nesnenin daha önce tespit edilen nesnelerle aynı olup olmadığı konusunda mantık yürütmelidir. Bu akıl yürütme genellikle ayrık bir süreçtir: Nesne ya daha önce tespit edilenle aynıdır ya da değildir.

Tahmin teknikleri iki ana kategoriye ayrılabilir: Filtre tabanlı yaklaşımlar ve optimizasyon tabanlı yöntemler (Bresson vd., 2017). Filtre tabanlı yaklaşımlar, verilerin artan bir şekilde işlenmesine dayanır ve sistemin mevcut durumunu sürekli olarak günceller. Öte yandan, optimizasyon tabanlı yöntemler, tüm verileri bir arada işleyerek en olası çözümü bulmaya çalışır.

Filtre Tabanlı SLAM: Filtre tabanlı SLAM yöntemleri, Bayes filtrelemesine dayanan ve iki aşamalı yinelemeli süreçler olarak çalışan yöntemlerdir. İlk aşamada, aracın ve haritanın durumları, bir evrim modeli ve kontrol girdileri u_t kullanılarak tahmin edilir. İkinci aşamada, sensör verilerinden gelen mevcut gözlem z_t , harita ile eşleştirilir ve bu sayede önceki tahminler düzeltilir. Gözlem ile harita arasındaki ilişkiyi kuran modele "gözlem modeli" denir. Bu iki adım, sensör verilerini yinelemeli olarak entegre ederek aracın konumunu ve haritayı tahmin eder.

Genişletilmiş Kalman Filtresi (EKF): Filtre tabanlı yöntemlerin ilk dalı, Kalman Filtresi (KF) (Kalman, 1960) türevlerini içerir. KF'ler, temel olarak doğrusal sistemleri ele almak için tasarlanmıştır ve yakınsama özellikleri iyidir, ancak SLAM için nadiren kullanılırlar. Genişletilmiş Kalman Filtresi (EKF) (Kalman ve Bucy, 1961), doğrusal olmayan filtrelemede yaygın olarak kullanılır. EKF, doğrusal olmayan modeller için bir doğrusallaştırma adımı ekler. Bu adım, birinci dereceden Taylor serisi genişlemesiyle mevcut tahminin etrafında

gerçekleştirilir. EKF'nin optimal çözümü, doğrusallaştırmanın durum vektörünün gerçek değeri etrafında yapılması durumunda elde edilir. Pratikte ise, bu değer tahmin edilendir ve dolayısıyla mevcut değildir. Bu durum, tutarlılık sorunlarına yol açabilir; gerçek değer, tahmin edilen belirsizliğin dışında kalabilir (Julier ve Uhlmann, 2001; Bar-Shalom vd., 2004). Ancak, tahminler genellikle gerçeğe yeterince yakındır.

Harita boyutunun sürekli büyümesi, EKF'nin büyük ölçekli SLAM'ı destekleyememesine neden olur çünkü güncelleme süresi, durum vektörünün boyutuna bağlı olarak ikinci dereceden artar. Bu sorunu aşmak için "alt haritalar" (submaps) kavramı geliştirilmiştir. Bir harita çok büyük hale geldiğinde (çeşitli kriterler bu kararı vermek için kullanılır), yeni bir boş harita eski haritanın yerini alır. Üst düzey bir harita, alt haritalar arasındaki bağlantıları takip ederek bilgiyi kaybetmemeyi sağlar.

Kokusuz Kalman Filtresi (UKF): EKF'nin yüksek derecede doğrusal olmayan sistemlerdeki zayıflıklarını telafi etmek amacıyla (Julier ve Uhlmann, 1997), Jacobian hesaplamalarını önleyen Kokusuz Kalman Filtresi (UKF) geliştirilmiştir. Bu yöntemde, sigma noktaları adı verilen ve bir olasılık fonksiyonu yardımıyla beklenen değer etrafında analiz edilen örnekler kullanılır. Bu sigma noktaları daha sonra doğrusal olmayan fonksiyona geçirilir ve tahmin yeniden hesaplanır. Bu yöntemin ana dezavantajı, yüksek hesaplama süresidir.

Bilgi Filtresi (IF): Kalman Filtresi'nin bir başka varyantı olan Bilgi Filtresi (IF), Kalman Filtresi'nin ters formudur (Maybeck, 1982). IF'nin özelliği, bilgi matrisini kovaryans matrisinin tersi olarak tanımlamasıdır. Ana avantajlarından biri, güncelleme adımının eklemeli hale gelmesi ve gözlemlerin hangi sırayla entegre edildiğine bağlı olmamasıdır (Liu ve Thrun, 2003). Bilgi matrisini, veriler arasındaki zayıf bağlantıları kırarak daha seyrek hale getirmek de mümkündür (Thrun vd., 2004), bu da neredeyse sabit zamanlı bir güncellemeyi garanti eder (Eustice vd., 2005). IF, tek araçlı SLAM'da EKF kadar popüler değildir çünkü her ölçümün ters formuna dönüştürülmesi gereklidir, bu da maliyetli olabilir. Ancak, IF çoklu araçlı SLAM'da daha yaygın kullanılır.

Parçacık Filtresi: Parçacık filtresi, durumu olasılık yoğunluğuna göre bir dizi parçacıkla örnekler. Her filtrede olduğu gibi, her parçacığın yer değiştirme tahmini yapılır ve gözleme bağlı olarak bir güncelleme gerçekleştirilir. Güncelleme aşamasında, parçacıklar, ölçümlere göre olasılıklarına göre ağırlıklandırılır. En olası

parçacıklar tutulur, diğerleri elenir ve yenileri üretilir (Dellaert vd., 1999).

Optimizasyon Tabanlı SLAM: Optimizasyon tabanlı SLAM yaklaşımları, genellikle filtre tabanlı SLAM’da olduğu gibi iki alt sistemden oluşur. İlk alt sistem, yeni gözlemler ile harita arasında bir eşleşme bularak sensör verilerine dayalı problemin kısıtlarını tanımlar. İkinci alt sistem, aracın duruşunu (ve geçmiş duruşlarını) ve haritayı, bu kısıtları göz önünde bulundurarak hesaplar veya iyileştirir, böylece tutarlı bir bütün elde edilir. Filtrelerde olduğu gibi, bu yöntemler de iki ana dalda incelenebilir: Demet Ayarı (Bundle Adjustment) ve Grafik SLAM.

Demet Ayarı (Bundle Adjustment): Demet ayarı, 3D yapı ve kamera parametrelerini (duruş) birlikte optimize eden bir görüntüleme tekniğidir. Ana fikir, genellikle Levenberg-Marquardt algoritması kullanılarak bir amaç fonksiyonunu optimize etmektir. Bu amaç fonksiyonu, en iyi kamera ve işaretçi konumlarını veren yeniden projeksiyon hatasını (görüntüdeki gözlemler ile geçmiş özelliklerin yeniden projeksiyonu arasındaki mesafe) en aza indirir. Ancak, temel demet ayarı algoritması, tüm değişkenleri bir kerede optimize etmeyi içerdiği için hesaplama açısından ağır olabilir.

Grafik SLAM: Bayesian SLAM’ın grafiksel temsili, optimizasyon yöntemleriyle çözüm için uygundur. Grafiksel temsil temelinde, işaretçiler ve araç duruşları arasındaki ilişkileri tanımlayan bir matris kolayca oluşturulabilir ve bir optimizasyon çerçevesinde kullanılabilir. Birçok yerelleştirme problemi, bir grafik temsili kullanılarak modellenilebilir ve bu formu izleyen bir maliyet fonksiyonunun minimumunu bularak çözülebilir. Alt yöntemlerde, tüm trajektoryayı optimize ederek veya gerçek zamanlı kullanım için küçük yer değiştirme artışlarıyla çalışabilir. Filtreleme yaklaşımlarında olduğu gibi, minimizasyonun başarısı, başlangıç tahminine bağlıdır.

3.2. Robot İşletim Sistemi (ROS) ve SLAM

Robot İşletim Sistemi (ROS (Robot Operating System)), robotik sistemlerin geliştirilmesi, simülasyonu ve uygulamaya konulmasında geniş çapta kullanılan açık kaynaklı bir çerçevedir. ROS, robotik yazılım geliştirmeyi kolaylaştıran araçlar, kütüphaneler ve konfigürasyon dosyalarından oluşur. Modüler yapısı sayesinde, ROS farklı robotik projelerde tekrar kullanılabilir bileşenler sunarak SLAM gibi karmaşık

algoritmaların geliştirilmesi ve uygulanmasını önemli ölçüde kolaylaştırır.

ROS'un modüler yapısı, SLAM algoritmalarının farklı sensörler, veri işleme yöntemleri ve haritalama teknikleriyle kolayca entegre edilmesine olanak tanır. Bu, araştırmacılar ve mühendisler için SLAM algoritmalarını çeşitli robotik platformlarda test etmeyi ve uygulamayı kolaylaştırır. ROS, robotik yazılım geliştirmeyi kolaylaştıran birçok kütüphane ve araç sunar. Bu kütüphaneler arasında hareket planlama, görüntü işleme ve sensör veri entegrasyonu gibi yaygın olarak kullanılanlar bulunmaktadır. Gazebo gibi simülasyon platformları, robotik sistemlerin gerçek dünya ortamlarında nasıl performans göstereceğini test etmek için kullanılabilir. Bu, SLAM algoritmalarının güvenli ve etkili bir şekilde geliştirilmesine ve test edilmesine olanak tanır.

Geniş ve aktif bir topluluğa sahip olan ROS, kullanıcıların karşılaştıkları sorunları çözmelerine yardımcı olacak dokümantasyon, forumlar ve eğitim materyalleri sunar. ROS üzerinde geliştirilen çalışmaların GitHub platformu ile dağıtımını kolaylıkla yapılabilir.

GitHub, açık kaynaklı projelerin barındırılması, paylaşılması ve işbirliği yapılması için yaygın olarak kullanılan bir platformdur. GitHub, kullanıcıların projelere kolayca erişmesini ve indirmesini sağlar. Bu, ROS paketlerinin geniş bir kullanıcı kitlesi tarafından kullanılmasını sağlar. SLAM paketleri gibi karmaşık projeler, GitHub üzerinden kolayca paylaşılabilir ve güncellenebilir. Projelerin farklı sürümlerinin yönetilmesini ve takip edilmesini sağlar, bu da ROS paketlerinin geliştirilmesi sırasında yapılan değişikliklerin izlenmesini ve gerektiğinde eski sürümlere dönülmesini kolaylaştırır. Topluluk üyeleri, ROS paketlerine katkıda bulunabilir, hata düzeltmeleri ve yeni özellikler ekleyebilir. Bu, SLAM algoritmalarının ve diğer ROS paketlerinin sürekli olarak geliştirilmesini sağlar. GitHub, projeler için ayrıntılı dokümantasyon oluşturmayı ve paylaşmayı kolaylaştırır. Kullanıcılar, ROS paketlerinin nasıl kullanılacağına dair rehberler ve örnekler bulabilir. Ayrıca, GitHub Issues ve Pull Requests gibi araçlar, kullanıcıların karşılaştıkları sorunları bildirmelerini ve çözüm önerileri sunmalarını sağlar.

ROS üzerinde geliştirilen SLAM paketleri, GitHub üzerinden dağıtılarak geniş bir kullanıcı kitlesine ulaşmıştır. Örneğin, ORB-SLAM, RTAB-Map, Hector SLAM gibi popüler SLAM algoritmaları ROS paketleri olarak mevcuttur ve GitHub üzerinde barındırılmaktadır. Bu paketler, kullanıcıların robotik projelerinde yüksek doğrulukta haritalama ve konumlandırma yapmalarını sağlar. GitHub'ın sağladığı

avantajlar sayesinde, bu paketler kolayca erişilebilir, güncellenebilir ve topluluk tarafından geliştirilebilir. ROS ve GitHub kombinasyonu, robotik yazılım geliştirme sürecini önemli ölçüde basitleştirir ve hızlandırır. Çalışmamızda kullandığımız SLAM paketleriyle ilgili bilgiler aşağıda paylaşılmıştır.

Gmapping: Gmapping, robotik ve otonom sistemlerde yaygın olarak kullanılan önemli bir SLAM algoritmasıdır. Bu algoritma, olasılıksal bir yaklaşım kullanarak genellikle 2D Lidar sensörleriyle çalışır. Gmapping, parçacık filtreleri kullanarak çalışır; bu filtreler, robotun olası konumlarını temsil eden bir dizi parçacık (partikül) kullanır. Özellikle Rao-Blackwellized Partikül Filtreleri (RBPF) tekniğini kullanarak standart parçacık filtrelerine göre daha verimli bir şekilde çalışır ve haritalama doğruluğunu artırır.

Gmapping, düşük hesaplama maliyeti ve yüksek doğruluk oranıyla öne çıkar. Çevresel belirsizlikleri ve sensör hatalarını yöneterek güvenilir haritalar oluşturur. ROS ile entegrasyonu, Gmapping'in geniş bir kullanıcı kitlesi tarafından benimsenmesini sağlamıştır. Bu algoritma, özellikle hizmet robotları ve endüstriyel robotlar gibi birçok alanda yaygın olarak kullanılmaktadır.

Hector Mapping: Hector Mapping, özellikle yüksek frekanslı Lidar verileriyle çalışan robotlar için tasarlanmış bir SLAM algoritmasıdır. Bu algoritma hızlı, hassas konumlandırma ile haritalama yetenekleri sunar. Hector Mapping, enkoder veya IMU gibi hareket sensörlerine ihtiyaç duymadan, yalnızca Lidar verilerine dayanarak çalışır, bu da onu düşük maliyetli ve basit robotik sistemler için ideal kılar.

Hector Mapping, tarama eşleştirme (scan matching) tekniklerini kullanarak robotun hareket ettikçe aldığı Lidar taramalarını birbirine hizalar ve bu hizalamayı kullanarak robotun hareketini ve çevrenin haritasını belirler. Bu süreç, yüksek çözünürlüklü haritalar oluşturmak için sürekli olarak tekrarlanır ve robotun konumunu yüksek doğrulukla belirler. Ayrıca, çevresel değişikliklere hızlı bir şekilde uyum sağlar ve gerçek zamanlı uygulamalar için idealdir.

Cartographer: Cartographer, Google tarafından geliştirilen ve robotik sistemlerde kullanılan gelişmiş bir SLAM algoritmasıdır. Hem 2D hem de 3D haritalama yeteneklerine sahip olup, Lidar, IMU ve enkoder verilerini kullanarak robotların çevresini yüksek doğrulukla haritalar ve kendi konumunu belirler. Cartographer, gerçek zamanlı performansı ve çoklu sensör entegrasyonu ile öne çıkar, bu da onu çeşitli robotik uygulamalar için ideal hale getirir.

Bu algoritma, sensör verilerini kullanarak kısa vadeli hareket tahminleri yapar ve bu tahminleri kullanarak sürekli bir haritalama süreci gerçekleştirir. Lidar sensöründen gelen taramalar, robotun çevresel algısını oluşturur. Bu taramalar, çevredeki nesnelerin mesafelerini ölçer ve harita oluşturma sürecinde kullanılır. IMU, robotun hızlanma ve dönme hareketlerini ölçer ve bu veriler, hareket tahmini ve konum belirlemede kullanılarak Lidar verileriyle birleştirilir. Cartographer, Graph-based SLAM yöntemlerini kullanarak robotun konumu ve harita bilgilerini sürekli olarak optimize eder ve bu sayede global bir harita ve kesin konum bilgisi elde eder.

Cartographer, çoklu sensör entegrasyonu ve gerçek zamanlı performansı ile öne çıkar. Yüksek doğruluklu haritalama ve konumlandırma yetenekleri, onu karmaşık ve dinamik ortamlarda kullanıma uygun hale getirir. Otonom araçlar, mobil robotlar, dronlar ve bina içi haritalama gibi çeşitli uygulamalarda yaygın olarak kullanılır. Cartographer, SLAM problemini çözmek için gelişmiş ve çok yönlü bir yöntem sunar. ROS ile entegrasyonu ve güçlü optimizasyon teknikleri sayesinde, otonom sistemlerde geniş bir kullanım alanı bulmuş ve robotik araştırmaların yanı sıra endüstriyel uygulamalarda da yaygın olarak benimsenmiştir.

Görsel algılamının hızla gelişmesiyle birlikte araştırmacılar, kameraların Lidar'a kıyasla daha zengin bilgiler sağlayabildiğini keşfetmiştir. Lidar-SLAM, ortamın nokta bulutu verilerini oluştururken, Görsel-SLAM gri veya renkli görüntüler üretir ve bu, çevrenin daha ayrıntılı ve anlamlı bir temsilini sağlar. Ayrıca, görüntü sensörlerinin düşük maliyeti, basit yapıları ve çeşitli kurulum yöntemleri, hem iç hem de dış ortamlarda kullanılabilir olmalarıyla öne çıkar. Bu avantajları sunan önemli algoritmalarından biri RTAB-Map, bir diğeri ise ORB-SLAM3'tür.

RTAB-Map, özellikle RGB-D kameralarla uyumlu çalışarak hem görsel hem de geometrik bilgileri entegre ederken; ORB-SLAM3, çok çeşitli kamera türleriyle çalışabilen ve IMU verilerini de entegre ederek daha stabil ve doğru konumlandırma sağlayan gelişmiş bir algoritmadır. Bu algoritmalar, hem akademik araştırmalarda hem de endüstriyel uygulamalarda yaygın olarak kullanılmaktadır.

RTAB-Map: RTAB-Map (Real-Time Appearance-Based Mapping), mobil robotlar ve otonom sistemler için kullanılan bir SLAM algoritmasıdır. Özellikle RGB-D kameralar, stereo kameralar, Lidar ve diğer sensörlerle uyumlu olan RTAB-Map, hem görsel hem de geometrik bilgileri kullanarak yüksek doğrulukta haritalama ve konumlandırma sağlar. Algoritma, görsel özellikleri ve mekansal

verileri entegre ederek robotun çevresini üç boyutlu olarak haritalar ve robotun konumunu belirler.

RTAB-Map, temel olarak görünüm tabanlı bir yaklaşım kullanır ve görsel özellikleri tanıyarak çevresel haritalar oluşturur. Algoritma, görsel verileri kullanarak robotun bulunduğu konumu belirler ve bu verileri birleştirerek 3D haritalar oluşturur. Hem yerel hem de global optimizasyon teknikleri kullanarak haritaları sürekli olarak günceller ve robotun konumunu optimize eder. Görsel özellik tanıma, kapatma döngüsü algılama ve Graph-based SLAM yöntemlerini kullanarak, robotun konumunu ve çevresel haritayı yüksek doğrulukla belirler.

Görsel Özellik Tanıma: RGB-D veya stereo kameralar kullanarak çevredeki görsel özellikleri tanıyarak ve bu özellikleri haritalama sürecinde kullanır. Özellik tanıma algoritmaları, görüntülerdeki belirli noktaları veya nesnelere tespit eder.

Kapatma Döngüsü Algılama (Loop Closure Detection): Robotun aynı alanı tekrar ziyaret ettiğinde, bu alanı tanıyarak ve haritayı düzeltmek için kapatma döngüsü algılama teknikleri kullanır. Bu, haritanın doğruluğunu artırır ve tutarlılığını sağlar.

Graph-based SLAM: Görsel ve geometrik verileri kullanarak bir grafik yapısı oluşturur ve bu yapıyı optimize ederek robotun konumunu ve çevresel haritayı sürekli günceller.

Sensör Entegrasyonu: RTAB-Map, Lidar, IMU ve enkoder gibi çeşitli sensörlerden gelen verileri entegre ederek daha kesin ve güvenilir haritalar oluşturur. Bu sensör verileri, robotun hareketini ve çevresel değişiklikleri daha doğru bir şekilde algılamasını sağlar.

ORB-SLAM3: ORB-SLAM3, ORB-SLAM2'nin üzerine eklenen önemli iyileştirmeler ve genişletmelerle daha güçlü ve çok yönlü bir SLAM algoritmasıdır. Her iki algoritma da görsel tabanlı SLAM sistemleridir ve ORB (Oriented FAST and Rotated BRIEF) özelliklerini kullanarak çevresel haritalama ve robotun konumlandırmasını gerçekleştirir. Ancak, ORB-SLAM3, hem özelliklerde hem de yeteneklerde bir dizi önemli farkla geliştirilmiştir.

ORB-SLAM2 ve ORB-SLAM3 Arasındaki Farklar: Multi-Map Desteği: ORB-SLAM2 tek bir harita ile çalışırken, ORB-SLAM3 çoklu harita desteği sunar; bu da aynı anda birden fazla ortamın haritalanmasına ve yönetilmesine olanak tanır.

IMU Entegrasyonu: ORB-SLAM2 IMU verilerini kullanmazken, ORB-SLAM3 IMU verilerini entegre ederek, özellikle hızlı hareketlerde daha stabil ve doğru bir konumlandırma sağlar.

Kamera Desteđi: ORB-SLAM2 mono, stereo ve RGB-D kameralarla alıřırken, ORB-SLAM3 bu kameraların yanı sıra, Uuř Suresi (Time-of-Flight), Olay (Event) Kameraları ve multi-camera sistemlerini de destekler. Bu, daha karmařık ve geniř aılı grř gerektiren uygulamalarda kullanım esnekliđi sađlar.

ORB-SLAM3, iyileřtirilmiř algoritmalar ve optimizasyon teknikleri ile daha yksek dođruluk, daha verimli veri iřleme ve optimizasyon sunar, bu da zellikle byk ve karmařık ortamlarda daha gvenilir sonular elde edilmesini sađlar (Campos vd., 2021). Bu sistem, byk, kk, i ve dıř ortamlarda gerek zamanlı olarak alıřabilir. Őiddetli hareket karmařasına karřı dayanıklıdır, geniř kapsamlı dnglerin kapatılmasına ve yeniden konumlandırılmasına izin verir ve tam otomatik bařlatma ierir.

4. SLAM ALGORİTMALARI İÇİN DURUMSAL FARKINDALIK TEMELLİ DİNAMİK KARAR VERME MODELİ

Gelecekte hizmet robotlarının, dinamik ortamlarda uzun süre boyunca otonom olarak çalışabilmesi gerekecektir. Bu, robotların insanlarla birlikte yaşamalarını ve insanların çevrelerinde yaptığı değişikliklere uyum sağlamalarını zorunlu kılar. Bu tür bir çalışma ortamında, SLAM algoritmalarının değişen koşullara adapte olabilmesi ve karşılaştığı zorluklarla başa çıkabilmesi büyük bir önem taşımaktadır. Dinamik ve sürekli değişen bir dünyada hayatta kalabilmek için, bir robotun sürekli öğrenme yeteneğine sahip olması kaçınılmazdır. Örneğin, dinamik ortamlarda çevreler farklı hızlarda değişebilir ve bu değişiklikler kalıcı veya geçici olabilir; bu nedenle, robotun eski durumu hatırlayabilmesi veya bu değişiklikleri dahil edebilmesi arzu edilir. Sürekli öğrenme, hem yeni kalıplara uyum sağlamayı hem de eski kalıpları korumayı gerektirir. Bunun yanı sıra, çevredeki değişikliklerin robotun algılama ve işleyişine nasıl etki ettiği ve olumsuz durumların nasıl üstesinden gelinebileceği gibi kavramlar da gündeme gelmektedir.

Bu kavramlardan hareketle, uzun süreli SLAM için eksiksiz bir sistemin, SLAM algoritmalarına entegre edilecek adaptif yaklaşımlar ile yalnızca haritalama yapmanın ötesine geçerek, çeşitli durumlara uygun tepkiler geliştirmesi gerektiği açıktır. Bu çalışmada, bütüncül bir yaklaşımla, SLAM algoritmalarının yalnızca bir algoritma değil, aynı zamanda bir teknoloji olarak değerlendirilmesi gerektiğine inanıyor ve bu doğrultuda bir model öneriyoruz. Önerilen model, insanların dinamik karar verme süreçlerinden esinlenilerek geliştirilmiştir.

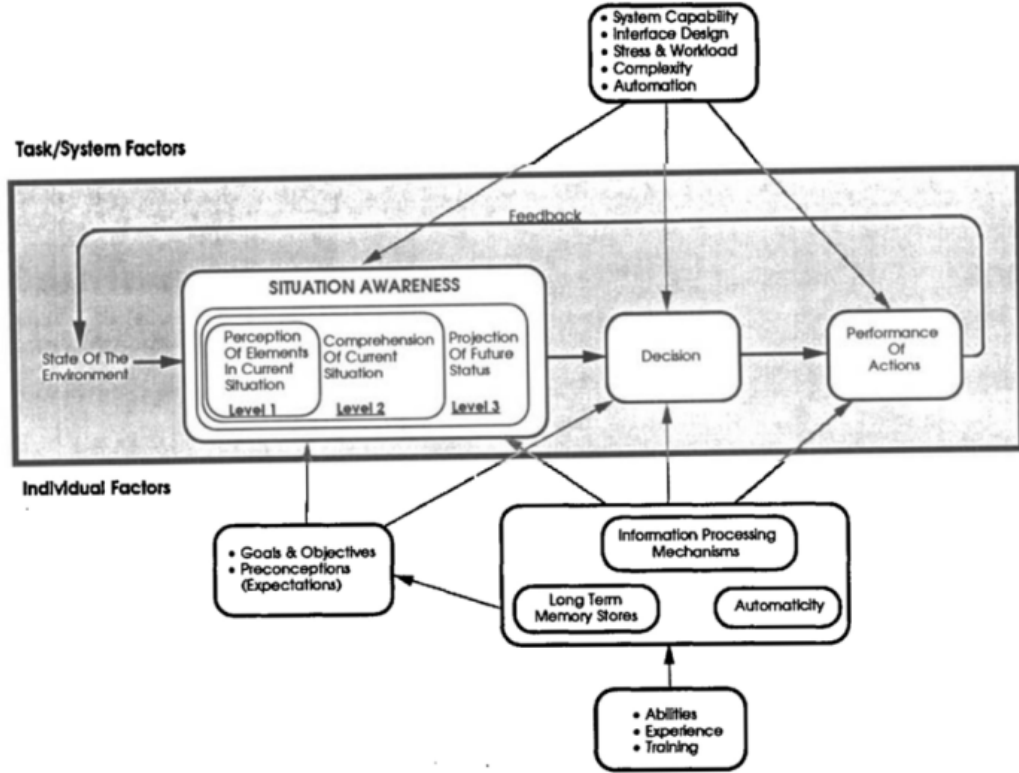
4.1. Karar Verme Modelleri

Temel olarak insanların karar alma süreçlerini modelleyen Endsley (1995)'in sunduğu Dinamik Karar Modeli ve Gonzalez vd. (2003)'in sunduğu Örnek Tabanlı Öğrenme Modeli baz alınarak SLAM teknolojisi için yeni bir model oluşturulmuştur.

Bu modelin daha iyi anlaşılabilmesi için öncelikle bu iki model ve yaklaşımlarının detaylı bir şekilde incelenmesi gerekmektedir. Ardından, SLAM

algoritmalarına uyarlanmış olan yeni modelimiz açıklanacaktır.

4.1.1. Dinamik Karar Verme Modeli (DDM)



Şekil 4.1. Dinamik Karar Verme Modeli (Endsley, 1995)

Endsley'nin Dinamik Karar Modeli (DDM), üç ana bileşenden oluşur: Bilgisayar ve sistem tarafı, göreve yönelik Durumsal Farkındalık (SA), karar ve eylem süreçleri ile hedef ve beklentilere sahip, farklı yetenek ve becerilere sahip operatör tarafı. Bilgisayar sistemlerinin, gerekli bilgileri sağlama ve temel insan bilgi işleme yetenekleriyle uyumlu olması beklenir. SA'nın kazanılması, bireysel yetenekler doğrultusunda gerçekleşirken, karar verme mekanizmaları operatörün sorumluluğundadır. Stres, iş yükü, yorgunluk ve deneyim gibi faktörlerin, insan performansını ve dolayısıyla sistemin işleyişini etkilediği açıktır. Otonominin gelişmesiyle birlikte, bu modelin robotik sistemlere uygulanması fikri, operatörün yaptığı görevlerin sisteme aktarılması üzerine kuruludur.

Model incelendiğinde, DDM'nin ana çerçevesini görevler belirler ve diğer alt modüller, bu görevler doğrultusunda çalışır. Diğer önemli bir bileşen ise Durumsal Farkındalıktır. SA, bir sistem veya görevle ilgili karar verebilmek için gereklidir ve

bu nedenle, karar alma süreçleriyle entegre bir şekilde çalışır; temelde bir kişinin hedefleriyle bağlantılıdır. Hedefler, dinamik ortamlarda karar verme süreçlerinin temelini oluşturur. Karar verme sürecinde, Cohen (1993) "Yukarıdan Aşağıya" karar verme süreci tanımında belirtildiği gibi, bir kişinin hedefleri ve planları, SA'nın geliştirilmesinde çevrenin hangi yönlerinin dikkate alınacağını belirler. Karar verici, çevresini algılayarak anladığı temelinde planlarını ve hedeflerini gerçekleştirmek için eylemleri seçer. Bu "Yukarıdan Aşağıya" süreçle eş zamanlı olarak, "Aşağıdan Yukarıya" işleme de gerçekleşir. Çevrede tanınan desenler, aktif hedeflere yanıt verebilmek için yeni planların gerekliliğini veya farklı hedeflerin etkinleştirilmesi gerektiğini gösterebilir. Bu şekilde, bir kişinin mevcut hedefleri ve planları, çevredeki olaylara duyarlı olacak şekilde değişebilir. Bu süreç, dinamik bir ortamda etkili bir şekilde işlem yapmayı sağlar.

Bilgisayar sistemleri veya operatörler (ajanlar), en azından ilgi alanları içinde, hedeflerinin tatminine yol açacak şekilde çevreye müdahale etme eğilimindedir. Bu amaçla, ajanlar çevre hakkında bilgi toplamalı (genellikle birçok farklı kaynaktan), toplanan bilgiye dayanarak kararlar almalı, bu kararlara göre harekete geçmeli, eylemlerine tepki olarak çevreden geri bildirim toplamalı ve gelecekte daha iyi kararlar almak için bilgilerini güncellemelidir (öğrenmelidir). Hem insanlar hem de bilgisayarlar, bilgi miktarı ve karar verme süreçlerinin karmaşıklığı nedeniyle bilgi işleme zorluklarıyla karşılaşabilirler.

SA, karar verme sürecinin temel girdisi olmasının yanı sıra, karar verme sürecini de etkileyebilir. Bir kişinin durumu karakterize etme biçimi, sorunu çözmek için seçilen karar verme sürecini belirlemektedir. Bu nedenle, modelde ilk olarak SA bileşeninin incelenmesi gerekmektedir.

4.1.1.1. Durumsal Farkındalık

"Durumsal Farkındalık (SA), belirli bir zaman ve mekân hacmi içinde çevredeki unsurların algılanması, bu unsurların anlamlarının kavranması ve durumlarının yakın gelecekteki projeksiyonunu içerir."

Durumun Algılanması: SA'nın ilk adımı, çevredeki ilgili unsurların durumunu, niteliklerini ve dinamiklerini algılamaktır. Endsley'in örnekleriyle açıklamak gerekirse; bir pilot, uçaklar, dağlar veya uyarı ışıkları gibi unsurları ve

bunların ilgili özelliklerini (örneğin, renk, boyut, hız, konum) algılar. Taktik bir komutan, belirli bir alandaki düşman ve dost kuvvetlerin konumu, türü, sayısı, yetenekleri ve dinamikleri hakkında doğru verilere ihtiyaç duyar. Esnek bir üretim sistemi operatörü, makinelerin, parçaların, akışların ve birikimlerin durumu hakkında bilgi toplar. Bir otomobil sürücüsü ise diğer araçların ve engellerin konumlarını, bu unsurların dinamiklerini ve kendi aracının durumunu bilmelidir.

Durumun kavranması: Durumun kavranması, sadece mevcut unsurların farkında olmaktan öte, bu unsurların anlamını ve ilgili operatör hedefleri doğrultusunda yorumlama yeteneğini içerir. Örneğin, askeri bir pilot veya taktik komutan, belirli bir coğrafi konumda birbirine yakın üç düşman uçağının görünmesinin, onların hedefleri hakkında ne anlama geldiğini anlamalıdır. Bir enerji santrali operatörü, çeşitli sistem değişkenlerine ilişkin verileri birleştirerek, sistem bileşenlerinin performansını, beklenen değerlerden sapmaları ve herhangi bir sapma okumasının belirli yerini tespit etmelidir. Yeni başlayan bir operatör, daha deneyimli karar vericilerle aynı seviyede SA'ya ulaşabilir, ancak verileri hedeflerle bütünleştirme konusunda yetersiz kalabilir.

Durumun Geleceğe Yansıtılması: Çevredeki unsurların gelecekteki eylemlerini, en azından çok yakın bir zamanda öngörebilme yeteneği, SA'nın üçüncü ve en yüksek seviyesini oluşturur. Bu, unsurların durumu ve dinamikleri bilgisi ile durumun anlaşılması yoluyla sağlanır. Örneğin, bir savaş pilotu, bir tehdit uçağının şu anda saldırgan olduğunu ve belirli bir konumda bulunduğunu bilerek, uçağın belirli bir şekilde saldırma olasılığını kestirir. Bu bilgi, hedeflere ulaşmak için en uygun hareket tarzını belirlemek için gerekli zamanı sağlar. Benzer şekilde, bir hava trafik kontrolörü, hangi pistlerin boş olduğunu ve çarpışma olasılığını belirlemek için çeşitli trafik modellerine ilişkin bilgileri bir araya getirmelidir. Bir otomobil sürücüsü, gelecekteki olası çarpışmaları tespit etmeli ve bir üretim sistemi operatörü de gelecekteki darboğazları ve kullanılmayan makineleri etkili bir şekilde planlamak için tahminde bulunmalıdır. Bu nedenle, SA, çevre hakkında bilgi algılamaktan çok daha fazlasına dayanır; bilgilerin anlamını entegre eder, operatör hedefleriyle karşılaştırır ve çevrenin gelecekteki durumlarını kestirir. SA, geniş bir uygulama alanına sahip, birçok bilişsel süreçle ilişkili geniş bir yapıdır.

Durumsal Farkındalığın özellikleri:

SA, belirli bir zamanda robotun çevresi hakkındaki bilgisi olarak tartışılmış olsa da, doğası gereği zamanla oluşur. SA, çevrenin dinamiklerini dikkate alarak,

sadece mevcut durumu değil, aynı zamanda çevrenin geçmiş ve gelecekteki durumlarını da içerir. SA, mekansal olarak da gözlemlenebilir; birçok alan, sistem bileşenleri arasındaki mekansal ve işlevsel ilişkilerle ilgilidir. Mekansal bilgi, SA için önemli olan çevrenin hangi yönlerinin belirlenmesi için oldukça faydalıdır. Bir robotun SA'sı, görevler ve hedefler için alakalı olan çevrenin o alt kümesi hakkında bilgiyi içermelidir. Bu sınır içinde, unsurlar önem derecelerine göre alt bölümlere ayrılabilir veya problem bağlamına göre bir alaka sürekliliği varsayabilir.

Bir operatörün SA'sı, görevler ve hedeflerle ilgili çevrenin alt kümesi hakkındaki bilgileri içermelidir. Bu sınır içinde, zamansal, mekansal ve işlevsel ilişkilerine göre alt bölümlere ayrılabilir veya önem derecesine bağlı olarak bir süreklilik gösterebilir.

Algılama sürecinde dikkatin dağılımı, bir kişinin paralel olarak birden fazla öğeyi doğru bir şekilde algılama yeteneğine sınırlamalar getirir ve bu nedenle SA'nın önemli bir sınırınıdır. Dikkat, algılama ve işleme aşamalarının yanı sıra, karar verme ve tepki aşamalarında da gereklidir. Karmaşık ve dinamik ortamlarda, bilgi yüklenmesi, karmaşık karar verme ve çoklu görevlerin dikkat talepleri, bir kişinin sınırlı dikkat kapasitesini hızla aşabilir.

Karmaşık sistem operatörleri genellikle bu sınırı aşmak için bilgi örnekleme süreçleri kullanırlar. Bilgilere, uzun vadeli bellekle ilgili öncelikler ve bilgilerin değişme sıklığına göre dikkat ederler. Çalışma belleği, bir kişinin diğer algılanan bilgiler veya aktif hedefler temelinde dikkat dağılımını değiştirmesine olanak tanır. Örneğin, garip bir ses algılandığında, bir pilot motor durumu göstergesine bakma ihtiyacı hissedebilir. Bir düşman uçağına ateş etme hedefine odaklanıldığında, dikkat öncelikli olarak o hedefe yönlendirilebilir.

Dikkat çekici işaretler, dikkati çekmesinin yanı sıra, insanlar aynı zamanda hedeflerine ve amaçlarına dayalı olarak dikkati yönlendirerek SA'larının bir parçası olacak çevrenin hangi unsurlarının belirlenmesi konusunda aktif rol oynarlar. Bir pilot SA çalışmasında, sınırlı dikkat kaynaklarının, görev başarısına katkıda bulunma yeteneklerine göre çevresel unsurlara tahsis edildiği gösterilmiştir. Dikkat arzı sınırlı olduğundan, bazı unsurlara daha fazla dikkat edilmesi (bu unsurlarda iyileştirilmiş SA) diğer unsurlarda SA kaybına neden olabilir ve bu durum karmaşık ortamlarda oldukça hızlı bir şekilde gerçekleşebilir.

SA'nın hatalı olduğunu anlamak, kişinin içsel modeline dayanan beklentilerle uyuşmayan yeni veri parçalarının algılanmasıyla mümkündür.

Beklentilerle uyuşmayan algılanan veriler, yeni bir model benimseyerek, mevcut modeli gözden geçirerek veya yeni durum sınıflandırmalarına uyum sağlamak için hedeflerin ve planların değiştirilmesiyle çözülebilir. Geçmiş bilgileri içeren uzun vadeli, çok katmanlı bir durum modeli oluşturulması önemlidir. Belirli bir anlık zaman diliminde dikkate alınmayan küçük hatalar bile, zaman içinde robotun durumu ile harita tahmini arasında büyük bir sapmaya neden olabilir.

4.1.1.2. Karar Verme

Karar verme, algı, bellek ve dikkat gibi temel bilişsel süreçlere dayanarak ayrışan "üst düzey" bir bilişsel süreçtir ve tercih süreci ile benzersiz şekilde tanımlanır. Tercih, alternatifler arasında seçim yapma eylemidir; bu alternatifler aynı anda mevcut olabilir veya zamanla gelişebilir. Tercih süreci, bir tercihte bulunmadan önce gerçekleşen bilişsel süreçlerden (örneğin, algı, tanıma ve yargı) ve bir tercihte bulunduktan sonra gerçekleşen süreçlerden (örneğin, geri bildirim ve öğrenme) büyük ölçüde etkilenir. Özellikle, tercih öncesinde yargı süreci, farklı alternatiflerin değerlendirilmesini ve tercihlerinin belirlenmesini içerir.

İnsan (karar verici), bir görevi yerine getirmek için çevre ile etkileşimde bulunur ve bu görevi tekrarlanan kararlar yoluyla gerçekleştirir. Bir karar verici, çevreden bilgi algılar ve bu bilgiyi alternatifler bulmak ve yaratmak, tercihler oluşturmak ve seçenekleri değerlendirmek için dönüştürür. Bu süreç, bir tercihe yol açar ve eylem gerçekleştirildiğinde çevrede değişikliklere neden olur. Daha sonra, geri bildirim (yani, alınan eylemlerin sonuçlarının bilgisi) işlenmeli ve geçmiş kararları pekiştirmek (veya pekiştirmemek) için kullanılmalıdır (yani, geçmiş tercihlerden öğrenmek). Karar verme, kapalı çevrim bir öğrenme döngüsüdür: kararlar önceki tercihlere ve ayrıca dış olaylara ve koşullara bağlıdır.

Mevcut hedefler, karar verme süreci üzerinde ve durumsal farkındalık bileşenlerinde etkili olur. Bir hedefi gerçekleştirirken, karar verici kritik özelliklere dikkat eder. Örneğin, bir pilot yeni bir tehdit algıladığında, etkinleştirilen hedefler değişebilir ve tehdit değerlendirme modeli bu hedefle uyumlu olarak etkinleştirilebilir. Detaylı seçilen bir model, durum anlayışını, gelecekteki projeksiyonu ve karar verme sürecini yönlendirmek için kullanılabilir. Bir tehdit değerlendirme modeli, örneğin, tehditlerin ve tehdit hareketlerinin hangi

desenlerinin saldırı veya savunma faaliyetleri oluşturduğuna dair bilgiler içerebilir. Gelecekteki tehdit hareketleri, mevcut tehdit hareketleri ve bilinen taktiklere dayalı olarak modelden tahmin edilebilir. Belirli tehdit eylemlerine karşı uygun taktikler de karar verme sürecini basitleştiren scriptler şeklinde mevcut olabilir.

4.1.1.3. Bellek Yapıları

Endesley, bellek yapılarını yavaşça gelişen yapılar olarak tanımlar. Bireyler, bir sistemle ilgili bazı hedefler belirler ve bu hedefler, ulaşılmak istenen ideal durumları temsil eder. Aynı hedef seti, belirli bir sistem için sıklıkla var olabilir veya değişebilir. Bununla birlikte, bir hedef seti birden fazla sistem modeliyle ilişkilendirilebilir. Bir kişinin güncel hedef(ler)i, rekabet eden hedefler arasında en önemli olarak seçilir ve zihinsel bir modelin seçilmesini yönlendirir.

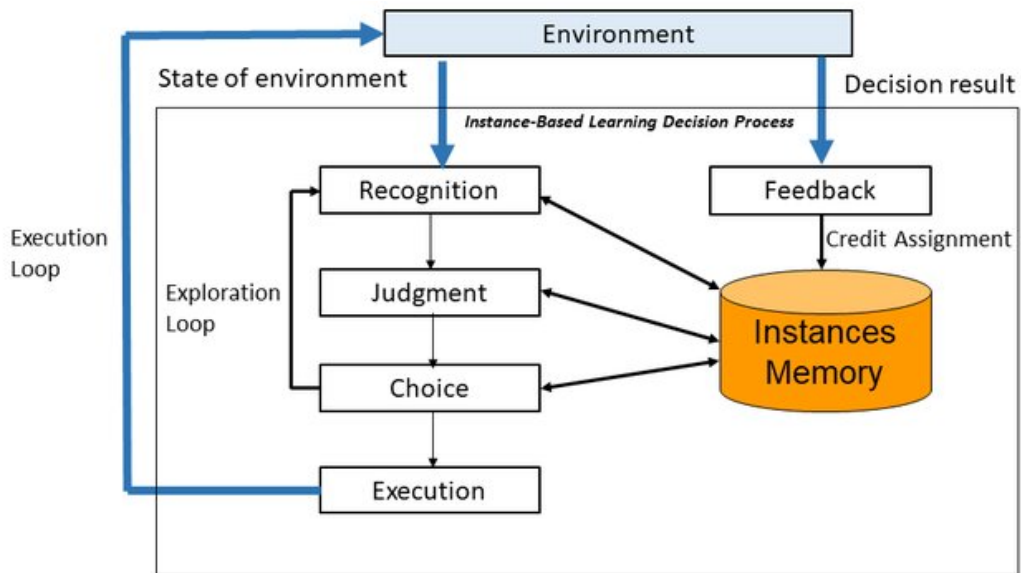
Uzun Süreli Bellek: Uzun süreli bellek, algılanan bilgiyi bilinen kategorilere veya zihinsel temsillere sınıflandırmada önemli bir rol oynar. Bu süreç, algı aşamasında neredeyse anında gerçekleşir. Kategorizasyon, entegre bilgiye dayanır ve genellikle deterministik, neredeyse optimal bir şekilde gerçekleşir. Bilginin temsillere sınıflandırılması, durumsal algılama (SA) oluşturur ve SA'nın daha yüksek seviyeleri için temel yapı taşlarını sağlar. İyi gelişmiş bellek depolarıyla, çok ince kategorizasyonlar yapılabilir. Uzmanlık seviyeleri arttıkça, insanlar çevredeki kritik ipuçlarını geliştirme bilgisine sahip görünürler ve bu, çok ince sınıflandırmalar yapmalarına olanak tanır. Algı aşamasında yapılan sınıflandırma (doğru veya yanlış, detaylı veya kaba), mevcut bilgiye bağlıdır.

Bilgi, durumların çoğu detayları şematize edildiğinde daha tutarlı ve organize hale gelir; bu, depolama, geri alma ve işleme süreçlerini kolaylaştırır. Şemalar, bilgiyi anlama için tutarlı çerçeveler sağlar, karmaşık sistem bileşenlerini, durumlarını ve işleyişini kapsar. Tek bir şema, birkaç bilgi setini düzenleyebilir ve bu nedenle, ele alınan durum için ayrıntıları doldurulabilecek değişkenlere sahip olabilir. Bir senaryo, farklı görev performansları için uygun eylem dizilerini sağlar. Şemalar ve senaryolar arasındaki bağlantılar, bireyin her adımda uygun eylemleri aktif olarak karar vermek zorunda kalmamasını sağlar ve bilişsel süreci büyük ölçüde kolaylaştırabilir.

Çalışma Belleği: Algılandıktan sonra bilgi, çalışma belleğinde depolanır. Diğer mekanizmaların (ilgili uzun süreli bellek depoları gibi) yokluğunda, bir

kişinin bilgi işleme sürecinin çoğu çalışma belleğinde gerçekleşmelidir. Yeni bilgi mevcut bilgi ile birleştirilmeli ve durumun birleşik bir resmi oluşturulmalıdır (durumsal kavrama). Gelecek durumların yansımaları ve uygun eylem planları da çalışma belleğinde gerçekleştirilmelidir. Bu bağlamda, çalışma belleği üzerinde ağır bir yük oluşur; çünkü aynı anda SA'nın daha yüksek seviyelerine (durumsal kavrama ve yansıma) ulaşmak, tepkileri formüle etmek ve seçmek, ve sonraki eylemleri gerçekleştirmekle görevlidir. Belirli prototipik durum koşulları için senaryolar geliştirildiğinde, alternatif davranışların oluşturulması ve bunlar arasından seçim yapılması için çalışma belleğine olan yük daha da azalır. Bu mekanizmanın önemli bir avantajı, mevcut durumun daha önce karşılaşılan bir duruma tam olarak benzememesi gerekliliğidir. Bu, kategorizasyon eşleştirmesi sonucudur (durumun özellikleri ile bilinen kategorilerin veya prototiplerin özellikleri arasında en uygun eşleşmenin sağlanması). Bu sürecin anlık olabilmesi, insanların üstün desen eşleştirme mekanizmaları sayesinde mümkündür.

4.1.2. Örneğe Dayalı Öğrenme Teorisi (IBLT)



Şekil 4.2. IBLT modeli (Gonzalez vd., 2003)

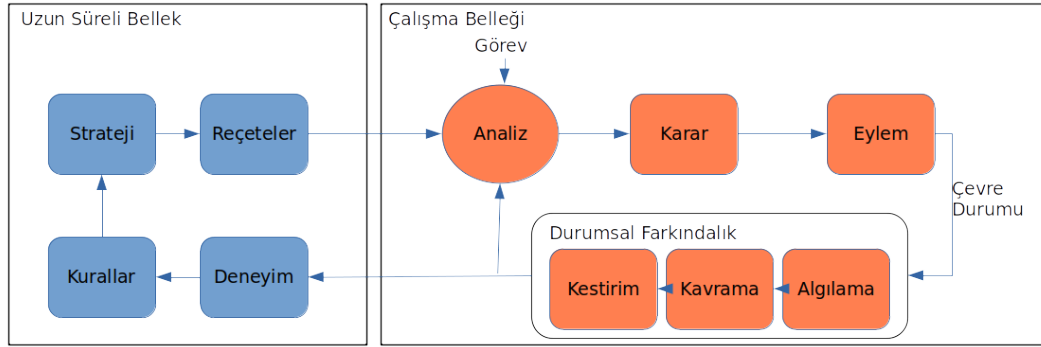
Örneğe Dayalı Öğrenme Teorisi (Instance-Based Learning Theory, IBLT) (Gonzalez vd., 2003, 2005), dinamik görevlerde karar verme davranışını yeniden üretmek amacıyla geliştirilmiştir. Bu teori, insan ve çevre etkileşimleriyle geri bildirim döngüsü süreci aracılığıyla deneyimlenen olaylar tarafından üretilen

eylem-sonuç bağlantılarının bellekte saklanmasını öğrenmenin temel karakteristikleri olarak kabul eder. IBLT, geçmiş benzer durumlardan elde edilen deneyimlere başvurarak ve bu geçmiş deneyimlerden yararlı kararları uygulayarak dinamik görevlerde karar verme olasılığını sunar. IBLT'nin en önemli katkısı, deneyimlerin nasıl inşa edilebileceğini, geri çağırılabilirliğini, değerlendirilebileceğini ve etkileşimler sırasında nasıl pekiştirilebileceğini açıklayan öğrenme süreçleri ve mekanizmalarını tanımlamasıdır. Bu model, basitliği ve geniş kapsamlı tahmin yapabilme yeteneği açısından önemlidir.

IBLT, Şekil 4.2'de gösterildiği gibi, dinamik görevlerde öğrenmeyi, karar alma olaylarını deneyimleyerek hafızada "örnekler" olarak depolayarak karakterize eder. Bu örnekler, üç ana öğeden oluşur: Durum (S, situation), bir dizi nitelik veya ipucu ile tanımlanan; Karar (D, decision), Durum S'de alınan eyleme karşılık gelen; ve Fayda veya Değer (U, utility or value), durum S'de karar D'nin beklenen veya alınan değeridir. Süreç, tanıma (benzerlik temelli geri çağırma), yargılama (bir durumda kararın beklenen faydasının değerlendirilmesi), seçim (optimal mevcut alternatifini seçme), yürütme (seçilen kararın uygulanması) ve geri bildirim (karar örneklerinin değerinin güncellenmesi) adımlarından oluşur.

Belirli bir karar durumuyla karşı karşıya kaldığında, insanların muhtemelen bellekten benzer SDU'ları (benzer durumlarla olan SDU'lar) geri çağıracağı düşünülmektedir (Tanıma adımı). Tipik bir durumda (geçmiş SDU'lara benzer durumlar), bir eylemin beklenen faydası, bellekten geri çağırılan benzer örneklerin faydalarını birleştirerek hesaplanır (Bu prosedüre Blending denir). Ancak, atipik durumlarda, insanlar eylemin beklenen faydasını değerlendirirken sezgisel yöntemlere geri dönerler. Bir kararın beklenen faydasının değerlendirilmesi Yargılama adımıyla yapılır. Alternatif eylemler sırasıyla değerlendirilir ve her değerlendirmeden sonra daha fazla alternatifin değerlendirilip değerlendirilmeyeceği, gereklilik mekanizması tarafından belirlenir. Gereklilik, karar vericinin kişisel tercihlerine veya zaman kısıtlamaları gibi dışsal faktörlere bağlı olarak subjektif bir şekilde belirlenebilir. Değerlendirilen alternatifler arasında en yüksek faydaya sahip olanı seçilir (Seçim adımı) ve uygulanır (Yürütme adımı), bu süreçle ortam değiştirilir ve bellekte hangi SDU'nun uygulandığı kaydedilir. Bir karar verildikten sonra, kararın sonucu, karar örneklerinin fayda değerini geri bildirim olarak değiştirmek için kullanılır (Geri Bildirim adımı).

4.2. Adaptif SLAM Karar Modeli



Şekil 4.3. Adaptif SLAM Karar Modeli

Adaptif SLAM Karar Modeli, Dinamik Karar Verme modeli ile Bellek Yönetimi'nde Örneğe Dayalı Öğrenme (IBLT) ve Deneyimlerden Öğrenme teorilerini bir araya getirerek oluşturulmuştur. Bu model, SLAM algoritmalarının bütüncül bir yaklaşımla çalışmasını sağlamak amacıyla tasarlanmıştır. Temel hedef, haritalama işlemi sırasında, göreve yönelik önemli nesnelere veya alanları keşfetmek ve zorlu durumların üstesinden gelmek için adaptif tepkiler üreten bir sistem geliştirmektir. Bu süreçte, sistemin kendi kendini beslemesi ve zaman içinde öğrenerek kendini sürekli geliştirmesi amaçlanmıştır.

Model, iki ana bellek bölümünden oluşur: uzun süreli bellek ve çalışma belleği. Çalışma belleği, kapalı döngü şeklinde işleyen bir karar mekanizmasını barındırır. Bu mekanizma, anlık kararlar almak ve adaptif tepkiler üretmek için kullanılır. Uzun süreli bellek ise, bu karar mekanizmasının çıktıları ve dahili kuralları çerçevesinde stratejiler ve reçeteler oluşturur. Bu stratejiler ve reçeteler, karar mekanizmasının gelecekteki kullanımı için çalışma belleğine sunulur. Bu iki bellek türü, sistemin etkinliğini ve performansını artırmak için işbirliği içinde çalışır.

Bu yapıyla, SLAM algoritmaları, dinamik ortamlarda etkin bir şekilde çalışabilir ve karşılaştıkları durumlara adapte olarak performanslarını artırabilir. Modelin bileşenleri ve işleyişi, insan karar verme süreçlerine benzer bir yaklaşımla tasarlanmış olup, zaman içinde sistemin kendi deneyimlerinden öğrenerek gelişmesine olanak tanır.

4.2.1. Çalışma Belleği

Çalışma belleği, kapalı döngü şeklinde bir durum farkındalığı bileşenini içerir. Bu bileşen, çevrenin durumunu algılar, kavrama sonrasında kestirim çıktıları ve hazır reçetelerle birlikte görev doğrultusunda analiz eder. Analiz sonucunda, reçeteler arasından eyleme yönelik karar verilir. Eylem sonuçları, çevreyi ve robot durumunu etkilediğinden, çıktılar geri bildirim olarak alınır ve bu geri bildirimler, analiz bileşeninde ve uzun süreli bellekte güncellenen deneyimlerle birlikte değerlendirilir.

4.2.1.1. Durumsal Farkındalık

Durumsal farkındalık, SLAM algoritmalarının etkinliğini ve doğruluğunu artırmada kritik bir rol oynar. Durumsal farkındalık, bir robotun çevresindeki ortamı anlamasını, değişen koşullara hızla adapte olmasını ve bu doğrultuda doğru kararlar almasını sağlar.

Durumsal farkındalığın SLAM'a uyarlanması, algoritmanın çevresel faktörleri ve hareket koşullarını sürekli olarak izlemesini ve değerlendirmesini içerir. Örneğin, aydınlatma koşullarındaki değişiklikler, hareket hızındaki ani artışlar veya dinamik engellerin varlığı gibi faktörler, algoritmanın davranışını etkileyebilir. Bu farkındalık, SLAM algoritmasının çevredeki değişiklikleri tespit etmesini ve bu değişikliklere göre yol haritasını, yer belirleme ve haritalama stratejilerini dinamik olarak ayarlamasını sağlar.

Bu bağlamda, durumsal farkındalığın çıktısı olarak bir desen oluşturulur. Bu desen, anlamlandırılmış haritalarla ilişkilendirilir ve önemli nesnelere, zorluklara ve yakın geleceğe dair tahminler gibi bilgileri içerir. Desen, odometri verileri, çevresel zorluklar ve diğer durum bileşenlerini içeren bir yapıyı temsil eder.

Desen İçeriği:

Önemli Nesnelere: Çevredeki kritik nesnelere, nesnelere konumları ve işlevleri hakkında bilgi sunar.

Zorluklar: Çevredeki dinamik engeller veya zorluklar, algoritmanın performansını etkileyebilecek potansiyel sorunları temsil eder.

Yakın Geleceğe Dair Tahminler: Çevredeki değişikliklerin ve robotun

hareketlerinin yakın gelecekte nasıl bir etki yaratabileceğine dair tahminler sunar.

Desen, çevresel bağlamın—robotun konumu, hızı, duruşu ve çevredeki engeller gibi faktörlerin—derinlemesine anlaşılmasıyla oluşturulur ve bu verilerin birleşimi, robotun gelecekteki durumlarını daha doğru bir şekilde tahmin etmesine olanak tanır. Bu desen, SLAM algoritmasının çevresel değişikliklere karşı adaptif ve esnek olmasını sağlar. Böylece, çevresel faktörlerdeki değişiklikler, aydınlatma koşullarındaki varyasyonlar ve diğer dinamik etkiler göz önüne alındığında, algoritma uygun bir şekilde tepki verir. Bu sayede, SLAM sistemleri daha doğru yer belirleme ve haritalama sonuçları elde edebilir. Desen içeriğinin zenginleştirilmesi, daha sağlam SLAM sistemlerine ulaşılmasını sağlar.

Durumsal farkındalığın SLAM algoritmalarına entegre edilmesi, aynı zamanda hatalı tahminlerin ve yanlış pozisyonlamaların önüne geçilmesine yardımcı olur. Böylece, gerçek dünya uygulamalarında daha güvenilir, esnek ve dayanıklı sistemler geliştirilmesine olanak tanır.

Durumun Algılanması: Geleneksel robot tasarımı genellikle sensör seçimini ve konfigürasyonunu belirli bir göreve uyarlamayı içerir. Ancak bu yaklaşım, insan benzeri algılamayı sağlamakta yetersiz kalabilir. Bu sınırlamayı aşmak için, robotların çevrelerini ve dinamik unsurları detaylı ve doğru bir şekilde algılayabilen standart ve çok yönlü bir sensör seti ile donatılması gerekmektedir. Bu sensör seti, gelişmiş işleme algoritmalarıyla birleştiğinde, farkındalık artırılarak farklı açılardan dayanıklılık sağlar.

Durumsal algılama, dış çevreye dair bilgiler (örneğin, görsel ışık yoğunluğu veya mesafe) ve robotun iç parametreleri (örneğin, hız veya sıcaklık) gibi bilgilerin toplanmasını içerir. Bu bilgiler, robotun çevresindeki detayları algılayarak düşük performansı telafi etmek için kullanılabilir. SLAM algoritmalarında, Lidar, kamera ve IMU gibi sensörlerin yanı sıra çevresel zorlukları anlamaya yönelik sensörler de kullanılmalıdır. Doğru durumsal algılama, bilinçli kararlar almanın temelini oluşturur. Ancak, çevresel değişiklikler, örneğin olumsuz hava koşulları, yetersiz ışık veya sensör arızaları, algılama sistemini olumsuz etkileyebilir. Bu durumlarda, sistemin yeterli doğrulukta bilgi elde edememesi yanlış kararlar alınmasına yol açabilir. Ancak, hataya dair algılamayı sağlayacak veriler ile hata farkındalığının kazandırılması, adaptif tepkiler üretilmesine olanak tanıyarak, sistemin kendini toparlamasını ve hataları telafi etmesini sağlar.

Durumun Kavranması: Durumsal kavrama, sensörlerden alınan bilginin

çevre veya robot durumu ile ilgili bilgi edinmeye yönelik işlenmesini ifade eder. Bu kavrama, görüntü işleme ile nesne tespiti veya çoklu sensör füzyonu ile olumsuz şartların tespiti gibi işlemleri kapsar. Durumsal kavrama, algılamanın anlaşılmasından olası anlamsal ilişkilerin dikkate alınmasına kadar geniş bir yelpazeyi içerir ve belirli bir zaman diliminde algısal gözlemler kullanılarak kısa vadeli bir kavrama oluşturur. Ayrıca, geçmişte edinilen bilgileri içeren uzun vadeli bir model, birikmiş durumsal kavrama olarak adlandırılır. Bu model, geometrik (örneğin, nesnelerin şekli), anlamsal (örneğin, nesnelerin türü ve işlevi), topolojik (örneğin, mekandaki sıralama), ontolojik (örneğin, genel bilgi kavramlarının hiyerarşisi), dinamik (örneğin, nesnelere arasındaki hareket) veya stokastik (örneğin, belirsizlik bilgisini dahil etmek) gibi çoklu soyut ilişkiler içerir. Bu ilişkilerin kavranması ve tespit edilmesi çevrenin bütünsel bir resmini oluşturmayı ve nesnelerin önemini kavramayı sağlar.

Sensörlerden elde edilen veriler, bu zorlukların ve nesnelerin haritadaki yerlerini ve etkilerini doğru bir şekilde yansıtmak için kullanılır. Bu verilerle, zorluklar ve nesnelere ve diğer ilişkiler için bir sözlük oluşturulur. Sözlük, her bir zorluk veya nesne için detaylı tanımlar ve bu tanımların şiddetini belirten değerler içerir. Bu değerler, zorlukların veya nesnelerin sistem üzerindeki etkisini nicel olarak ifade eder. Sözlükteki her bir durum, odometriden elde edilen X, Y koordinat verileri bilgisi ile ilişkilendirilir. Bu şekilde, her bir zorluk veya nesne, harita üzerinde doğru bir konumda yer alır ve zamanla değişen çevresel koşulları yansıtacak şekilde güncellenir. Örneğin, bir odadaki karanlık bölgeler veya belirli bir nesnenin hareketi, sözlükteki bir durumu temsil eder. Bu, robotun çevresel değişikliklere daha iyi yanıt vermesini ve daha doğru haritalama sonuçları elde etmesini sağlar.

Her bir zorluğun anlamlandırma ve zorluk seviyelerinin hesaplamalandığı sözlük çalışmaları bu adımda gerçekleştirilir. Her bir özellik noktası, bir (x, y) koordinatına sahip ve bu noktalar için farklı zorluk seviyeleri z_1, z_2, \dots, z_n olarak tanımlandığı durumda, bu bilgileri bir matris yapısında şu şekilde gösterebiliriz:

$$D = \begin{pmatrix} x_1 & y_1 & z_1 & z_2 & \dots & z_n \\ x_2 & y_2 & z_1 & z_2 & \dots & z_n \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ x_m & y_m & z_1 & z_2 & \dots & z_n \end{pmatrix}$$

D : Durumsal kavramayı temsil eden durum matrisidir.

x_i, y_i : Her bir i özellik noktasının koordinatları.

z_1, z_2, \dots, z_n : Her bir özellik noktası için hesaplanan zorluk seviyeleri.

Her satır, bir özellik noktasını ve o nokta için hesaplanan zorluk seviyelerini temsil eder. Bu zorluk seviyeleri, örneğin bir görüntünün farklı zorluk düzeylerinde (ışık, hareket, bulanıklık gibi) nasıl performans gösterdiğini temsil edebilir.

Kavrama adımında elde edilen bu matris bir desen olarak yansıtma adımına SLAM haritaları ile birlikte aktarılır.

Durumun Geleceğe Yansıtılması: Durumun geleceğe yansıtılması, robotik sistemlerin karar verme süreçlerinde kritik bir rol oynar. Çevresel bağlamın—robotun konumu, hızı, duruşu, çevredeki engeller gibi faktörlerin—derinlemesine anlaşılması, daha doğru bir çıkarım yapılmasına olanak tanır. Bu model, robotun hem kendi gelecekteki durumunu hem de çevresindeki değişiklikleri tahmin etmesine, olası davranışları ve etkileşimleri kestirim yapmasına ve hedeflerine ulaşmak için eylemlerini uyarlamasına yardımcı olur.

Kavrama adımında elde edilen matris, yansıtma adımında çevresel koşullara dair kestirimler oluşturmak için kullanılır. Örneğin, z_1 ortam aydınlatmasına dair bilgiyi içeren bir değişken olabilir. Eğer matris içerisinde zorluk seviyeleri karanlığa doğru bir eğilim gösteriyorsa, sistem bu durumu çıkarır ve robotun karanlık bir ortama girmekte olduğunu kestirim yapabilir. Her bir zorluk tipi için yapılan kestirimler bir dizi olarak durum matrisi ile birlikte çıktı olarak verilir.

$$K = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & \dots & k_n \end{bmatrix}$$

T : Her bir durum değişkeni olan z_1, z_2, \dots, z_n zorluk için elde edilen tahmin dizisi.

Bu sürecin sonunda, robotun çevresel farkındalığını artırarak bilinçli kararlar almasına olanak tanır ve gelecekteki durumların daha isabetli bir şekilde tahmin edilmesine katkıda bulunur.

4.2.1.2. Analiz

Analiz bileşeninde, Durumsal Farkındalık bölümünden elde edilen desenler, görev doğrultusunda strateji ve reçetelerin değerlendirilmesi için kritik bir girdi sağlar.

Analiz sürecinde, desen mevcut strateji ve reçetelerle karşılaştırılır. Eğer sistem, daha önce karşılaştığı benzer durumlar varsa, bu durumlara uygun strateji

ve reçeteler tercih edilir. Bu süreç, daha önce elde edilen desenlerin ve stratejilerin sistemin mevcut durumuna uygunluğunu değerlendirmeyi amaçlar. Eğer mevcut reçeteler mevcut duruma uygun değilse, desen analizi kullanılarak benzer durumlar tespit edilir ve bu benzer durumlar için en uygun strateji seçilir.

Bu süreç, bir karar ağacı yaklaşımına benzer şekilde işler ancak her durumun belirli bir reçeteye doğrudan karşılık gelmesi gerekmez. Durumsal farkındalık bileşeninden gelen bilgilerin genel analizi yapılır ve en uygun strateji belirlenir.

4.2.1.3. Karar Verme

Seçilen reçetelerin, anlık sensör verileri ve durumsal farkındalık desenleri ile gerçek zamanlı olarak kontrol edilmesi, SLAM algoritmalarının adaptif yeteneklerini optimize etmek için kritik bir adımdır. Bu aşama, sistemin çevresel koşullara hızla uyum sağlamasını ve hatasız bir haritalama ve konumlandırma performansı elde etmesini hedefler.

Reçete, belirli bir durumsal farkındalık senaryosuna uygun eylem planını içeren bir dizi yönergeyi ifade eder. Seçilen reçetenin uygulanması, sistemin mevcut durumunu sürekli olarak izleyerek ve anlık sensör verilerini işleyerek gerçekleştirilir.

4.2.1.4. Eylem

Eylem bileşeni, karar verme sürecinin ardından belirlenen stratejilerin ve reçetelerin uygulamaya geçirilmesi aşamasıdır. Bu basamak, robotun belirlenen hedeflere ulaşmak için gerekli eylemleri gerçekleştirir ve sistemin çevresel değişikliklere ve hedeflere yönelik tepkilerini dinamik olarak yönetir.

4.2.2. Uzun Süreli Bellek

Uzun süreli bellek, deneyimlerin ve geçmiş bilgilerin depolandığı ve değerlendirildiği bir sistem bileşeni olarak, stratejilerin ve hazır reçetelerin oluşturulmasında kritik bir rol oynar. Çalışma belleği, bu reçeteleri gerektiğinde anlık olarak uygulayarak, sistemin çevresel koşullara dinamik bir şekilde yanıt

vermesini sağlar. Uzun süreli belleğin en önemli işlevlerinden biri, çalışma belleğindeki yükü hafifletmek ve strateji geliştirme süreçlerini desteklemektir.

Bu bellek, haritalar, yörüngeler, ilişkilendirilmiş durumlar ve tahmin desenleri gibi çeşitli verileri içerir ve bu veriler deneyim bileşeninde değerlendirilir. Deneyim bileşeninden elde edilen bilgiler, kurallar bileşeni doğrultusunda stratejiler geliştirilmesine ve reçeteler hazırlanmasına olanak tanır. Özellikle sistemin ilk çalışmaya başladığı durumlarda, bu belleğe önceden tanımlanmış strateji ve reçeteler yüklenebilir. Böylece, sistem başlangıçta belirli durumlar için hazırlıklı olur ve daha hızlı tepki verebilir.

Uzun süreli bellek, şematik bir yapı üzerine inşa edilmiştir. Çalışma belleğinden gelen bilgilerin, tipik durumlar için şematize edilmesi, belleğin daha hızlı ve düzenli yönetilmesini sağlar. İnsanlarda olduğu gibi, olayların karakterize edilmesi ve eylemlerin önceden belirlenmesi, sistemin tepki süresini kısaltır ve kritik durumlarda daha hızlı ve doğru kararlar almasına olanak tanır. Örneğin, ışık koşullarının değiştiği bir ortamda sistem, otomatik olarak aydınlatma seviyesine uygun bir SLAM algoritması seçebilir.

Şemalar, hafızadaki unsurlar ile çevredeki kritik ipuçları arasında desen eşleştirmesine dayalı olarak çalışır. Tipik durumların şemaları, tek adımda eylemlerin hafızadan alınmasını sağlamak için reçetelerle ilişkilendirilebilir. Bu süreç, sistemin daha hızlı ve verimli bir şekilde çalışmasına katkıda bulunur.

Deneyim bileşeni, haritalama süreçlerinde yaşanan başarısızlıklar ve bu durumların yerleştirme sistemine olan etkilerini dikkate alarak sürekli gelişim sağlar. Kurallar bileşeni ise SLAM algoritmalarının yürütülmesi ve diğer işlemler için genel kuralları ve sınırlamaları belirler.

Sonuç olarak, uzun süreli bellek, strateji geliştirme ve çevresel değişikliklere uyum sağlama süreçlerinde önemli bir bileşen olarak işlev görür. Bu bellek, harita üzerinde belirli bölgelere dair kritik bilgileri ve olumsuzlukları depolar ve bu bilgiler, sistemin gelecekteki görevlerde daha bilinçli ve etkin bir şekilde hareket etmesine olanak tanır.

4.2.2.1. Deneyim

Deneyimler, haritalama ve yerleştirme süreçlerinde karşılaşılan özel durumların ve bu durumlara verilen tepkilerin kayıt altına alındığı önemli bir bileşendir. Örneğin, belirli bir yol üzerindeki çukurlar veya dinamik engeller gibi çevresel unsurlar burada saklanır ve bu bilgiler, sistemin çevresel faktörlere yönelik tepkilerini optimize etmek amacıyla kullanılır. Bu süreçte, elde edilen sonuçlar uzun süreli hafızada depolanarak, gelecekteki karar süreçlerinde referans alınır.

Sistem, yeni durumları tanıma ve geçmiş deneyimlerle ilişkilendirme yeteneği sayesinde, daha önce başarılı bir şekilde uygulanan stratejileri hızla devreye sokabilir. Örneğin, bir insanın bir çukuru fark ettikten sonra sonraki geçişlerinde ondan kaçınmak için geliştirdiği refleks, sistemde benzer şekilde işlev görür. Mevcut durum daha önce deneyimlenmemişse ve bu duruma yönelik bir reçete bulunmuyorsa, sistem yeni bir eylem planı geliştirir. Bu plan, önceki deneyimlerle mevcut durum arasındaki benzerliklerin değerlendirilmesi yoluyla oluşturulur ve olumlu sonuç beklentisi taşıyan bir eylem bileşenine dönüştürülür. Planın başarısı, geri bildirimlerle değerlendirilir ve bu bilgiler uzun süreli hafızaya kaydedilir. Bu süreç, sistemin öğrenme ve adaptasyon yeteneklerini sürekli olarak geliştiren bir döngü yaratır.

Görev ve çevre koşullarıyla en iyi uyum sağlayan stratejinin belirlenmesi, sistemin dinamik ortamlara adaptasyonu ve doğru kararlar alması açısından kritik bir öneme sahiptir. Olumsuz koşullar altında geliştirilen stratejilerin başarı oranları dikkate alınarak, sistemin çevresel değişimlere karşı daha dirençli hale gelmesi sağlanır. Eğer geri bildirimler, mevcut stratejinin beklenen sonuçları vermediğini gösterirse, strateji güncellenir veya yeni bir strateji geliştirilir. Bu dinamik süreç, yansıtma bileşeninden elde edilen desenlerin sürekli değerlendirilmesi ve uzun süreli hafızada biriken bilgilerle beslenir, böylece sistemin gelecekteki yansıtma yetenekleri optimize edilir.

Sonuç olarak, deneyim yönetimi, sistemin haritalama ve yerleştirme süreçlerinde karşılaştığı zorlukları değerlendiren ve gelişmesini sağlayan temel bir bileşen olarak öne çıkar. Sistemin hafızasında biriken bilgiler, çevresel değişikliklere karşı daha dirençli ve adaptif stratejiler geliştirilmesini destekler. Örneğin, sistem, daha önce karanlık bir ortamda yaşanan bir başarısızlığı kaydederek, benzer bir

duruma tekrar girdiğinde alternatif bir strateji uygulayabilir. Bu süreç, sistemin öğrenme yeteneklerini sürekli olarak besleyerek, daha etkili ve hızlı kararlar almasını sağlar.

4.2.2.2. Kurallar

Kurallar bileşeni, SLAM algoritmalarının yürütülmesi ve diğer sistem işlemleri için genel yönergeler ve sınırlamaların tanımlandığı bir alandır. Bu kurallar, sistemin genel stratejilerini yönlendirir ve belirli durumlarda hangi eylemlerin uygulanacağını belirler. Ayrıca, çevresel farkındalığı etkileyen faktörlere yönelik rehberlik sağlayarak, sistemin çevresel koşullara uygun kararlar almasını destekler.

4.2.2.3. Stratejiler

Stratejiler, sistemin çevresel koşullara karşı nasıl tepki vereceğini belirleyen, deneyim ve kurallar çerçevesinde oluşturulmuş uzun vadeli planlardır. Bu stratejiler, sistemin çevresini algılamasında ve anlamlandırmasında önemli bir rol oynar ve sistemin çeşitli durumlar karşısında vereceği tepkileri organize eder. Stratejiler, üç temel süreci içerir:

Algıda Seçicilik: Stratejiler, sistemin dikkatini odaklayacağı önemli çevresel faktörleri belirler. Örneğin, belirli bir bölgedeki engeller veya aydınlatma seviyeleri, stratejiler tarafından önceliklendirilir. Bu süreç, sistemin kritik bilgilere hızlı bir şekilde erişmesini ve bu bilgilere dayalı olarak tepki vermesini sağlar.

Anlamlandırma: Stratejiler, sistemin çevresel verileri nasıl anlamlandıracağını belirler. Bu süreçte, çevresel faktörler değerlendirilir ve mevcut durumun bir bağlam içinde nasıl ele alınacağına karar verilir. Örneğin, bir çukur algılandığında, bu bilginin nasıl yorumlanacağı ve sistemin bu duruma nasıl tepki vereceği stratejilerle belirlenir. Anlamlandırma, sistemin daha önceki deneyimlere dayalı olarak uygun eylemleri seçmesini sağlar.

Tepki Süreçleri: Stratejiler, çevresel faktörlere nasıl tepki verileceğini tanımlar. Bu süreçte, sistemin önceden belirlenmiş kurallara ve deneyimlere dayalı olarak hangi eylemleri gerçekleştireceği belirlenir. Örneğin, bir engelle karşılaşıldığında, stratejiler sistemin en uygun alternatif yolu seçmesini sağlayabilir.

Bu, sistemin hızlı ve etkili bir şekilde karar almasını mümkün kılar.

Stratejiler, sistemin çevresel farkındalığını ve adaptasyon yeteneğini geliştirir. Her strateji, belirli koşullara karşı optimize edilmiş bir tepki süreci sunar ve bu süreçler, sistemin genel performansını artırır. Örneğin, bir alanda sürekli tekrarlanan zorluklar karşısında stratejiler, o bölgeye özgü özel eylem planları oluşturabilir. Bu planlar, gelecekte benzer zorluklarla karşılaşıldığında, sistemin en uygun ve hızlı tepkiyi vermesini sağlar.

Stratejiler, sistemin çevresel koşulları dinamik bir şekilde değerlendirmesine ve bu değerlendirmeler doğrultusunda en etkili kararları almasına olanak tanır. Stratejiler, uzun süreli bellekteki deneyimlerle desteklenir ve bu deneyimler sayesinde sürekli olarak geliştirilebilir. Bu da sistemin çevresel faktörlere karşı esnek ve etkin bir tepki mekanizması geliştirmesine olanak tanır.

4.2.2.4. Reçeteler

Reçeteler, belirli durumlar için uygulanacak işlem adımlarını tanımlayan, sistemin hızlı ve etkili bir şekilde tepki vermesini sağlayan temel bir bileşendir. Bu bölüm, sistemin karşılaştığı tipik durumlar için önceden belirlenmiş eylem planlarını içerir. Reçeteler, sistemin çevresel değişikliklere anında ve otomatik olarak uyum sağlayabilmesi için tasarlanmıştır. Bu yapı, uzun süreli belleğin işlevselliğini artırır ve sistemin performansını optimize eder.

Reçeteler, belirli ve tekrarlayan durumlar için geliştirilmiş eylem adımlarını içerir. Bu durumlar, geçmiş deneyimlerden elde edilen bilgilerle şekillendirilir ve sistemin daha önce karşılaştığı benzer olaylar doğrultusunda optimize edilir. Örneğin, bir engelle karşılaşıldığında, sistemin hızlı bir şekilde alternatif bir yol bulması ve bu yolu takip etmesi için bir reçete devreye girer. Bu sayede, sistemin tepki süresi minimuma indirilir ve etkili bir çözüm sağlanır.

Bazı reçeteler, belirli koşullar altında otomatik olarak devreye girer. Bu, sistemin çevresel değişikliklere anında tepki vermesini sağlar. Örneğin, ışık seviyesinin belirli bir değerin altına düştüğü durumlarda, sistemin otomatik olarak farklı bir sensör veya haritalama algoritmasına geçiş yapmasını sağlayan bir reçete devreye girebilir. Bu tür otomatik reçeteler, sistemin çevresel koşullara dinamik bir şekilde uyum sağlamasına yardımcı olur ve kesintisiz bir performans sunar.

Reçeteler, stratejilerin uygulanmasında kritik bir rol oynar. Seçilen bir stratejiyi destekleyen bir reçete mevcutsa, sistem bu reçeteyi kullanarak stratejiyi hayata geçirir. Ancak, mevcut bir reçete yoksa, stratejinin başarılı bir şekilde tamamlanabilmesi için yeni eylemler geliştirilir. Bu süreçte, sistemin dikkati çevresel özelliklere yöneltilir ve bu özellikler, hedef-aktive edilmiş modeller tarafından yorumlanır. Bu durum, sistemin dinamik koşullar altında bile doğru ve hızlı kararlar almasını sağlar.

Reçeteler, uzun süreli bellek ile çalışma belleği arasında bir köprü oluşturarak, sistemin performansını artırır. Çalışma belleği, bu hazır reçeteleri anlık olarak kullanır ve sistemin çevresel koşullara hızlı ve etkili bir şekilde tepki vermesini sağlar. Böylece, çevresel farkındalık artar ve SLAM algoritmasının olumsuz koşullar karşısındaki dayanıklılığı güçlenir.

4.3. Tartışma

Adaptif SLAM karar modeli, robotların dinamik ortamlarda daha etkili ve esnek bir şekilde hareket etmelerini sağlamak üzere geliştirilmiştir. Model, çevresel farkındalık, sensör çeşitliliği, yapay zeka ve derin öğrenme gibi teknolojilerle ilerlemelerden yararlanarak, robotların karmaşık görevleri yerine getirme ve gelecekteki durumları daha başarılı bir şekilde tahmin etme kapasitelerini artırmayı hedeflemektedir.

Bu modelin en büyük avantajlarından biri, durumsal farkındalık düzeyini artırarak robotların çevresel değişikliklere hızlı ve uygun şekilde tepki verme yeteneğini geliştirmesidir. Durumsal farkındalık, yalnızca çevredeki verilerin algılanmasından ibaret değildir; bu verilerin anlamlandırılması ve bu doğrultuda uygun adaptif tepkilerin üretilmesini de içerir. Bu süreç, robotların görevlerini daha hızlı ve etkin bir biçimde yerine getirmelerine olanak tanır.

Adaptif SLAM modelinin başarısında kilit rol oynayan unsurlardan biri de sensör çeşitliliğidir. Farklı sensörlerin sağladığı verilerin entegre edilmesi, belirli koşullar altında daha doğru sonuçlar elde edilmesine olanak tanır. Sensörlerin ağırlıklarının akıllıca seçimi, robotların çevresel zorluklara karşı daha dirençli olmasını sağlar. Tek bir sensöre bağımlı kalmaksızın, çeşitli sensörlerden elde edilen verilerin kombinasyonu, hata oranını azaltır ve genel performansı iyileştirir.

Bununla birlikte, uzun süreli çalışma ve tekrar eden görevler, sistemin aşırı öğrenme riskini ortaya çıkarabilir. Aşırı öğrenme, modelin belirli durumlarda performansını artırırken, beklenmedik değişikliklere karşı esnekliğini kaybetmesine neden olabilir. Bu durum, özellikle yüksek çevresel farkındalık gerektiren senaryolarda sorun yaratabilir ve sistemin adaptasyon yeteneğini sınırlayabilir.

Durumsal farkındalık, robotların çevresel koşulları anlamlandırmasına yardımcı olmakla birlikte, gelecekteki durum tahminleri hala önemli bir zorluk olarak karşımızda durmaktadır. Mevcut algoritmalar, bu tahminleri yaparken algılama ve anlamlandırma katmanlarından mükemmel verilere ihtiyaç duyarlar. Ancak, verilerin kusursuz olmadığı durumlarda algoritmaların performansı ciddi şekilde etkilenebilir, bu da robotların beklenmedik durumlarla karşılaştıklarında hata yapma olasılığını artırır.

Bugüne kadar, SLAM yöntemleri daha çok navigasyon doğruluğu üzerine odaklanmış durumdadır. Ancak, süreklilik, bütünlük ve kullanılabilirlik gibi diğer navigasyon performans parametrelerinin de dikkate alınması gerekmektedir. Bu parametrelerin değerlendirilmesi, sistemin gerçek dünya uygulamalarında daha güvenilir ve dayanıklı olmasını sağlayacaktır.

Yapay zeka ve derin öğrenme algoritmalarının adaptif SLAM modelinde kullanımı, bu modelin en karmaşık görevlerde dahi başarılı olmasını sağlar. Bu algoritmalar, genelleme yeteneği ve dinamik ortamlarda adaptasyon gibi zorlukların üstesinden gelerek, robotların belirli bir çevrede daha hızlı öğrenmesini ve daha etkin bir şekilde hareket etmesini mümkün kılacaktır. Aynı zamanda, model içerisinde yapay zeka ve derin öğrenme algoritmalarının uygulanmasına yönelik bir prosedür oluşturulması, sistemin performansını artıracaktır.

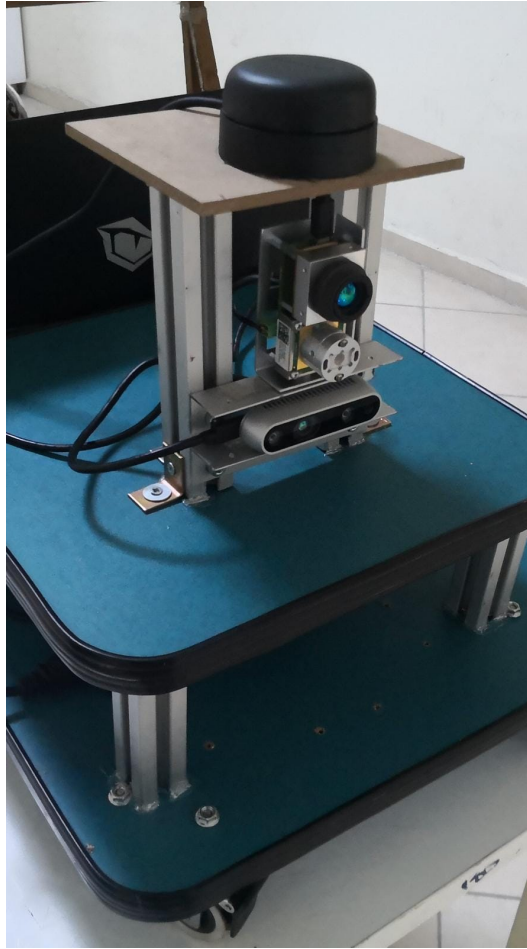
5. KONYA VERİ SETİ

Konya veri seti, stereo kameralar, Lidar ve IMU sensörleri ile donatılmış bir kurulumdan oluşmakta olup, stereo görüşe dayalı algoritmaları, görüntü-IMU füzyonunu, görüntü-Lidar füzyonunu, görüntü-Lidar-IMU füzyonunu vb. SLAM algoritmalarını desteklemektedir. Toplulukla paylaştığımız bu veri seti, 2D Lidar, RGB-D kamera, Leddar katı hal Lidar ve bir ışık sensörü verilerinden oluşmaktadır. Farklı ortamlarda tekrarlı olarak toplanan bu diziler, çeşitli zorluklar barındırarak gerçek dünyayı en iyi şekilde yansıtmaya çalışmaktadır.

Veri seti, ROS aracılığıyla toplandı ve ROSbag dosyaları olarak kaydedildi. ROS, robotik uygulamaların geliştirilmesi, test edilmesi ve dağıtılması için gerekli birçok araç, kütüphane ve konfigürasyon dosyasını sunar. Modüler yapısı sayesinde, farklı sensörler, aktüatörler ve yazılım bileşenleri arasında kolayca entegrasyon sağlanabilir. ROS, bağımsız süreçler aracılığıyla çalışan ve bu süreçler arasında mesajlaşma sistemini kullanan bir mimari sunar. Bu sayede, robot sistemlerinin karmaşık görevleri modüler ve yeniden kullanılabilir bir şekilde yerine getirmesi sağlanır. ROSbag, özellikle sensör verilerinin ve sistem durumlarının zaman damgalı olarak saklanması sağlar, bu da gerçek zamanlı testler yapmadan verilerin analiz edilmesine ve sistemlerin geliştirilmesine olanak tanır. Bu özellik, özellikle karmaşık robotik sistemlerde gerçek donanım üzerinde test yapmanın mümkün olmadığı durumlarda önemli avantajlar sağladığından, araştırmacılar tarafından yaygın olarak kabul görmektedir.

5.1. Sensör Platformu ve Sensörler

Veri seti toplama sürecinde kullanılan mobil platform, bir veri setinin hareket desenini belirler. Örneğin, hızlı hareketin (sallanma, keskin dönüş/rotasyon/yer değiştirme ve yüksek hız) veri kalitesini etkileyebileceği (kamera hareketi bulanıklığı ve Lidar bozulması gibi) belirlenmiştir. Ayrıca, saf rotasyon, monoküler durumda temel matrisin dejenerasyonuna neden olabilir ve yüksek hız, bitişik çerçeveler arasında dar örtüşmeye sebep olabilir. Bu zorlukları yansıtmayı

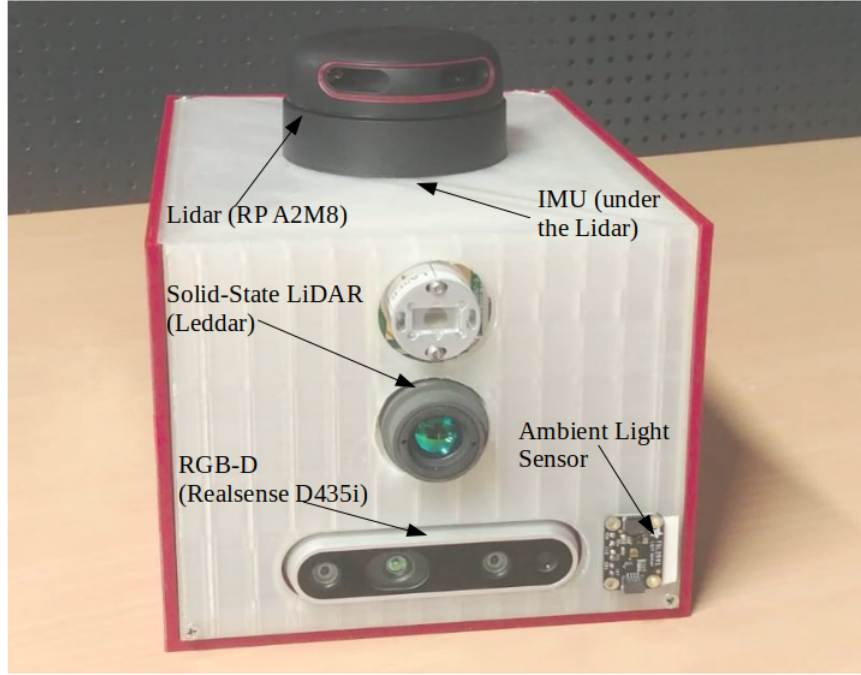


Şekil 5.1. Tekerlekli Sensör Platformu

amaçladığımız veri setimizde, ilk olarak prototip niteliğinde olan bir tekerlekli platform ile veri toplanmıştır Şekil 5.1. Tekerlekli araç üzerinde sensörler, y ekseninde eş merkezli olarak yerleştirilmiş olup, veriler doğrudan araba üzerine yerleştirilen dizüstü bilgisayar ile kaydedilmiştir. Devam eden çalışmalarda ise, sensörlerin füzyona elverişli olarak yerleşim planına sahip, tüm donanımı içerisine yerleştirmeye uygun ve elde taşımaya veya herhangi bir platform üzerine monte etmeye elverişli bir platform tasarlanmıştır Şekil 5.2. Bu platform, 3D yazıcıdan çıktı alınarak üretilmiş olup, serbestlik derecesi 6 DoF'dır.

Robot simülasyonlarında, görselleştirme araçlarında ve kontrol yazılımlarında platformun doğru bir şekilde modellenmesi ve davranışlarının tahmin edilmesi için ROS ortamında URDF dosyaları oluşturulmuştur. URDF (Unified Robot Description Format), robotların fiziksel yapılarını ve bileşenlerini tanımlamak için kullanılan XML tabanlı bir dosya formatıdır. URDF, robotun bağlantı noktalarını (link) ve bu bağlantılar arasındaki eklemleri (joint) detaylı bir şekilde tanımlar. Bu sayede, robotun geometrik yapısı, kinematik zinciri ve fiziksel

özellikleri bir dosya içinde standart bir formatta ifade edilir.



Şekil 5.2. Elde Taşınan Sensör Platformu

Veri seti toplama sürecinde kullanılan sensörler, Intel Realsense D435i RGB-D kamera, Slamtec RP A2 M8 2D Lidar, Leddar Vu8 model katı hal Lidar ve TSL2591 model ışık sensörü gibi çeşitli algılama teknolojilerine sahiptir. Veriler, bu sensörlerle donatılmış bir kutu içerisine yerleştirilen Asus PN50 AMD Ryzen 5 4500U 16 GB 1TB SSD mini bilgisayar ile toplanmış ve kaydedilmiştir. Sensör özellikleri ve kullanılan topic isimleri ile mesaj tipleri, Tablo 5.1’de özetlenmiştir.

Topic ve Mesaj Veri Tipleri: ROS içinde haberleşme mekanizması olan topic’ler, farklı ROS düğümlerinin (nodes) birbirleriyle veri alışverişi yapmasını sağlar. Düğümler, belirli bir topic’e mesajlar yayınlayabilir (publish) veya bu topic’teki mesajları dinleyebilir (subscribe). Mesaj veri tipi (message type), belirli bir topic üzerinden iletilen verinin yapısını tanımlar. ROS’ta mesaj veri tipleri genellikle std_msgs, sensor_msgs gibi önceden tanımlanmış tiplerdir, ancak kullanıcılar kendi özel mesaj tiplerini de tanımlayabilirler.

Intel RealSense D435i: Bu gelişmiş stereo derinlik kamerası, hem renkli (RGB) hem de derinlik (D) bilgileri sağlayarak çeşitli görüntüleme ve algılama uygulamaları için kullanılmaktadır. Dahili bir IMU (İnertial Measurement Unit) sayesinde, cihazın hareket bilgilerini hassas bir şekilde algılar; bu da cihazın

Çizelge 5.1. Sensör Tipleri ve Özellikleri

Sensör	Özellikleri	Topic ismi
Lidar	Distance Range: 0.15-8m Angular Range: 0-360 Scan Rate: 5-15 Hz	/scan
RGB-D	Horizontal Field of View: 91.2° Vertical Field of View: 65.5° Diagonal Field of View: 100.6° IR Projector (Wide) FOV: H:90 / V:63 / D:99 Range: D435i, 0.2-10 m IMU: 6DoF	/camera/depth/camera_info /camera/depth/image_rect_raw /camera/color/camera_info /camera/color/image_raw /camera/infra1/camera_info /camera/infra1/image_rect_raw/ /camera/infra2/camera_info /camera/infra2/image_rect_raw camera/imu /camera/accel/imu_info /camera/gyro/imu_info
Solid-State Lidar	Horizontal FOV: 47.5° Vertical FOV: 3° Resolution: 1 cm Data refresh rate: Up to 100 Hz Distance: Up to 75 meters	/LeddarTech_1/scan_raw /LeddarTech_1/scan_cloud
Ambient Light Sensor	Wide Dynamic Range: 600 M:1 Operational Light Range: 188 μ Lux to 88,000 Lux	/lux

konumlandırılmasında önemli bir rol oynar. Ayrıca, kızılötesi (IR) projeksiyon ve IR kamera, düşük ışık koşullarında bile daha iyi derinlik bilgisi sunarak kullanım alanını genişletir.

SLAMTEC RPLidar A2 M8: Bu Lidar sensörü, 360 derece 2D tarama özelliği ile çevresel engel algılama ve taramada yüksek performans sağlar. Hafif ve kompakt boyutları, robotik ve navigasyon uygulamalarında kolay entegrasyonu destekler. SLAMTEC RPLidar A2 M8, düşük güç tüketimi (0.5W) ve 300-400 USD arasında değişen fiyatıyla öne çıkan bir Lidar sensörüdür.

LeddarTech Leddar Vu8: Leddar Vu8, LeddarTech tarafından geliştirilen yüksek çözünürlüklü bir Lidar sensörüdür. 8 aktif algılama bölgesine sahip olup, kullandığımız model 75 metreye kadar algılama menzili sunmaktadır. Leddar Vu8, hafif ve kompakt bir tasarıma sahip olup, çeşitli uygulamalara kolayca entegre edilebilir. Aynı zamanda düşük enerji tüketimiyle verimlilik ve enerji tasarrufu sağlar.

TSL2591: TSL2591, geniş bir dinamik aralık sunan ve çeşitli aydınlatma

koşulları altında doğru ölçümler yapan bir ortam ışığı sensörüdür. Çok karanlık ortamlarda (örneğin, gece) ve çok aydınlık ortamlarda (örneğin, doğrudan güneş ışığı altında) bile hassas ölçümler yapabilen TSL2591, iki adet fotodiyot kullanır; biri görünür ve kızılötesi ışığı, diğeri ise sadece kızılötesi ışığı algılar. Bu çift fotodiyot tasarımı, ışığın spektral yanıtını hesaplayarak son derece hassas bir lux ölçümü sağlar.

5.2. Sensor Zaman Senkronizasyon

Araç, farklı türlerde sensörlerle donatılmış olup, her biri farklı hızlarda veri üretmektedir. Bu nedenle, veri kayıtlarını kaydetme esnasında sensörlerden gelen eşzamansız akışlar doğru bir şekilde senkronize edilmelidir. ROSbag dosyalarının zaman damgalarıyla senkronizasyon işlemi, ROS içinde veri kaydı ve oynatma sürecinde kritik bir rol oynar. ROS düğümleri (nodes) arasında mesaj alışverişi yapılırken, her mesaj bir zaman damgası (timestamp) ile etiketlenir. Bu zaman damgaları, mesajın oluşturulduğu veya yayımlandığı zamanı temsil eder ve ROS sistem saatine dayalıdır.

Rosbag kaydedici (rosvag record) çalıştırıldığında, belirli bir veya birden fazla konu (topic) üzerindeki tüm mesajlar zaman damgalarıyla birlikte kaydedilir. Bu zaman damgaları, mesajların kaydedildiği sıranın ve zamanlamanın korunmasını sağlar. Rosbag oynatıcı (rosvag play) kullanıldığında, ROSbag dosyasındaki mesajlar orijinal zaman damgalarına göre yeniden yayımlanır. Bu sayede sensör verileri, komutlar ve diğer mesajlar kaydedildiği zamandaki gibi doğru sırayla ve doğru zamanlamayla tekrar üretilir. Oynatma hızı ayarlanabilir olsa da, zaman damgaları arasındaki relatif zaman farkları korunur.

Veri setimizde, ROSbag dosyasında farklı frekanslarda veri toplayan Lidar, kamera ve IMU sensörleri bulunmaktadır. Oynatma sırasında her bir sensör verisi, kaydedildiği frekans ve zaman aralıklarına uygun şekilde yeniden yayımlanır. Bu mekanizma, robotik sistemlerin ve sensör verilerinin doğru bir şekilde kaydedilmesini ve analiz edilmesini sağlar ve verilerin doğru sırayla ve zamanlamayla yeniden üretilmesine olanak tanır. Bu özellik, robotik sistemlerin geliştirilmesi, test edilmesi ve hata ayıklanması sürecinde büyük bir avantaj sağlar.

5.3. Veri Dizileri

SLAM çalışmaları için kapalı ortam zorluklarını yansıtmayı hedeflediğimiz veri setinde, farklı karakterlere sahip birçok ortamda çeşitli veri dizileri oluşturulmuştur. Bu dizilerin tekrarlı olarak toplanmasının amacı, farklı zorlukların bulunduğu senaryoları sunarak algoritmaların hataya düştüğü durumları belirlemektir. Gerçek dünyayı daha iyi yansıtmak adına, daha fazla sayıda dizi, çeşitli güzergahlar ve daha uzun yollar belirlenmiş; ayrıca, algoritmanın olası yer tanıma ve döngü kapatma işlevlerini test etmek amacıyla, aynı güzergahın birden fazla kez geçildiği diziler oluşturulmuştur. Genellikle, daha büyük veri setleri, daha uzun yörüngeler ve çeşitli güzergahlar, sahneyi daha kapsamlı bir şekilde kapsar ve böylece algoritmaların zayıf yönlerini inceleme şansı artar.

Hedef sahneler, test ortamının türünü, koşulunu ve ölçeğini kapsar. Sahnenin türü, veri görünümünü ve yapısını belirler; bu da algoritmaların hareketleri tahmin etmek için güvendiği unsurlardır ve dolayısıyla performansı doğrudan etkiler. Örneğin, bir kamera tamamen beyaz bir duvarın önünde hareket ediyorsa (yani zayıf doku), bir görüntü tabanlı algoritmanın hareketi takip etmek için belirgin özellikler çıkarması imkansızdır. Benzer şekilde, bir Lidar tabanlı yöntem uzun bir koridorda (yani tekrarlayan yapı) çalışıyorsa, Lidar taramaları arasında eşleşme elde etmek oldukça zordur. Sahne türü ayrıca tekrarlayan doku, zayıf yapı ve su ve cam gibi yansıtıcı yüzeyler gibi diğer zorlayıcı unsurları da içerebilir. Sahne koşulları ise esas olarak aydınlatma, dinamik nesnelere, hareket bozulmaları, çok katlı yapılar ve zaman içerisindeki değişiklikleri içerir.

Topladığımız veri seti, birçok diziyi içeren ve toplamda yaklaşık 1TB'a ulaşan verilerden oluşmaktadır. GitHub sayfasında paylaştığımız veri setimizde bulunan diziler, özellikleriyle birlikte tablolar halinde sunulmuştur. Karşılaşılan zorluklar, 1 ile 5 arasında kolaydan zora doğru derecelendirilmiştir. Tablolar, toplanan her bir ortam için hazırlanmıştır. Dizilerin isimlendirilmesi, ardışık bir şekilde numaralandırılan her bir ortamı temsil eden bir sayıyı içerir (örneğin, '1', '2' vb.). Bunun yanı sıra, aynı ortam içinde toplanan farklı veri dizileri, desimal noktadan sonraki bir sayı ile gösterilmiştir. Örneğin; 'home1.0'. Ayrıca, aynı ortamda belirli bir zorluk için çalıştığımız diziler de ayrıca gruplandırılmıştır. Bu gruplara örnek olarak, Home2'de aynaların bulunduğu diziler ve

Akademik_Building1’de asansör ve merdivenlerin kullanıldığı özel diziler gösterilebilir.

Home1: İki yatak odası, bir çalışma odası ve bir oturma odasından oluşan bir dairedir. Home1 içerisinde toplanan verilerde, mobilya değişimleri, hareket kaynaklı ve ışık değişimlerinin ağırlıklı olduğu uzun süreli çalışmalar bulunmaktadır (Tablo 5.2). Şekil 5.3’de görüldüğü gibi, mobilya konum değişimleri ve perde ile mobilya renklerinde de değişimler bulunmaktadır.



Şekil 5.3. Zaman içerisinde gerçekleşen değişimlere ait görseller

Home2: Dört yatak odası ve iki oturma odasından oluşan, üç katlı bir evdir. Merdivenler ile katlar arasında geçişlerin yapıldığı, büyük aynalar nedeniyle yansımaların ve harekete bağlı bozulmaların olduğu diziler Tablo 5.3’de sunulmaktadır. Aynaların SLAM haritalama üzerindeki etkisinin incelenebilmesi için, Home2 içerisinde aynaların bulunduğu odada özel diziler oluşturulmuştur (5.4).

Academic_Building1: Laboratuvar ve sınıfların bulunduğu orta ölçekli bir binadır. Farklı katlar arasında asansör ve merdivenlerin kullanıldığı, aydınlatma şartlarının tamamen karanlık ile ışık parlamaları arasında değiştiği ve agresif hareketlerin ağırlıklı olduğu diziler Tablo 5.5’de sunulmuştur. Bu bina içerisinde asansör ve merdivenler ile hareketlerin bulunduğu özel dizi grubu içerisinde, bir katın gezilip asansör ile bir başka kata gidildiği diziler bulunmaktadır. Kullanılan asansörün panoramik olması ise ayrıca bir zorluk teşkil etmiştir. Katlar arasında merdivenlerin kullanıldığı diziler de Tablo 5.6’de yer almaktadır.



Şekil 5.4. Aynaların bulunduğu sahne

Academic_Building2: Büyük ölçekli, uzun dar koridorların ve camlı bölmelerin bulunduğu, aydınlatma koşullarının değişken olduğu ve dinamik engellerin görüldüğü diziler Tablo 5.7’de sunulmaktadır.

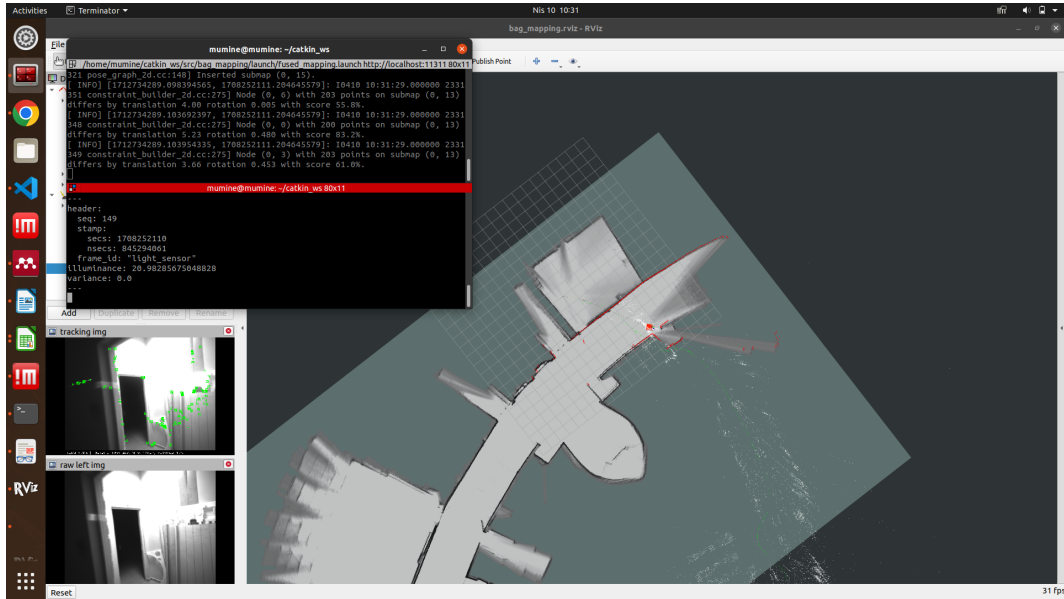
Academic_Building3: Camlı bölmelerle ayrılmış, açık konsept bir ofis olan bu bina içerisinde, agresif hareketler ve farklı aydınlatma şartları altında toplanmış dizilerden oluşan veri seti Tablo 5.8’te sunulmuştur.

Hospital1: Hastane giriş katında ve yürüyen merdivenlerle çıkılan üst kattaki dizilerden oluşmaktadır. Dinamik engellerin çok yoğun olduğu bir veri setidir.

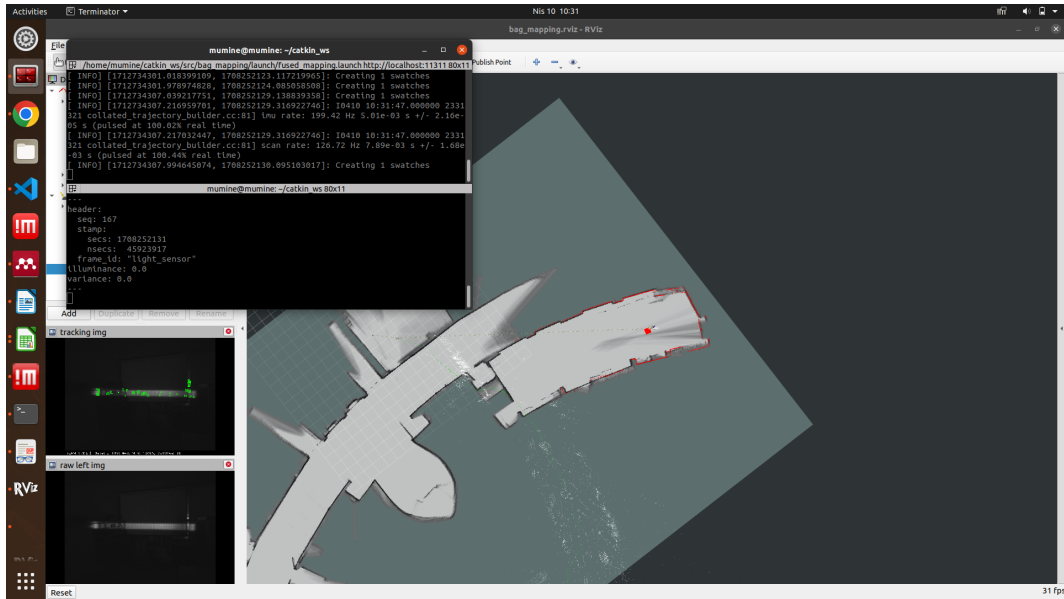
Store1: Tek dizilik bir veri setidir. Bir depo ve merdivenlerle inilen bir bodrum katında toplanan dizide, aydınlatma koşullarında değişim, özniteliksiz sahneler ve karmaşık yüzeyler ön plana çıkmaktadır.

Market1: Tek dizilik bir veri setidir. Bir market içinde, reyonlar etrafında agresif hareketlerle toplanmış dizilerden oluşmaktadır.

düştüğü tespit edildi. Bu tür hatalar, robotların gerçek dünyada karşılaşılabileceği yansıtıcı yüzeyler karşısında oldukça kritik olabilir.



Şekil 5.6. Parlama ve özneliksiz sahne nedeniyle yörünge kopması



Şekil 5.7. Aşırı karanlık sahne nedeniyle yörünge kopması

Akademik_building1.2 dizisinde yapılan testlerde, karanlık ve ışık parlamalarının olduğu özneliksiz ortamlarda ORB-SLAM3 algoritmasının yörünge ve haritayı kaybettiği gözlemlendi. Işık parlamasının olduğu bir ortamda, ışık sensöründen alınan değerin 30 lux olduğu ve duvarlara yakın hareket nedeniyle özneliklerin yetersiz kaldığı bir noktada kopma gerçekleşti (Şekil 5.6). Tamamen karanlık bir ortamda test edilen algoritma, lux değerinin 5 olduğu noktalara kadar

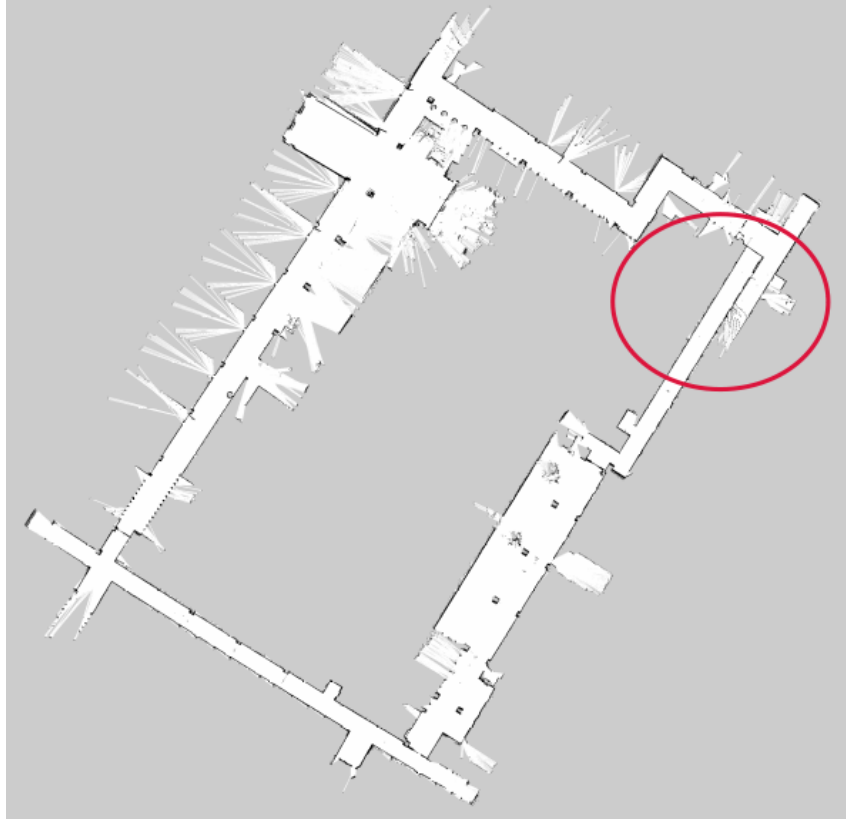
kızılötesi kameralar ve platformda bulunan Leddar sensörünün lazer dizisiyle haritalamaya devam edebildi, ancak lux değerinin 5'in altına düştüğü yerlerde haritalama başarısız oldu (Şekil 5.7).



Şekil 5.8. Akademik building2.1 ORB-SLAM3 ve Cartographer yörünge çıktıları

Akademik_building2.1 dizisinde, Cartographer ve ORB-SLAM3 algoritmalarıyla eş zamanlı olarak elde edilen harita ve yörüngelerde Cartographer algoritması yörüngeyi tamamlarken, ORB-SLAM3 algoritması başlangıçta özniteliksiz bir sahne nedeniyle bir süre yörüngeyi takip edemedi (Şekil 5.8).

Cartographer algoritmasının, dar ve uzun koridorlar gibi zorluklara karşı başarılı olmasının nedeni, bu tür alanlarda alt haritalar oluşturarak, haritalama



Şekil 5.9. Cartographer Akademik Building2.1 haritasında kayma hatası

esnasında optimizasyon yapabilmesidir. Şekil 5.9 de görüldüğü gibi, haritadaki kaymalardan dolayı başlangıç ve bitiş noktaları örtüşmezken, yapılan optimizasyon ile örtüşme sağlanmıştır. Ancak, sarsıntının fazla olduğu yerlerde ve çok katlı binalarda Cartographer algoritmasının da başarısız olduğu gözlemlenmiştir.

5.5. Tartışma

Bu çalışmada, çeşitli zorlu koşullarda SLAM algoritmalarının performansını değerlendirdik ve elde ettiğimiz sonuçlar, mevcut algoritmaların bazı kritik zayıflıklarını ortaya koydu. Özellikle, yansıtıcı yüzeylerin bulunduğu ortamlarda Cartographer algoritmasının aynadaki yansımaları yanlış bir şekilde haritaya eklediği, bu sırada ORB-SLAM3 algoritmasının ise yörünge hatasına düştüğü tespit edilmiştir. Bu sonuçlar, her iki algoritmanın da yansıtıcı yüzeylere karşı hassas olduğunu göstermektedir. Robotların gerçek dünyada sıkça karşılaşabileceği bu tür yüzeylere karşı güvenilir performans sergileyememeleri, potansiyel kullanım alanlarında ciddi kısıtlamalar yaratabilir.

Ayrıca, karanlık ve özniteliksiz ortamlarda yapılan testlerde, ORB-SLAM3

algoritmasının yörüngeyi ve haritayı kaybetmesi, ışık ve özniteliklerin algılama sistemleri üzerindeki etkisini vurgulamaktadır. Özellikle, düşük lux değerlerinde algoritmanın başarısız olması, ışık koşullarının yeterince iyi olmadığı ortamlarda bu tür algoritmaların güvenilirliğini sorgulatmaktadır.

Cartographer algoritması, dar koridorlar gibi Lidar tabanlı algoritmalar için zorlayıcı olan koşullarda başarılı bir performans sergilese de, sarsıntılı hareketler ve çok katlı binalarda benzer başarıyı gösterememiştir. Bu, algoritmanın optimizasyon yeteneklerine rağmen, bazı fiziksel ve yapısal zorluklar karşısında yeterince dayanıklı olmadığını göstermektedir.

Sonuç olarak, bu testler, SLAM algoritmalarının farklı zorluklar altında nasıl performans gösterdiğini ve hangi koşullar altında başarısız olduklarını anlamamıza yardımcı olmuştur. Gelecekteki çalışmalar, bu zayıflıkları ele almak ve daha sağlam ve esnek SLAM algoritmaları geliştirmek için yeni yaklaşımlar ve metrikler üzerinde odaklanmalıdır. Gerçek dünya SLAM uygulamalarında, bu tür hataların yaygın olduğu göz önünde bulundurularak, farklı zorluklar nedeniyle daha uygun metrikler geliştirilmelidir. Metrikler, SLAM algoritmalarının sahne anlama yeteneklerinin daha kapsamlı bir şekilde değerlendirilmesine olanak tanıyabilmelidir.

Çizelge 5.2. Home1 veri seti dizi özellikleri

İsim	Zorluklar	Boyut(GB)	Süre	Lazer Projector
Home1.0	M*, I**, TR** LL***(according to Home1.3,4,5)	11.2	145s	Off
Home1.1	M*,I**, TR** LL***(according to Home1.3,4,5)	11.6	149s	Off
Home1.2	M*,I**, F* TR** LL***(according to Home1.3,4,5)	9.3	119s	Off
Home1.3	M**,I*** F***, TR** LL***(according to Home1.0,1,2)	15,7	127s	On
Home1.4	M*****, I*****, F**, TR** LL*** (according to Home1.0,1,2)	12.5	101s	On
Home1.5	M*****,I*****, F***, TR** LL***(according to Home1.0,1,2)	14.9	121s	On

Çizelge 5.3. Home2 veri seti dizi özellikleri

İsim	Zorluklar	Boyut(GB)	Süre	Lazer Projector
Home2.0	M*, I**, D*, TR** LL**(according to Home2.4,5)	8.6	110s	Off
Home2.1	M*,I**, TR** LL**(according to Home2.4,5)	5.1	66s	Off
Home2.2	M****,I**, F*, MS***** TR**** LL**(according to Home2.4,5)	15.7	202s	Off
Home2.3	M****,I****, F*, MS*****, TR**** LL**(according to Home2.4,5)	9.5	123s	Off
Home2.4	M*****, I****, LL**(according to Home1.0,1,2,3)	10.2	82s	On
Home2.5	M*****,I*****, F***, MS*****, TR** LL**(according to Home1.0,1,2,3)	13.2	107s	On

Çizelge 5.4. Home2 ayna özel veri seti dizi özellikleri

İsim	Zorluklar	Boyut(GB)	Süre	Lazer Projector
Home2_mirror0	M**, I***, D*, TR*****	6.2	80s	Off
Home2_mirror1	M***, I*, TR*****	3.4	44s	Off
Home2_mirror2	M***, I*, TR*****	3.3	42s	Off

Çizelge 5.5. Academic_building1 veri seti dizi özellikleri

İsim	Zorluklar	Boyut(GB)	Süre	Lazer Projector
Academic_building1.0	M*, I**, F**, S** TR*	58.8	758s	Off
Academic_building1.1	M**, I*****, F**, S**, TR*	21.3	275s	Off
Academic_building1.2	M****, I**, F**, S***, TR*	39.3	319s	On
Academic_building1.3	M****, I*****, F**, S***, D***, TR*	29	235s	On
Academic_building1.4	M**, I***, F**, S**	17.1	221s	Off
Academic_building1.5	M**, I****, F**, S**, TR*	18.1	233s	Off

Çizelge 5.6. Academic_building1 çok katlı veri seti dizi özellikleri

İsim	Zorluklar	Boyut(GB)	Süre	Lazer Projector
Academic_building1_MS0	M*, I**, F**, S** TR*	58.8	758s	Off
Academic_building1_MS1	M**, I*****, F**, S**, TR*	21.3	275s	Off
Academic_building1_MS2	M****, I**, F**, S***, TR*	39.3	319s	On
Academic_building1_MS3	M****, I*****, F**, S***, D***, TR*	29	235s	On
Academic_building1_MS4	M**, I***, F**, S**	17.1	221s	Off
Academic_building1_MS5	M**, I****, F**, S**, TR*	18.1	233s	Off

Çizelge 5.7. Academic_building2 veri seti dizi özellikleri

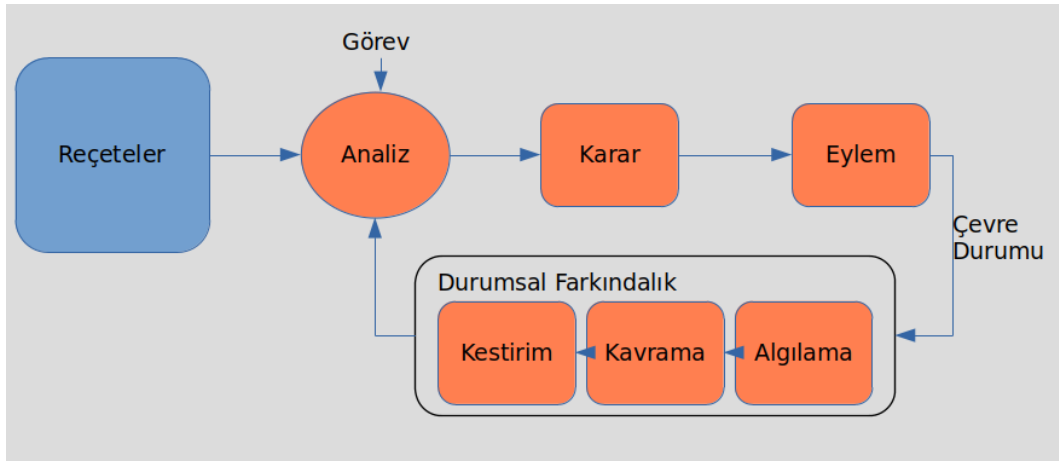
İsim	Zorluklar	Boyut(GB)	Süre	Lazer Projector
Academic_building2.0	M**, I***, F**, D**, TR****	45.7	589s	Off
Academic_building2.1	M***, I****, F**, D*, TR****	55.3	713s	Off
Academic_building2.2	M***, I****, F**, S*, TR***	63.1	814s	Off
Academic_building2.3	M****, I****, F**, D***, TR****	27.3	222s	On
Academic_building2.4	M****, I****, F**, S*, D****, TR****	35.7	290s	On
Academic_building2.5	M****, I****, F*, D****, TR**	41.4	337s	On
Academic_building2 _ground_floor	M**, I****, F**, S****, TR****	13.6	226s	Off
Academic_building2 _section	M****, I****, F****, TR*	14.5	186s	Off

Çizelge 5.8. Farklı alanlarda toplanan veri seti dizi özellikleri

İsim	Zorluklar	Boyut(GB)	Süre	Lazer Projector
Academic_building3.0	M****, I***, C** TR*****	23.6	192s	On
Academic_building3.1	M****, I****, C**, TR*****	15.1	122s	On
Hospital1.0	M****, I***, F***, S**, D****, TR****	25.1	204s	On
Hospital1.1	M****, I***, F**, S**, D****, TR***, MS*****	27.3	222s	On
Store1.0	M****, I****, F****, S****, C****, TR**, MS****	25.7	215s	On
Market1.0	M****, I**, S****, C***, D**	16.4	134s	On

6. ADAPTİF SLAM UYGULAMASI

Adaptif SLAM karar modelinin etkinliğini test etmek için uzun süreli bellek yapısında sadeleştirmeye gidilerek yüklenen hazır reçeteler ve çalışma belleği işlemleri gerçekleştirilmiştir.



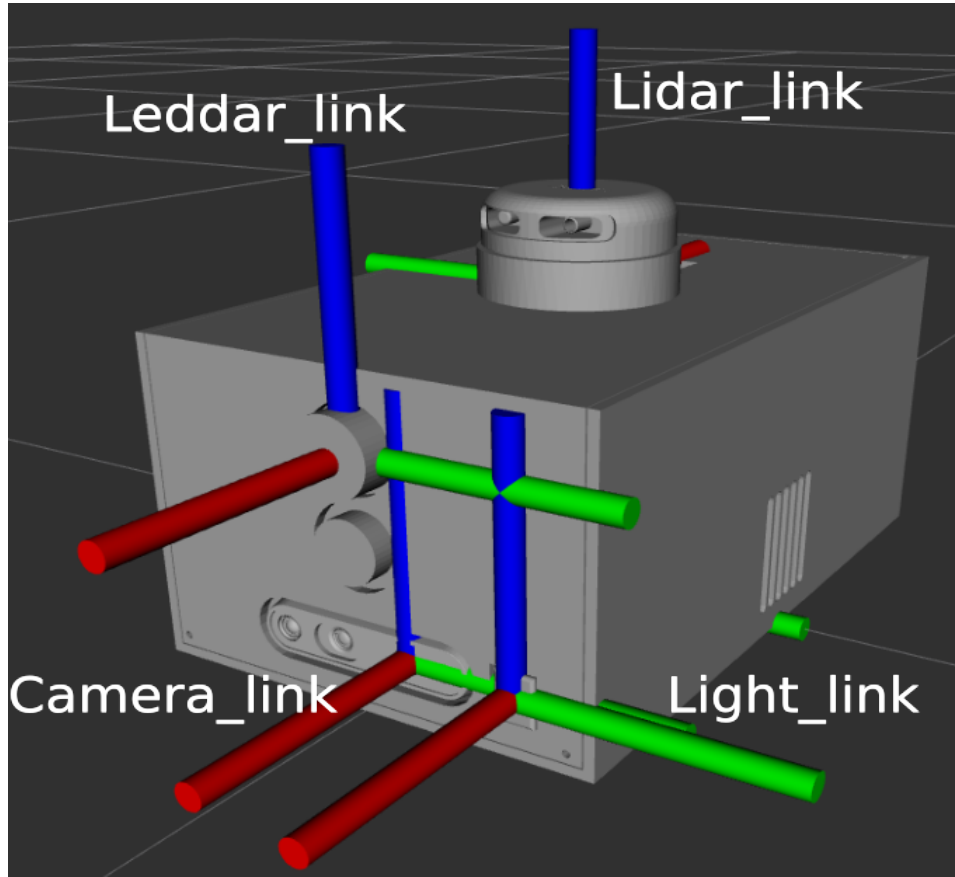
Şekil 6.1. Sadeleştirilmiş Adaptif SLAM Karar Modeli

6.1. Durumsal Farkındalık

Bu uygulamada, durumsal farkındalık kapsamında ortamın aydınlatma durumu ve Görsel-SLAM algoritmasının özellik çıkarımları üzerine çalışmalar yapılmıştır.

6.1.1. Algılama

Algılama aşamasında sensör takımı olarak 2D Lidar, RGB-D ve infrared stereo özelliklerine sahip kamera ve kamerada dahili olarak bulunan IMU sensörleri kullanılmıştır. Bu sensörlere ek olarak, ortamın aydınlatma durumunu değerlendirmek amacıyla bir ışık sensörü de kullanılmıştır.



Şekil 6.2. Robot modeli ve sensör bağlantıları

6.1.2. Kavrama

Kavrama aşamasında ORB-SLAM3 algoritması ile Görsel-SLAM, Hector ve Cartographer algoritmaları ile Lidar-SLAM tabanlı harita ve yörünge bilgileri elde edilmiştir. Ayrıca, ışık sensöründen elde edilen veriler değerlendirilerek ortamın aydınlatma durumu hakkında çıkarım yapılmıştır. ORB-SLAM3 algoritmasının çıkardığı özellik sayısı incelenerek, özneliksiz sahneler hakkında da çıkarımlar yapılmıştır.

Desen oluşturma işlemleri için ortamın aydınlatma durumuna dair farkındalık sağlayan l değişkeni tanımlanmıştır. l değişkeni, ışık sensöründen okunan L lux değerinin 1-5 aralığında ölçeklendirilmesi ile elde edilmiştir.

Ölçeklendirme işlemi için;

$$l = 1 + \left(\frac{L - L_{\min}}{L_{\max} - L_{\min}} \right) \times (5 - 1)$$

denklemini kullanılmıştır. Burada $L_{\min} = 0$ ve $L_{\max} = 500$ olarak alındığında formül

şu şekilde sadeleşir:

$$\begin{aligned}
 l &= 1 + \left(\frac{L - 0}{500 - 0} \right) \times (5 - 1) \\
 &= 1 + \frac{L}{500} \times 4 \\
 &= 1 + 0.008L
 \end{aligned}$$

ORB-SLAM3 algoritması, yörünge çıkarma ve haritalama işlemleri için özellik çıkarımı gerçekleştirir. /num_tracked_key_points adında yayınlanan veri, SLAM sisteminde aktif olarak izlenen anahtar noktaların sayısını yansıtır ve bu sayı, köşe belirleme, anahtar nokta takibi ve eşleme süreçlerinin sonucunda elde edilir. Anahtar nokta sayısı kullanılarak algoritmanın özellik çıkarımında başarısız olduğu durumlar tespit edilir.

Anahtar noktasının yeterliliğinin belirlenebilmesi için farklı ortamlarda toplanmış veri setleri ile 50'nin üzerinde test yapılmıştır. Gözlemlenen değerler şu şekildedir:

- Anahtar nokta sayısı yaklaşık olarak 45 ve altında ise harita ve yörünge kaybı yaşanır.
- Anahtar nokta sayısının 115 ve altına düştüğü durumlarda hata oranı artar.
- Değerin 115 ile 190 arasında olduğu durumlarda haritalama ve yörünge takibi için yeterli olur.
- Değerin 190 ile 250 arasında olduğu durumlarda ise iyi sonuçlar elde edilir.
- 250'nin üzerinde ise hatanın sıfıra yakın olduğu çok iyi sonuçlar gözlemlenmiştir.

Bu gözlemler dikkate alınarak ölçeklendirme yapılmış ve f değeri elde edilmiştir.

$$f(x) = \begin{cases} 1 & \text{eğer } x \leq 45 \\ 2 + \frac{x-45}{115-45} \cdot (3 - 2) & \text{eğer } 45 < x \leq 115 \\ 3 + \frac{x-115}{190-115} \cdot (4 - 3) & \text{eğer } 115 < x \leq 190 \\ 4 + \frac{x-190}{250-190} \cdot (5 - 4) & \text{eğer } 190 < x \leq 250 \\ 5 & \text{eğer } x > 250 \end{cases}$$

Burada:

- x gözlemlenen özellik sayısını ifade eder.
- $f(x)$ ise ölçeklendirilmiş değerdir.

Aydınlatma şartları ve özellik çıkarımları, elde edilen odometri verisinden x ve y değerleri ile birlikte matrise kaydedilerek desen oluşturulur.

$$D = \begin{pmatrix} x_1 & y_1 & l_1 & f_1 \\ x_2 & y_2 & l_2 & f_2 \\ x_3 & y_3 & l_3 & f_3 \\ x_4 & y_4 & l_4 & f_4 \end{pmatrix}$$

6.1.3. Kestirim

Kavrama aşamasında elde edilen desen matrisi incelenerek, daha önce tanımlanmış olan sözlükler yardımıyla bir kestirim dizisi oluşturulur.

$$\text{Aydınlatma Durumu} = \begin{cases} \text{Karanlık}=1 & \text{eğer en az 3 tane } l_i < 2 \text{ ise,} \\ \text{Loş}=2 & \text{eğer en az 3 tane } 2 \leq l_i \leq 3 \text{ ise,} \\ \text{İdeal}=3 & \text{eğer en az 3 tane } 3 \leq l_i \leq 4 \text{ ise,} \\ \text{Parlak}=4 & \text{eğer en az 3 tane } l_i > 4 \text{ ise.} \end{cases}$$

$$\text{Özellik Çıkarımı} = \begin{cases} \text{Özellik yok}=1 & \text{eğer tüm elemanlar } f_i < 2 \text{ ise,} \\ \text{Yetersiz}=2 & \text{eğer tüm elemanlar } 2 \leq f_i \leq 3 \text{ ise,} \\ \text{İdeal}=3 & \text{eğer tüm elemanlar } f_i > 3 \text{ ise,} \end{cases}$$

Aydınlatma durumu ve özellik çıkarımına dair kestirimler, tanımlanmış sözlükler yardımıyla belirlenir. Bu kestirimler, bir dizi şeklinde çıktı olarak üretilir.

$$\text{Kestirim} = \left[\text{Aydınlatma Durumu} \quad \text{Özellik Çıkarımı} \right]$$

Burada:

- x , gözlemlenen özellik sayısını ifade eder.
- $f(x)$ ise ölçeklendirilmiş değerdir.

Aydınlatma şartları ve özellik çıkarımları, elde edilen odometri verisinden x ve y değerleri ile birlikte matrise kaydedilerek desen oluşturulur.

6.2. Reçeteler

Reçeteler bölümünde, farklı durumlar için oluşturulmuş işlem adımlarının belirlendiği, hızlıca uygulanabilir işlemler bulunmaktadır. Bu reçeteler, durumlara göre kategorize edilerek mevcut duruma en yakın reçetelere kolayca erişim sağlanmasını mümkün kılar.

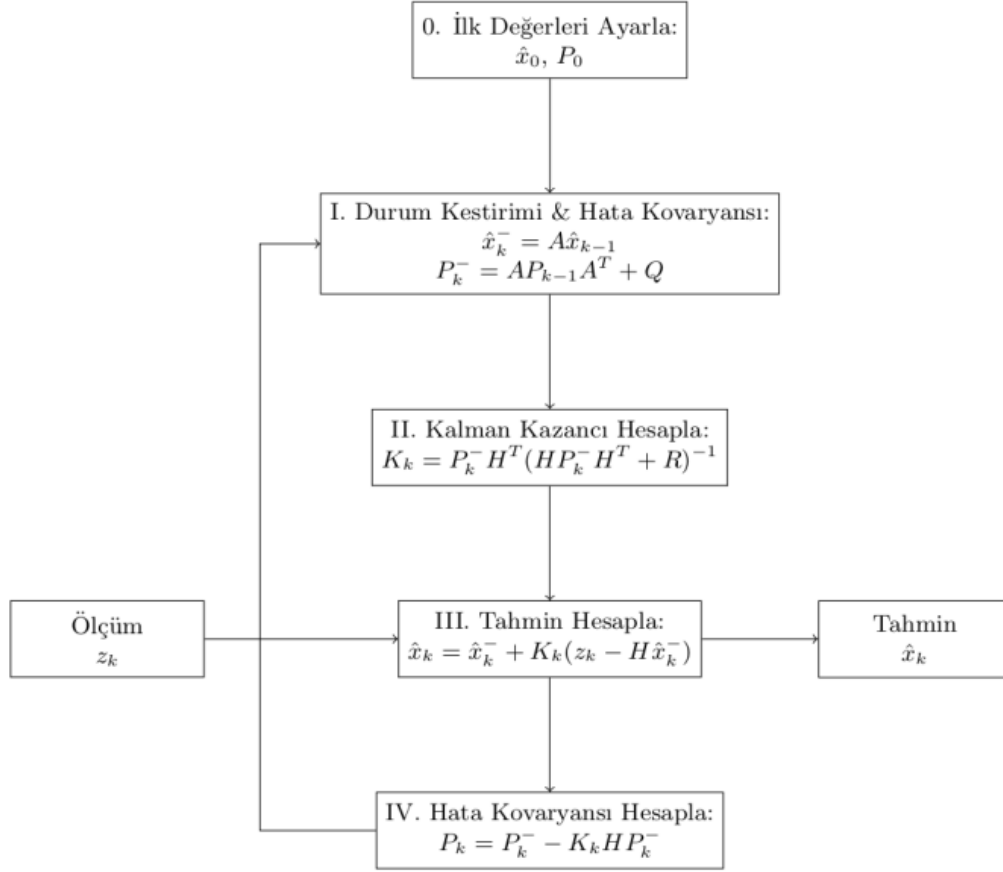
Test işlemleri için Reçeteler bölümüne, Kalman Filtreleri üzerinden üretilen farklı çözümler reçeteler olarak kaydedilmiştir. Genişletilmiş Kalman Filtresi (EKF) ile Lidar ve Görsel-SLAM bilgilerinin füzyon edilerek filtrelenmiş füzyon odometri bilgisinin üretildiği reçete, bunlardan biridir. Bir diğer reçete ise, görsel sensörlerin güvenilirliğinin düştüğü noktalarda görsel sensör verilerinin filtreleme işlemlerine dahil edilmediği ve Lidar verisine güvenildiği EKF konfigürasyonuna sahip reçetedir. Bu reçetede, bozuk verinin sisteme eklenmesi engellenerek daha doğru sonuçlara ulaşılması hedeflenmiştir.

Reçetelerin kavranması ve sisteme uyarlanması için anlaşılabilirliği için öncelikle Kalman Filtreleri açıklanacak, ardından Genişletilmiş Kalman Filtreleri ele alınacaktır. Sistemimiz için gerekli sistem modeli ve ölçüm modeli çıkarımları yapılmıştır. Sonrasında, ROS üzerinde algoritma yazılımının yapıldığı reçeteler için gerekli konfigürasyonlar açıklanmıştır.

6.2.1. Kalman Filtresi

Kalman filtresi, beyaz gürültü ile bozulmuş doğrusal dinamik bir sistemin anlık durumunu tahmin eden bir tahmin edicidir. Bu tahmin edici, tahmin hatasının kuadratik fonksiyonları açısından istatistiksel olarak optimaldir ve istatistiksel tahmin teorisinin en büyük keşiflerinden biri olarak kabul edilir. Kalman filtresi, sürekli üretim süreçleri, uçaklar, gemiler ve uzay araçları gibi dinamik sistemlerin kontrolünde kritik rol oynar. Özellikle tüm değişkenlerin doğrudan ölçülemediği veya istenmediği durumlarda, eksik bilgiyi gürültülü ölçümlerden çıkarmayı sağlar. Ayrıca, dinamik sistemlerin gelecekteki gelişmelerini tahmin etmek için de kullanılır (Welch vd., 1995). Kalman algoritmalarının anlatımında Kim ve Huh (2011) kitabından faydalanılmıştır. Dört adımlı hesaplama prosedüründen oluşan Kalman filtresi, z_k ölçümlerini alarak \hat{x}_k tahminini hesaplar. Algoritmanın hesaplama

prosedürü detaylıca incelendiğinde, öncelikle kestirim aşamasında kullanılan iki değişken, \hat{x}_k^- ve P_k^- hesaplanmaktadır. Üst simge '-' tahmin edilen değeri ifade eder. Bu değişkenler, IV. adımda Hata Kovaryansı hesaplanırken kullanılacaktır.



Şekil 6.3. Kalman Algoritması

Adım II'de, Kalman Kazancı K_k hesaplanır. P_k^- bir önceki adımda hesaplanır. H ve R değerleri ise Kalman filtresi dışında verilen değerlerdir.

Adım III'de, verilen z_k değeri ile tahmin hesaplanır. \hat{x}_k^- değeri adım I'de hesaplanır.

Adım I'de hesaplanan P_k^- hata kovaryansı, önceki adımın ne kadar doğru yapıldığını belirten bir ölçüdür ve genellikle tahmin edilen değere güvenip güvenmeme kararı verirken kullanılır.

- z_k : Ölçümler (Harici giriş)
- \hat{x}_k : Tahmin (Son çıktı)
- A, H, Q, R : Sistem Modeli
- $\hat{x}_k^-, P_k^-, P_k, K_k$: Dahili hesaplama değişkenleri

Sistem modelini belirleyen A , H , Q ve R deęişkenleri, Kalman filtresini uygulamadan önce belirlenmelidir. Başka bir deyişle, bu deęişkenler Kalman filtresi tarafından hesaplanmaz veya varsayılmaz. Bu deęerler, sistemin özellikleri ve Kalman filtresinin amacı doğrultusunda kullanıcı tarafından önceden ayarlanmış deęerlerdir. Kalman filtresinin performansı bu deęerlerle yakından ilişkilidir. Sistem modeli dışındaki deęişkenler Kalman Filtresi tarafından hesaplanır.

Yukarıdaki tabloda, 'Sistem modeli' olarak kategorize edilenler hariç tüm deęişkenler tasarımcı tarafından keyfi olarak deęiştirilemez çünkü bunlar algoritma tarafından ölçülür veya hesaplanır. Bu nedenle, tasarlanan Kalman filtresinin performansı tatmin edici olmadığında, ayarlanabilen tek deęişkenler sistem modellemesiyle ilgili dört deęişkendir. Başka bir deyişle, sistem modelinin dört deęişkeni tasarım faktörleridir. Kalman filtresinin performansı bunlar tarafından belirlenecektir. Referans olarak, Kalman filtresinin performansı, sistem modelinizin modellediğiniz gerçek sisteme yaklaştıkça daha da artar. Şekil 6.3'de gösterilen Kalman filtresi algoritması dört adıma bölünmüştür, ancak anlama göre bölersek iki parçaya ayrılır: Kestirim süreci ve Tahmin süreci.

Kestirim Süreci: Şekil 6.3'in I. Adımı bu sürece dahildir. Bir önceki zaman noktasından elde edilen tahmin ve hata kovaryansı (\hat{x}_{k-1}^- ve P_{k-1}^-) bu sürece girdi olarak kullanılır ve mevcut zaman noktasındaki kestirim (\hat{x}_k^- ve P_k^-) nihai sonuç olarak elde edilir. Bu deęerler, tahmin süreci için kullanılır. Kestirim sürecinde kullanılan sistem modeli parametreleri A ve Q 'dur.

Tahmin Süreci: Kalman filtresi algoritmasında, II., III. ve IV. adımlar bu sürece dahildir. Tahmin sürecinin sonuçları tahmin (\hat{x}_k) ve hata kovaryansı (P_k) deęerleridir. Bu süreçte, ölçüm (z_k) de kestirim sürecinden gelen kestirime (\hat{x}_k^- ve P_k^-) ek olarak girdi olarak kullanılır. Tahmin sürecinde kullanılan sistem modeli deęişkenleri H ve R 'dir.

Bu perspektiften bakıldığında, Kalman filtresi şu şekilde özetlenebilir:

1. Sistem modeline (A ve Q) dayanarak bir sonraki zaman noktası için durum ve hata kovaryansı kestirimi yap: \hat{x}_k^- ve P_k^- .
2. Ölçüm ile tahmin arasındaki farkı telafi et ve yeni tahmini hesapla. Bu tahmin, Kalman filtresinin nihai sonucudur: \hat{x}_k ve P_k .
3. Yukarıdaki iki adımı tekrarla.

Tahmin Süreci

İlk olarak, tahmin süreci incelendiğinde; III. adımdaki tahmin hesaplama denklemi, 1. dereceden düşük geçişli filtre ile çok benzerdir ve iki filtre arasında önemli bir fark vardır.

Kalman filtresinin yeni bir tahmin hesaplamak için kullanılan denklem 6.9:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (6.1)$$

Burada z_k , ölçümler; \hat{x}_k ise kestirim anlamına gelir.

1. dereceden düşük geçişli filtrenin denklemi 6.2:

$$\bar{x}_k = \alpha \bar{x}_{k-1} + (1 - \alpha) x_k \quad (6.2)$$

Bu ifade, Kalman filtresi tahmin hesaplaması için kullanılan denklem 6.9 ile karşılaştırıldığında: Birinci dereceden düşük geçiş filtresi, önceki tahmin ve ölçümün ağırlıklı değerlerini ekleyerek bir tahmin hesaplar. Kalman filtresi de benzer şekilde çalışır. Ağırlıklandırma yöntemleri benzerdir; ancak aralarındaki temel fark, Kalman filtresinin tahmin güncellemelerini doğrudan ölçümlere dayandırması ve önceki tahmini bu güncellemede doğrudan kullanmamasıdır.

Öncelikle, bir tahmin (\hat{x}_k^-) ve yeni bir ölçüm (z_k) gereklidir. Kalman filtresinin, düşük geçiş filtresi gibi, özyinelemeli bir filtredir. Ölçüm sisteme bir giriş olarak verildiği için bilinen bir değerdir. Tahmin değeri ise, bir sonraki bölümde tartışılacak tahmin sürecinin sonucudur, yani ilerleyen aşamalarda belirlenecektir. Sistem modeli ile ilgili bir matris olan H matrisinin nasıl kullanılacağı daha sonra ele alınacaktır. Geriye kalan tek bilinmeyen ise K_k 'dir. 'Kalman kazancı' olarak adlandırılan bu değişkenin değeri belirlendiğinde, yeni tahmin hesaplanabilir.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (6.3)$$

Adım II'de, Denklem 6.3 ile elde edilen Kalman kazancı, Adım III'te tahmin hesaplamada ağırlık olarak kullanılır.

Birinci dereceden düşük geçiş filtresinde, tahmin hesaplamada uygulanan ağırlık (α) sabittir. Sistematik bir şekilde hesaplanmamış, tasarımcı tarafından keyfi olarak atanmıştır. Ancak Kalman filtresinde ağırlık K_k her döngüde hesaplanır. Başka bir deyişle, tahmin hesaplamada kullanılan ağırlık her adımda ayarlanır. Bu, Kalman filtresinin düşük geçiş filtresinden önemli bir farkıdır.

IV. Adımda Hata kovaryansı Denklem 6.4'de yazıldığı gibi hesaplanır ve sonucu tahmin sürecine iletilir. Burada önemli olan, hata kovaryansının anlamıdır.

Hata kovaryansı, Kalman filtresinden elde edilen tahmin ile gerçek ancak bilinmeyen değer arasındaki farkı gösterir. Başka bir deyişle, hata kovaryansı, tahminin doğruluk derecesini ifade eder. Eğer P_k büyükse, tahminin hatası da büyüktür; eğer P_k küçükse, tahminin hatası küçüktür. Bu nedenle, Kalman filtresi uygulanırken, tahminle birlikte hata kovaryansını da göstermek oldukça yaygındır.

$$P_k = P_k^- - K_k H P_k^- \quad (6.4)$$

x_k , tahmini \hat{x}_k ve hata kovaryansı P_k arasında aşağıdaki gibi bir ilişki vardır.

$$x_k \sim N(\hat{x}_k, P_k) \quad (6.5)$$

Bu, temel olarak x_k değişkeninin, ortalaması \hat{x}_k ve kovaryansı P_k olan normal bir dağılıma uyduğunu ifade eder, ancak bu ifadenin çok daha derin bir anlamı vardır. Yani, Kalman filtresi, x_k değişkeninin tahmininin olasılık dağılımını hesaplar ve en yüksek olasılığa sahip olanı tahmin olarak seçer.

Normal dağılım, x_k 'nin sahip olabileceği değerlerin olasılığını çizersek, bunun çan şeklinde bir dağılım olarak ortaya çıkacağı anlamına gelir. Bu çan şeklindeki eğrinin genişliği P_k tarafından belirlenir.

Normal dağılım eğrisinde, 'çan'ın genişliği küçükse, x_k 'nin sahip olabileceği değerler çoğunlukla ortalama etrafında yer alacaktır. Burada, ortalama tahmin olduğundan, hata doğal olarak küçük olur. Öte yandan, eğer 'çan' genişlerse, x_k için olası değerlerin aralığı da genişler ve bu da tahmin hatasını artırır. Bu, Kalman filtresinin uygulanması açısından pek yararlı olmayabilir. Hata kovaryansı:

$$P_k = E \{ (x_k - \hat{x}_k)(x_k - \hat{x}_k)^T \} \quad (6.6)$$

Burada $E\{\cdot\}$ operatörü, süslü parantez içindeki değişkenin beklentisini hesaplar. Sağ tarafta yer alan $x_k - \hat{x}_k$, gerçek ancak bilinmeyen değer ile tahmin arasındaki farkı, yani tahmin hatasını ifade eder. Başka bir deyişle, hata kovaryansı, tahmin hatasının karesinin ortalamasıdır. Bu nedenle, hata kovaryansı tahmin hatası ile orantılıdır.

Kestirim Süreci

Kestirim prosedürü, Şekil 6.3'deki Adım I'e karşılık gelir. Bu süreç, tahmin sürecinden çok daha basittir. Kestirim prosedüründe, zaman t_k 'dan t_{k+1} 'e geçildiğinde tahminin \hat{x}_k 'in nasıl değişeceği öngörülür. Başka bir deyişle, mevcut zaman noktasındaki kestirimin, bir sonraki zaman noktası t_{k+1} 'de ne değer alacağı tahmin edilir.

Adım I'deki ifadeler, Denklem 6.7, durumu tahmin ederken, Denklem 6.8 hata kovaryansını tahmin eder.

$$\hat{x}_{k+1}^- = A\hat{x}_k \quad (6.7)$$

$$P_{k+1}^- = AP_kA^T + Q \quad (6.8)$$

Burada, \hat{x}_k ve P_k , sırasıyla Adım III ve IV'ten hesaplanan değerlerdir. A ve Q ise sistem modeli tarafından önceden tanımlanmıştır. H ve R gibi diğer sistem modeli değişkenleri burada kullanılmaz; bu değişkenler tahmin sürecinde kullanılır. Kestirim için kullanılan notasyona dikkat edildiğinde; alt simge ' $k + 1$ ', t_{k+1} zaman noktasındaki değeri göstermek için eklenir ve üst simge '-' ise bunun bir tahmin olduğunu belirtir. Bu üst simge, aynı zaman noktasındaki kestirim ile tahmini ayırt etmek için kullanılır. Kestirim ve hata kovaryansı dışındaki değişkenler için bu tür bir ayırıma gerek yoktur, çünkü onlarda kestirim yapılmamaktadır.

- \hat{x}_k : Durum değişkeninin tahmini
- \hat{x}_{k+1}^- : Durum değişkeninin kestirimi
- P_k : Hata kovaryansının tahmini
- P_{k+1}^- : Hata kovaryansının kestirimi

Kestirim adımı için formüller nispeten basittir, ancak Kalman filtresinin performansını büyük ölçüde etkiler.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (6.9)$$

Bu denklemde, $H\hat{x}_k^-$ ifadesi 'tahminle hesaplanan ölçüm' anlamına gelir. Diğer bir deyişle, bu terim ölçümün tahminidir. Bu durumda, $z_k - H\hat{x}_k^-$ ifadesi, gerçek ölçüm ile tahmin edilen ölçüm arasındaki farkı, yani ölçümün tahmin hatasını temsil eder. Bu analize dayanarak, Kalman filtresinin tahmini ölçüm hatasını uygun bir şekilde kalibre ederek nihai kestirimi hesapladığını söyleyebiliriz. Burada, Kalman kazancı (K_k), tahminin ne kadar kalibre edilmesi gerektiğini belirleyen faktördür.

Kestirimin performansını en çok etkileyen faktör, tahminin doğruluğudur. Tahminin doğruluğu yoksa, Kalman kazancı ne kadar iyi ayarlanmış olursa olsun, kestirim hatalı olacaktır. Denklem 6.7 ve 6.8'de gösterilen hesaplamalar, sistem modelinden A ve Q matrislerinin kullanıldığını gösterir. Bu iki matris (A ve Q),

tahminin sonucunu belirleyen kritik öneme sahiptir. Eğer bu iki matris gerçek sistemden çok farklıysa, tahmin yanlış olur ve kestirim de hatalı olur. Tahminin doğruluğu, sistem modelinin gerçek sisteme ne kadar uygun olduğuna bağlıdır ve kestirim performansı büyük ölçüde bu tahminin kalitesine dayanır. Kalman filtresinin performansının, sistem modelinin doğruluğuna doğrudan bağlı olduğunu söylemek yerinde olacaktır. Bu nedenle, sistem modelinin her zaman kritik bir öneme sahip olduğunu vurgulamak gerekir.

Sistem Modeli

”Sistem modeli” terimi, ”problemin matematiksel olarak ifade edilmesi” anlamına gelir. Verilen bir sistem modelini anlamak genellikle zor değildir. Eğer sistem modeli zaten biliniyorsa, Kalman filtresini tasarlamak ve uygulamak da zor değildir. Ancak, sistemi matematiksel olarak modellemek ve sistem modelini oluşturmak zor bir görevdir. Bununla beraber, Kalman filtresi geniş bir uygulama alanında kullanılmaktadır ve bu da çoğu durumda referans olarak kullanılabilen bir sistem modeli bulmayı kolaylaştırır.

Kalman filtresi, genellikle doğrusal bir sistem modeli ile ilgilenir. Bu model aşağıdaki gibi ifade edilir:

$$x_{k+1} = Ax_k + w_k \quad (6.10)$$

$$z_k = Hx_k + v_k \quad (6.11)$$

Burada:

- x_k : Durum değişkeni, $(n \times 1)$ boyutunda sütun vektörü.
- z_k : Ölçüm, $(m \times 1)$ boyutunda sütun vektörü.
- A : Durum geçiş matrisi, $(n \times n)$ boyutunda matris.
- H : Durumdan ölçüme geçiş matrisi, $(m \times n)$ boyutunda matris.
- w_k : Durum geçiş gürültüsü, $(n \times 1)$ boyutunda sütun vektörü.
- v_k : Ölçüm gürültüsü, $(m \times 1)$ boyutunda sütun vektörü.

Durum değişkeni, konum, hız, ağırlık gibi ilgilenilen fiziksel bir büyüklüktür. Kalman filtresinde gürültü, önemli bir rol oynar ve tüm gürültülerin beyaz gürültü olduğu varsayılır. w_k , sisteme giren ve durum değişkenini etkileyen gürültüdür; v_k ise sensör tarafından ölçülen gürültüdür. Hem A hem de H matrislerinin tüm elemanları

sabittir. A matrisi, sistemin zamanla nasıl değiştiğini açıklar; yani sistemin hareket denklemlerini içerir. H matrisi ise ölçüm ile durum değişkeni arasındaki ilişkiyi gösterir ve her bir durum değişkeninin ölçüme nasıl dönüştürüldüğünü tanımlar.

1. mertebeden fark denklemleri (veya diferansiyel denklemler) ile ifade edilen hareket denklemlerinin bir modeli, Denklem 6.10 ve Denklem 6.11 gibi, 'durum uzayı modeli' olarak adlandırılır.

Kalman filtresi algoritmasının sistem modeli ile ilgili kısımlarına bakıldığında iki önemli adım vardır. Birincisi, tahminin hesaplanması için kullanılan adım I'deki ifadedir:

$$\hat{x}_{k+1}^- = A\hat{x}_k \quad (6.12)$$

Bu ifade, sistem modelinin bir parçası olan Denklem 6.10'den türetilmiştir. Diğer kısım ise adım III'teki tahmin güncellemesi sırasında kullanılan ifadedir:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (6.13)$$

Bu terim, sistem modelinin Denklem 6.11 ile ilgilidir. Denklemler 6.10 ve 6.11 ile sistem modelini karşılaştırdığımızda, tek farkın gürültülerin (w_k ve v_k) olmaması olduğunu görmek kolaydır. H matrisi başka yerlerde de kullanılır, ancak bunlar ek bir açıklama gerektirecek öneme sahip değildir.

Gürültü Kovaryansı

Genel olarak, gürültü, öngörülemeyen ancak sadece istatistiksel olarak tahmin edilebilen bir değerdir. Bu nedenle, gürültüyü ifade ederken istatistikler kullanılır. Kalman filtresinde, gürültünün normal dağılımda olduğu varsayıldığı için bilinmesi gereken tek şey gürültünün varyansıdır ve bu da ortalamamın her zaman sıfır olduğu anlamına gelir. Kalman filtresindeki durum modelinin gürültüsü, aşağıdaki kovaryans matrisi ile ifade edilir:

- Q : w_k 'nin kovaryans matrisi, $(n \times n)$ boyutunda diyagonal matris.
- R : v_k 'nin kovaryans matrisi, $(m \times m)$ boyutunda diyagonal matris.

'Diyagonal matris' terimi, tüm köşegen dışı elemanları sıfır olan bir matrisi ifade eder. (Köşegen elemanlar, indeksi $(1, 1), (2, 2), \dots, (n, n)$ olan elemanlardır.) Kovaryans matrisi, değişkenin varyansından oluşan bir matris olarak tanımlanır. Örneğin, eğer n adet gürültü w_1, w_2, \dots, w_n ve her bir değişkenin varyansı sırasıyla $\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2$ ise, kovaryans matrisi aşağıdaki gibi yazılabilir. Ölçüm v_k 'nin

kovaryans matrisi R de aynı şekilde ifade edilir.

$$Q = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{bmatrix}$$

Q ve R matrislerini doğru bir şekilde tanımlamak arzu edilir, ancak bunu analitik olarak yapmak genellikle zordur çünkü çeşitli hata kaynakları vardır. Bu nedenle, ilgili sistemde deneyime dayalı olarak deneme-yanılma yoluyla ayarlamalar yapmak kaçınılmazdır. Bu iki matris, Kalman filtresinin tasarım faktörleridir ve uygun değerler, gürültü hakkında elde edilen bilgilerin en iyi şekilde kullanılması amacıyla deneme yanılma yoluyla kalibre edilmelidir.

Q ve R matrislerinin Şekil 6.3’de nasıl kullanıldığına bakıldığında; ilk olarak, R matrisi Kalman kazancı için olan formülde yer alır:

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

Bu tür bir formülü matris formunda anlamak zor olabilir. Eğer tüm değişkenlerin skaler olduğunu varsayarsak, yukarıdaki denklemde bir matrisin tersini alma işlemi basit bir bölme işlemine eşdeğer olur. Böylece denklemi şu şekilde yeniden yazabiliriz:

$$K_k = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (6.14)$$

Bu ifadede, Kalman kazancı R arttıkça azalır. Kalman kazancı azaldığında, tahminin ölçüme katkısı da azalır. Diğer yandan, tahminin etkisi artar. Başka bir deyişle, tahminin varyasyonu azalır çünkü tahmin, dış ölçümden daha az etkilenir. Bu nedenle, ölçümden daha az etkilenmek ve daha az varyasyonlu bir tahmin elde etmek isteniyorsa, R artırılır.

Şimdi Q matrisi ile ilgili olan bir ifadeye bakalım. Bu, hata kovaryansının tahminini elde etmek için kullanılır (P_{k+1}^-):

$$P_{k+1}^- = A P_k^- A^T + Q$$

Q artırıldığında, hata kovaryansı P_{k+1}^- artar. Bunun sonucunda, Kalman kazancı (Denklem 6.14) da artar ve ölçümün tahmin üzerindeki etkisi artar. Bu nedenle, ölçümden daha az etkilenmek ve daha az varyasyonlu bir tahmin elde etmek isteniyorsa, Q ’yu azaltılır. Bu, R matrisinin etkisine tam zıttır.

Kalman filtresinin başarısı, doğru bir sistem modeline sahip olmak ve gürültü kovaryanslarını doğru bir şekilde ayarlamaya bağlıdır. Q ve R matrislerinin uygun şekilde seçilmesi, Kalman filtresinin performansını önemli ölçüde etkiler. Bu nedenle, bu matrislerin dikkatli bir şekilde belirlenmesi ve sistemin spesifik gereksinimlerine göre ayarlanması gerekir.

6.2.2. Genişletilmiş Kalman Filtresi

Kalman filtresi, doğrusal bir sistem için geliştirilen bir algoritmadır; bu yüzden doğrusal olmayan bir sisteme doğrudan uygulanamaz. Ancak, çevremizdeki sistemlerin çoğu doğrusal değildir. Kalman filtresinin doğrusal olmayan bir sisteme uygulanabilmesi için Genişletilmiş Kalman Filtresi (EKF) yöntemi geliştirilmiştir.

EKF için kullanılacak sistem modeli aşağıdaki gibi doğrusal olmayan olmalıdır:

$$x_{k+1} = f(x_k) + w_k \quad (6.15)$$

$$z_k = h(x_k) + v_k \quad (6.16)$$

Lineer Kalman filtresinde kullanılan Ax_k ve Hx_k terimleri, doğrusal olmayan modeli tanımlayan denklemlerle değiştirilmiştir. Kalan tüm denklemler aynıdır.

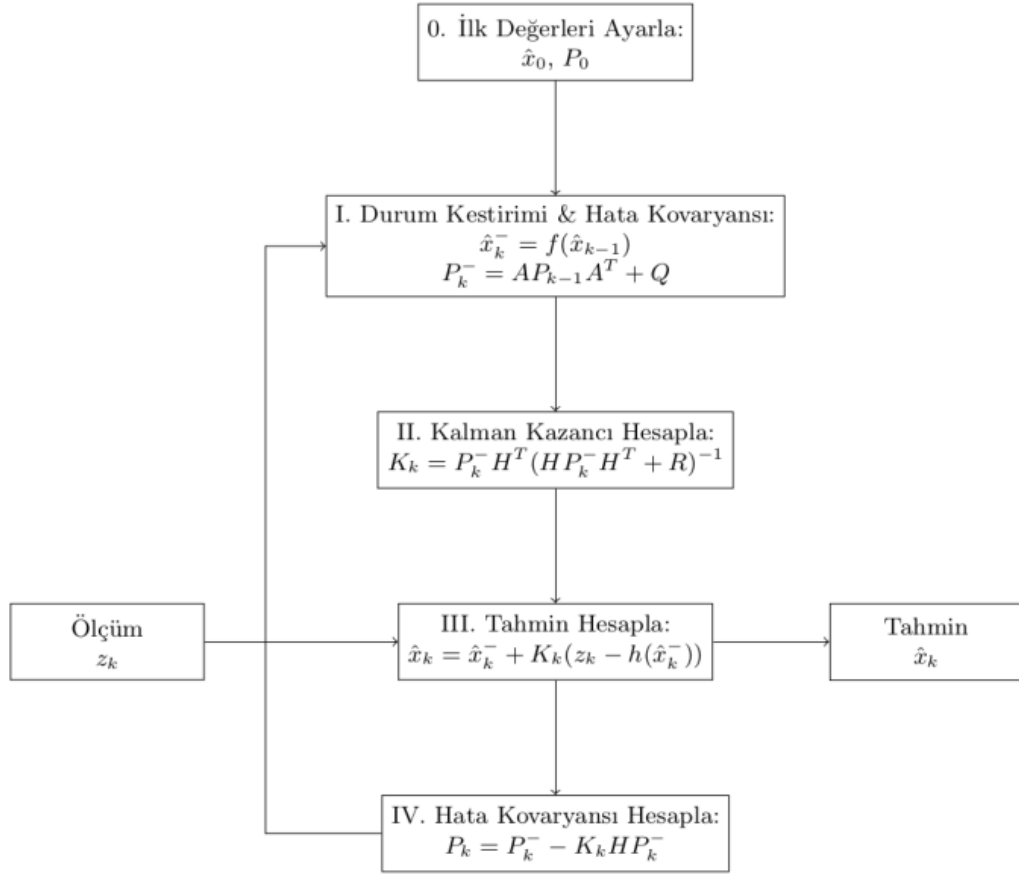
Denklemler 6.15 ve 6.16 doğrusal olmayan sistem modeliyle tanımlandığında, A ve H matrislerini türetmek için aşağıdaki işlemler yapılır.

Doğrusal Kalman filtresinde, doğrusal olmayan bir sistem modeli doğrusallaştırılarak çözülür. Aynı şekilde, Genişletilmiş Kalman filtresi de doğrusal olmayan modeli doğrusal hale getirir. Doğrusallaştırma tekniği klasik bir yöntemdir:

$$A \equiv \left. \frac{\partial f}{\partial x} \right|_{\hat{x}_k}, \quad H \equiv \left. \frac{\partial h}{\partial x} \right|_{\hat{x}_k^-} \quad (6.17)$$

Burada $\frac{\partial}{\partial x}$, fonksiyonun x 'e göre kısmi türevini almayı ifade eder. Örneğin, $\frac{\partial f}{\partial x}$, fonksiyon f 'nin x 'e göre kısmi türevini temsil eder. Eğer f ve x vektör biçimindeyse, sonuç bir matris olur. $\left. \frac{\partial f}{\partial x} \right|_{\hat{x}_k}$ değeri ise \hat{x}_k 'in x 'e uygulanması ile elde edilir.

Doğrusal ve genişletilmiş Kalman filtreleri arasındaki ortak nokta, her ikisinin de doğrusal olmayan sistem modellerini doğrusallaştırarak çözme sürecidir. Ancak, Genişletilmiş Kalman filtresi, bu süreç için referans noktası olarak \hat{x}_k 'i kullanır. Bu, doğrusal Kalman filtresinden en büyük farkıdır.



Şekil 6.4. Genişletilmiş Kalman Algoritması

Genişletilmiş Kalman filtresi uygulanırken, önceki durum kestirimi referans noktası olarak kullanılmaz. Bunun yerine, sistemin gerçek durumu mümkün olduğunca yakından tahmin edilir. Bu yöntem, doğrusal hale getirilmesi zor olan sistemlerde kullanılır. Örneğin, bir uydunun sabit veya nominal bir yörüngede hareket ettiği durumda Genişletilmiş Kalman filtresi uygundur.

Genişletilmiş Kalman filtresi, doğrusal modelini önceki kestirimlere dayalı olarak elde eder. Genişletilmiş Kalman filtresinin A ve H matrislerini Denklem 6.17 ile elde eder. Geri kalan tüm kısımlar, doğrusal Kalman filtresi algoritması ile aynıdır. Ancak, Genişletilmiş Kalman filtresinin bir dezavantajı vardır: sapma gösterebilir. Bu nedenle, genişletilmiş Kalman filtresi uygulanırken bu durum göz önünde bulundurulmalıdır.

6.2.3. Sistem modeli

Sistem modeli iki temel bileşenden oluşur: Durum Geçiş modeli ve Sistem Modeli Durum geçiş modeli, robotun hareket modeline dayanırken, Ölçüm modeli Lidar ve kamera sensörlerinden elde edilen (poseupdate ve odometri) verileri içerir. EKF'nin her adımında bu modeller kullanılarak robotun pozisyonu ve oryantasyonu güncellenir.

Durum Geçiş Modeli

EKF'de Durum Geçiş Modeli şu şekilde ifade edilir:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_k$$

Burada:

- \mathbf{x}_k : Zaman adımındaki durum vektörü.
- $\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$: Durum vektörünün bir önceki durum ve kontrol girdileri (\mathbf{u}_{k-1}) ile tahmin edilmesini sağlayan model fonksiyonu.
- \mathbf{w}_k : Süreç gürültüsü (genellikle Gaussian dağılımlı ve \mathbf{Q} ile ifade edilir).

Durum Vektörü (\mathbf{x}_k), genellikle robotun pozisyonu, oryantasyonu ve hızı gibi bileşenlerden oluşur. Örneğin:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ v_k \\ \omega_k \end{bmatrix}$$

Burada:

- x_k, y_k : Robotun x ve y pozisyonları.
- θ_k : Robotun oryantasyonu (yaw açısı).
- v_k : Robotun lineer hızı.
- ω_k : Robotun açısal hızı.

Sistem Modeli Fonksiyonu (f), robotun hareket denklemlerine dayalı olarak durum vektörünü bir sonraki zaman adımında tahmin eder. Bu denklemler genellikle diferansiyel sürüş modeline dayanır. Örneğin:

$$\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) = \begin{bmatrix} x_{k-1} + \Delta t \cdot v_{k-1} \cdot \cos(\theta_{k-1}) \\ y_{k-1} + \Delta t \cdot v_{k-1} \cdot \sin(\theta_{k-1}) \\ \theta_{k-1} + \Delta t \cdot \omega_{k-1} \\ v_{k-1} \\ \omega_{k-1} \end{bmatrix}$$

Burada:

- Δt : Zaman adımı.
- v_{k-1} : Robotun bir önceki zaman adımındaki lineer hızı.
- ω_{k-1} : Robotun bir önceki zaman adımındaki açısal hızı.

Süreç gürültüsü matrisi \mathbf{Q} , modeldeki belirsizlikleri ve sistem modelinin doğruluğunu temsil eder. Genellikle, robotun kontrol girdilerindeki belirsizlikler veya modelin doğruluğuna göre yapılandırılır.

$$\mathbf{Q} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_\omega^2 \end{bmatrix}$$

Burada $\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \text{vb.}$ terimler, modelin her bir bileşenine karşılık gelen belirsizlikleri temsil eder.

Lineerleştirme sürecinde, sistem modelinin Jacobian matrisi (\mathbf{F}) hesaplanır. Bu matris, sistem modelinin duruma göre türevlenmiş halidir:

$$\mathbf{F}_k = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{\mathbf{x}_{k-1}, \mathbf{u}_{k-1}}$$

Bu matris, tahmin aşamasında durum vektörünün nasıl değiştiğini belirlemek için kullanılır.

Ölçüm Modeli

Ölçüm modeli, sensörlerden gelen verilerin nasıl kullanıldığını tanımlar. EKF'nin güncelleme aşamasında kullanılır ve şu iki temel bileşeni içerir: Ölçüm

Matrisi (H), sensör verisinin durum vektörüne nasıl yansıtılacağını tanımlar. Ölçüm Gürültü Kovaryans Matrisi (R), sensör verisindeki gürültü ve belirsizliği tanımlar. Ölçüm modeli sensör konfigürasyonuna bağlıdır, hangi sensörlerin hangi bileşenlerinin kullanılacağını ve bu verilerin güvenilirliğini etkiler.

Ölçüm modeli genel olarak şu şekilde ifade edilir:

$$z_k = h(x_k) + v_k \quad (6.18)$$

Burada:

- z_k : ölçüm vektörüdür (örneğin, sensör verileri).
- $h(x_k)$: Durum vektöründen ölçüm vektörüne dönüştüren ölçüm fonksiyonu.
- v_k : Ölçüm gürültüsü (genellikle Gaussian dağılımlı).

Ölçüm vektörü (\mathbf{z}_k), sensör verilerinden elde edilen ve ölçüm olarak EKF'ye verilen Poseupdate ve Odometri değerleri için özelleştirilirse:

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{Z}_{\text{odom},k} \\ \mathbf{Z}_{\text{poseupdate},k} \end{bmatrix} = \begin{bmatrix} x_k^{\text{odom}} \\ y_k^{\text{odom}} \\ \theta_k^{\text{odom}} \\ v_{x,k}^{\text{odom}} \\ v_{y,k}^{\text{odom}} \\ \omega_k^{\text{odom}} \\ x_k^{\text{pose}} \\ y_k^{\text{pose}} \\ \theta_k^{\text{pose}} \end{bmatrix}$$

Bu vektörde:

- $x_k^{\text{odom}}, y_k^{\text{odom}}, \theta_k^{\text{odom}}$: Odometri verisinden alınan pozisyon ve oryantasyon.
- $v_{x,k}^{\text{odom}}, v_{y,k}^{\text{odom}}, \omega_k^{\text{odom}}$: Odometri verisinden alınan hız ve açısal hız.
- $x_k^{\text{pose}}, y_k^{\text{pose}}, \theta_k^{\text{pose}}$: Pose update verisinden alınan pozisyon ve oryantasyon.

Ölçüm Modeli (\mathbf{h}_k), ölçüm vektörünün gerçek durumu nasıl temsil ettiğini

gösterir:

$$\mathbf{h}_k(\mathbf{x}_k) = \begin{bmatrix} h_{\text{odom}}(x_k) \\ h_{\text{pose}}(x_k) \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ v_{x,k} \\ v_{y,k} \\ \omega_k \\ x_k \\ y_k \\ \theta_k \end{bmatrix}$$

\mathbf{R}_k , Ölçüm gürültüsü matrisi, her bir ölçümün hatalarını temsil eder. Bu genişletilmiş matris, her iki sensörün verilerinin ölçüm belirsizliklerini içerecektir.

Matrisin blok diyagonal bir yapısı vardır:

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{R}_{\text{odom}} & 0 \\ 0 & \mathbf{R}_{\text{poseupdate}} \end{bmatrix}$$

$$\mathbf{R}_k = \begin{bmatrix} \sigma_{x_{\text{odom}}}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{y_{\text{odom}}}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{\theta_{\text{odom}}}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{v_{x_{\text{odom}}}}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{v_{y_{\text{odom}}}}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\omega_{\text{odom}}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{x_{\text{pose}}}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{y_{\text{pose}}}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\theta_{\text{pose}}}^2 \end{bmatrix}$$

Burada:

- $\sigma_{x_{\text{odom}}}^2, \sigma_{y_{\text{odom}}}^2, \sigma_{\theta_{\text{odom}}}^2$: Odometri verisi için pozisyon ve oryantasyon hataları.
- $\sigma_{v_{x_{\text{odom}}}}^2, \sigma_{v_{y_{\text{odom}}}}^2, \sigma_{\omega_{\text{odom}}}^2$: Odometri verisi için hız ve açısal hız hataları.
- $\sigma_{x_{\text{pose}}}^2, \sigma_{y_{\text{pose}}}^2, \sigma_{\theta_{\text{pose}}}^2$: Pose update verisi için pozisyon ve oryantasyon hataları.

Bu denklemler doğrultusunda ROS üzerinde geliştirilen algoritma ile farklı

konfigurasyonlar yapılarak testler yapılmıştır.

Konfigurasyon matrisi :

$$\begin{bmatrix} x_pos & y_pos & z_pos \\ roll & pitch & yaw \\ x_vel & y_vel & z_vel \\ roll_vel & pitch_vel & yaw_vel \\ x_accel & y_accel & z_accel \end{bmatrix}$$

Poseupdate verisi pozisyon ve yönelim bilgilerini taşır.

$$\text{Pose} = \begin{bmatrix} \text{Konum} & \text{Yönelim} \\ \begin{bmatrix} x_{pos} \\ y_{pos} \\ z_{pos} \end{bmatrix} & \begin{bmatrix} x_{orient} \\ y_{orient} \\ z_{orient} \\ w_{orient} \end{bmatrix} \end{bmatrix}$$

Odometri verisi ise pozisyon, yönelim ve hız verilerini içerir.

$$\text{Odometri Verisi} = \begin{bmatrix} \text{Pozisyon} & \text{Yönelim} & \text{Hız} \\ \begin{bmatrix} x_{pos} \\ y_{pos} \\ z_{pos} \end{bmatrix} & \begin{bmatrix} x_{orient} \\ y_{orient} \\ z_{orient} \\ w_{orient} \end{bmatrix} & \begin{bmatrix} \text{linear}_x \\ \text{linear}_y \\ \text{linear}_z \\ \text{angular}_x \\ \text{angular}_y \\ \text{angular}_z \end{bmatrix} \end{bmatrix}$$

Poseupdate ve Odometri verilerinden gelen değerlerin hangilerinin kullanılacağını konfigürasyon matrisleri ile belirlenir. Farklı değerlerin alınması ölçüm modelinin değişimine sebep olur.

Sistem modeli, robotun dinamik modelinden ve sensör ölçümlerinden elde edilen bilgileri kullanarak durumu tahmin etmekte ve güncellemektir. Her iki model de robotun hareketini ve sensör verilerini entegre etmek için önemlidir.

Durum geçiş modeli, robotun hareket modeline dayanırken, ölçüm modeli hem poseupdate hem de odometri verilerini içerir. Robot Localization paketinde kullanılan EKF'de sensör verilerinin konfigürasyonu, EKF'nin ölçüm denklemlerini ve sensörlerin sisteme nasıl entegre edildiğini doğrudan etkiler. Ancak bu konfigürasyonlar, Durum geçiş modelini etkilemez. Durum geçiş modelinde yapılan

değişiklikler, sadece robotun nasıl hareket ettiğine dair tahminleri etkiler. Odometri ve poseupdate gibi sensör verilerinin konfigürasyonu, yalnızca ölçüm modelini etkiler. Yani, hangi sensör verisinin hangi bileşenlerinin (örneğin x, y pozisyonu, yaw açısı) kullanıldığı, hangi verilerin daha güvenilir olduğu gibi parametreler Ölçüm Matrisi (H) ve Ölçüm gürültü kovaryans matrisi (R) üzerinde etkili olur.

6.2.4. Reçeteler

Farklı konfigürasyonların yapılandırılmaları ile reçeteler oluşturulmuştur.

Reçete 1: Odometri verisinin x, y, yaw, x_vel, y_vel ve yaw_vel değerleri ile Poseupdate verisinin ise x, y, yaw ve zamana göre yönelim değişimlerinden hesaplanan yaw_vel kullanılır.

odom0_config:	pose0_config:
$\begin{bmatrix} \text{true} & \text{true} & \text{false} \\ \text{false} & \text{false} & \text{true} \\ \text{true} & \text{true} & \text{false} \\ \text{false} & \text{false} & \text{true} \\ \text{false} & \text{false} & \text{false} \end{bmatrix}$	$\begin{bmatrix} \text{true} & \text{true} & \text{false} \\ \text{false} & \text{false} & \text{true} \\ \text{false} & \text{false} & \text{false} \\ \text{false} & \text{false} & \text{true} \\ \text{false} & \text{false} & \text{false} \end{bmatrix}$

Reçete 2: Yalnızca Poseupdate verisinin x, y, yaw değerleri kullanılır.

pose0_config:
$\begin{bmatrix} \text{true} & \text{true} & \text{false} \\ \text{false} & \text{false} & \text{true} \\ \text{false} & \text{false} & \text{false} \\ \text{false} & \text{false} & \text{false} \\ \text{false} & \text{false} & \text{false} \end{bmatrix}$

Reçete 3: Odometri verisinin x_vel, y_vel ve yaw_vel değerleri ile Poseupdate verisinin ise x, y, yaw kullanılır.

odom0_config:	pose0_config:
$\begin{bmatrix} \text{false} & \text{false} & \text{false} \\ \text{false} & \text{false} & \text{false} \\ \text{true} & \text{true} & \text{false} \\ \text{false} & \text{false} & \text{true} \\ \text{false} & \text{false} & \text{false} \end{bmatrix}$	$\begin{bmatrix} \text{true} & \text{true} & \text{false} \\ \text{false} & \text{false} & \text{true} \\ \text{false} & \text{false} & \text{false} \\ \text{false} & \text{false} & \text{false} \\ \text{false} & \text{false} & \text{false} \end{bmatrix}$

Reçetelerin desenlerle karşılaştırılabilmesi için sistematik ve verimli bir karşılaştırma mekanizması oluşturmak gereklidir. Bu yapıyı formüle ederken, her bir reçetenin durumsal farkındalık modülünden elde edilen desenlere karşılık gelen durum değişkenlerini içermesi gerektiği göz önünde bulundurulmuştur.

Reçeteler, durum değişkenleri (l, f, k_l, k_f) ve bu değişkenlerin kabul edilen değer aralıklarını içeren bir dizi ile tanımlanır.

Örnek bir reçete yapısı:

- **Reçete 1:**

- l : [3, 4] (Aydınlatma durumu kabul edilen aralık: 1 ile 2)
- f : [1, 2] (Özellik çıkarımı durumu kabul edilen aralık: 3 ile 4)
- k_l : [4, 5] (Aydınlatma durumu kestirimi kabul edilen aralık: 4 ile 5)
- k_f : [4, 5] (Özellik çıkarımı durumu kestirimi kabul edilen aralık: 4 ile 5)

6.3. Analiz

Durumsal farkındalık bileşeninden gelen bilgiler reçeteler için tanımlanmış durumlarla karşılaştırılarak uygun reçeteler seçimi gerçekleştirilir.

Analiz aşamasında, elde edilen desen matrisi ve bu matrise dayalı olarak yapılan kestirimler, sistemin çevresel koşullara ve özellik çıkarım performansına verdiği tepkilerin etkinliğini değerlendirmek amacıyla kullanılır. Durumsal farkındalık bileşeninden gelen bilgiler, sistemin o anki çevresel durumu hakkında kapsamlı bir özet sunar ve bu özet, belirli kriterlere dayalı olarak çeşitli durumlarla eşleştirilir. Örneğin, aydınlatma durumu veya özellik çıkarımının yetersiz olduğu bir senaryoda, sistemin bu duruma uygun bir reçeteyi seçip uygulaması beklenir.

Bu süreç, bir karar ağacı yaklaşımına benzer şekilde işler. Ancak burada, her bir durumun belirli bir reçeteye doğrudan karşılık gelmesi gerekmeyebilir. Bunun yerine, durumsal farkındalık bileşeninden gelen bilgilerin genel bir analizi yapılır ve en uygun strateji belirlenir.

Desen içerisinde bulunan durum değişkenlerinin durumunun tespiti için medyan değer hesaplama yapılır. Medyan değer, aydınlatma durumunun ortanca

değerini verir ve bu durumun dağılımı içinde merkezi bir konumu temsil eder. Aşırı uç değerlerin etkisini ortadan kaldırmak için kullanışlıdır.

$$l_{\text{medyan}} = \text{Medyan}(l_1, l_2, \dots, l_n)$$

Hesaplanan medyan değerleri ve kestirim değerlerine göre desen çözme işlemleri yapılır.

Desenlerin durum medyan değerleri, reçetelerde tanımlanan durum aralıkları ile karşılaştırılır. Her bir reçete için benzerlik skoru hesaplanır. Bu skor, reçetenin her bir durum değişkeninin desen ile ne kadar uyumlu olduğunu gösterir. En yüksek benzerlik skoruna sahip reçete, desenle en uyumlu olan olarak seçilir.

Her bir durum değişkeni için desenin değeri, reçetede tanımlanan aralıklarla karşılaştırılır. Eğer desenin değeri reçetede tanımlanan aralık içinde ise, bu durum değişkeni için benzerlik skoru 1 olarak kabul edilir. Kısmi eşleşmeler için ise benzerlik skoru, eşleşme derecesine göre bir ağırlıklandırma ile hesaplanır.

Tüm durum değişkenlerinin benzerlik skorları toplanarak reçete için bir toplam benzerlik skoru elde edilir. Bu skor, reçetenin desen ile ne kadar uyumlu olduğunu gösterir. Tüm reçeteler için hesaplanan toplam benzerlik skorları arasında en yüksek skora sahip olan reçete seçilir.

$$S_i = \sum_{k=1}^m \text{Benzerlik}(D_k, R_{ik})$$

Burada:

- S_i : i -inci reçetenin toplam benzerlik skorudur.
- D_k : Desen matrisindeki k -inci durum değişkeninin medyan değeridir.
- R_{ik} : i -inci reçetede k -inci durum değişkenine karşılık gelen aralıktır.

Bu formülasyon, durumsal farkındalık bileşeninden gelen verilerin sistematik bir şekilde analiz edilmesi ve en uygun eylem planının seçilmesi sürecini ifade eder. Durumsal farkındalığın zenginliği, sensörlerden elde edilen bilgilerin çeşitliliği ile doğrudan ilişkili olup, sistemin çevresel değişikliklere hızlı ve adaptif bir şekilde yanıt vermesini sağlar.

6.4. Karar Verme

Seçilen reçetelerin, gelen anlık sensör verileri ve durumsal farkındalık desenleriyle gerçek zamanlı olarak kontrol edilmesi, SLAM algoritmasının adaptif yeteneklerini optimize etmek için kritik bir adımdır. Bu aşamada, sistemin çevresel koşullara hızlı bir şekilde uyum sağlamasını sağlayarak, hatasız bir haritalama ve konumlandırma performansı elde etmek hedeflenir.

Reçete, belirli bir durumsal farkındalık senaryosu için en uygun eylem planını içeren bir dizi yönergedir. Seçilen reçetenin uygulanması, sistemin mevcut durumunu sürekli olarak izleyerek ve anlık sensör verilerini işleyerek gerçekleştirilir.

Sistemin performansı gerçek zamanlı olarak izlenir ve gerekli durumlarda optimize edilir. Anlık sensör verileri ve reçeteler arasındaki sürekli karşılaştırma, sistemin durumsal farkındalık kapasitesini artırır ve çevreye adaptasyonunu geliştirir.

6.5. Test Sonuçları

Topladığımız Konya-Dataset dizileri üzerinde çeşitli testler gerçekleştirilmiş ve elde edilen sonuçlar karşılaştırılmıştır. Test işlemlerinde, ilk olarak belirlenen dizi için Hector, ORB-SLAM3 ve Cartographer paketleri çalıştırılmıştır. Ardından, aynı dizi ile Hector ve ORB-SLAM3 algoritmalarından elde edilen Odometri ve Poseupdate verileri kullanılarak EKF yardımıyla füzyon çalışmaları yapılmıştır. Son olarak, oluşturduğumuz model çerçevesinde durumsal farkındalıklı adaptif SLAM çalışmaları gerçekleştirilmiştir.

Tüm test sonuçları, farklı kombinasyonlarla karşılaştırılmıştır. Karşılaştırma işlemlerinde, kaydedilen yörünge bilgileri ROS ortamında "evo" paketinde bulunan Göreceli Poz Hatası (Relative Pose Error - RPE) ve Mutlak Poz Hatası (Absolute Pose Error - APE) hata metrikleri kullanılarak analiz edilmiştir.

RPE ve APE, robotik sistemlerde veya SLAM algoritmalarında bir robotun veya aracın tahmin edilen pozisyonu ile gerçek (referans) pozisyonu arasındaki hatayı ölçmek için kullanılan iki farklı hata metriğidir. RPE, zaman içindeki ardışık pozisyonlar arasındaki farkı, yani tahmin edilen ve referans pozisyonlar arasındaki göreceli hatayı ölçer. Bu metrik, özellikle odometri veya kısa süreli hareket tahminlerinin doğruluğunu değerlendirmek için kullanılır. RPE, iki ardışık pozisyon

arasındaki hatayı hesaplayarak kısa vadeli hata birikimlerini ve hareket sırasında ortaya çıkan hataları lokal olarak analiz eder.

Diğer yandan, APE, tahmin edilen pozisyon ile gerçek pozisyon arasındaki mutlak hatayı ölçer. Bu metrik, bir robotun veya aracın belirli bir zamanda nerede olduğunu mutlak referans çerçevesinde değerlendirir. APE, robotun tüm rotası boyunca tahmin edilen pozisyon ile gerçek pozisyon arasındaki farkı ölçmek için kullanılır ve SLAM algoritmalarının genel doğruluğunu ve küresel hatalarını değerlendirmek açısından önemlidir.

RPE ve APE metriklerinde, hataların analiz edilmesi için çeşitli istatistiksel ölçütler kullanılır. Bu ölçütler, hata dağılımını anlamayı ve algoritmanın performansını değerlendirmeyi sağlar:

- **Max (Maksimum Hata):** Tüm hata değerleri arasındaki en büyük değeri ifade eder. Maksimum hata, en kötü durum senaryosunu gösterir. Ancak, bu değer tek başına değerlendirilmesi yanıltıcı olabilir; bu nedenle diğer metriklerle birlikte incelenmelidir.
- **Mean (Ortalama Hata):** Tüm hataların aritmetik ortalamasını ifade eder. Ortalama hata, genel hata büyüklüğünü gösterir. Düşük bir ortalama, genellikle iyi bir performansın göstergesidir; ancak büyük sapmalar veya uç değerler ortalamayı etkileyebilir.
- **Median (Medyan Hata):** Hata değerlerinin sıralanmış listesindeki ortanca değeri ifade eder. Medyan, aşırı uç değerlerden etkilenmeyen merkezi bir eğilim ölçütüdür. Medyanın ortalamaya göre daha düşük olması, uç değerlerin ortalamayı yükselttiğini gösterebilir.
- **Min (Minimum Hata):** Tüm hata değerleri arasındaki en küçük değeri ifade eder. Minimum hata, en iyi durum senaryosunu gösterir. Ancak minimum değer tek başına anlamı sınırlıdır; genel performansı değerlendirmek için diğer metriklerle birlikte ele alınmalıdır.
- **RMSE (Root Mean Square Error - Kök Ortalama Kare Hatası):** Hataların karelerinin ortalamasının kareköküdür. RMSE, hataların büyüklüğünü anlamak için kullanılan yaygın bir metriktir. Genellikle ortalama hata gibi değerlendirilir; ancak büyük hatalar (uç değerler) RMSE'yi daha fazla etkiler. Düşük bir RMSE, yüksek doğruluğun göstergesi olabilir.

- **SSE (Sum of Squared Errors - Kare Hataların Toplamı):** Tüm hata değerlerinin karelerinin toplamını ifade eder. SSE, modelin toplam hata büyüklüğünü gösterir. SSE değeri büyük olan bir model, genel olarak daha fazla hata yapmış demektir. Ancak, SSE değeri veri setinin büyüklüğüne bağlı olarak artabilir; bu nedenle diğer metriklerle birlikte değerlendirilmesi önemlidir.
- **Std (Standart Sapma):** Hata değerlerinin ortalama etrafındaki dağılımını ifade eder. Standart sapma, hataların ne kadar çeşitlilik gösterdiğini, yani hataların ne kadar yayılmış olduğunu gösterir. Düşük bir standart sapma, hata değerlerinin çoğunun ortalamaya yakın olduğunu gösterir, bu da istikrarlı bir performansı işaret eder. Yüksek bir standart sapma ise hataların geniş bir aralığa yayıldığını ve istikrarsızlık olabileceğini gösterir.

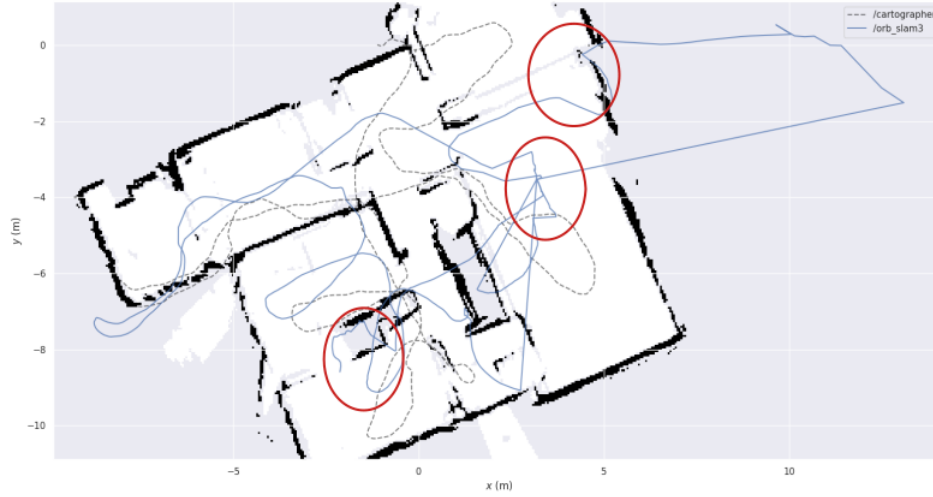
Genel performans, Ortalama (Mean), Medyan (Median) ve Kök Ortalama Kare Hatası (RMSE) gibi metrikler, sistemin genel performansını değerlendirirken, Maksimum (Max) ve Minimum (Min) değerler ise en iyi ve en kötü senaryoları göstermektedir. Hata dağılımı, Standart sapma (Std) ve medyan, hata dağılımının analiz edilmesine yardımcı olmakta, bu da sistemin ne kadar tutarlı olduğunu ortaya koymaktadır.

Bu istatistiksel ölçütler, durumsal farkındalıklı adaptif SLAM algoritmamızın performansını değerlendirmede kritik bir rol oynamaktadır.

Konya veriseti üzerinde yapılan testlerde, modelin kullanıldığı ve kullanılmadığı durumlar ile ORB-SLAM3, Hector, ve Cartographer algoritmaları ayrı ayrı değerlendirilmiştir. Bu çalışmada, Home1.0 dizisine dair test sonuçları paylaşılacak ve analiz edilecektir.

Home1.0 dizisinde ORB-SLAM3 algoritması ile elde edilen yörüngede, özniteliksiz sahneler, saf dönüşler ve hızlı hareketlerden kaynaklanan yörünge takibi sorunları tespit edilmiştir. Şekil 6.5’de, yörünge takibinin sağlanamadığı noktalarda algoritmanın başlangıç noktasından yeniden başladığı ve kopma süresince düz bir çizgi halinde anlamsız veri ürettiği gözlemlenmiştir.

Kopma noktalarının tespit edildiği sürelerde üretilen odometri verisinin kırılmasıyla Şekil 6.6 elde edilmiştir. Bu harita üzerinde 1, 2 ve 3 ile işaretlenen noktalarda, özniteliksiz sahneler, aynalar ve aydınlatma durumundaki değişimlerin kopmalara neden olduğu görülmüştür.



Şekil 6.5. Home1.0 Yörünge Çizgisi

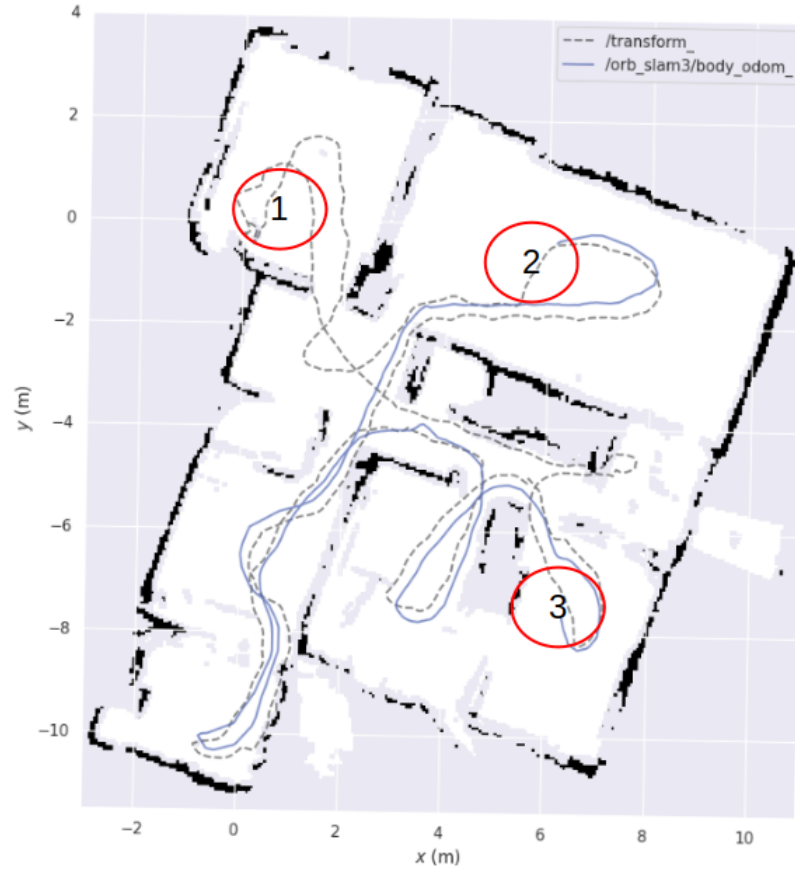
Sorunların tespit edildiği bu diziye, Adaptif SLAM Karar Modeli uygulanmıştır. Durumsal farkındalık kazandırılan model, ortam aydınlatma durumu ve aktif takip edilen öznitelik sayısına göre reçeteler bölümünde kaydedilen üç farklı reçete arasında geçişler yaparak süreci tamamlamıştır.

Önerilen model, modelsiz EKF ve EKF'siz ORB-SLAM çıktıları üzerinde yapılan Cartographer algoritması sonuçlarının referans alındığı APE ve RPE analizleri, modelin etkinliğini ve diğer yaklaşımlarla karşılaştırmalı performansını değerlendirmede önemli bulgular sunmaktadır.

EKF'siz ORB-SLAM, mutlak konumlandırma performansı açısından önemli ölçüde yüksek APE değerleri göstermektedir. Bu durum, mutlak pozisyon hatalarının büyük olduğunu ve sistemin genel olarak doğru konumlandırma yapmada zorlandığını göstermektedir. Buna karşın, RPE değerlerinin nispeten düşük olması, kısa vadeli tutarlılığın korunduğunu ve büyük sapmalara rağmen bağlı doğruluğun sağlanabildiğini göstermektedir.

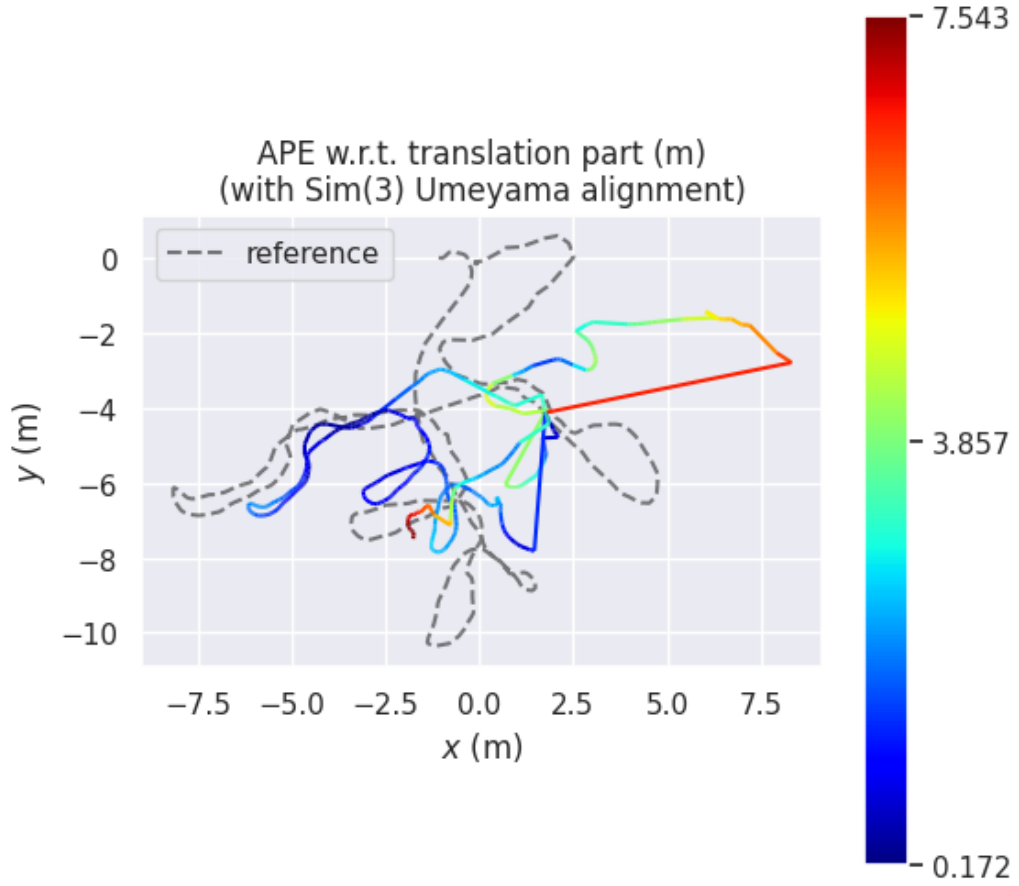
Modelsiz EKF, nispeten düşük APE ve RPE değerleri ile stabil sonuçlar vermektedir. Hataların dağılımı, düşük standart sapma değerleri ile birlikte, sistemin genel olarak dengeli ve tutarlı çalıştığını göstermektedir. Bununla birlikte, bozuk veriler filtreye dahil edildiğinde, filtrenin performansı olumsuz etkilenmektedir.

Önerilen model, modelsiz EKF'ye kıyasla APE'de hafif bir artış göstermektedir. Bu durum, modelin mutlak konumlandırma doğruluğu açısından bir miktar dezavantajlı olabileceğini göstermektedir. Ancak, RPE'deki iyileşme, modelin kısa vadeli tutarlılık açısından daha başarılı olduğunu ve daha istikrarlı bir



Şekil 6.6. Home1.0 Yörünge ve Kopma Noktaları

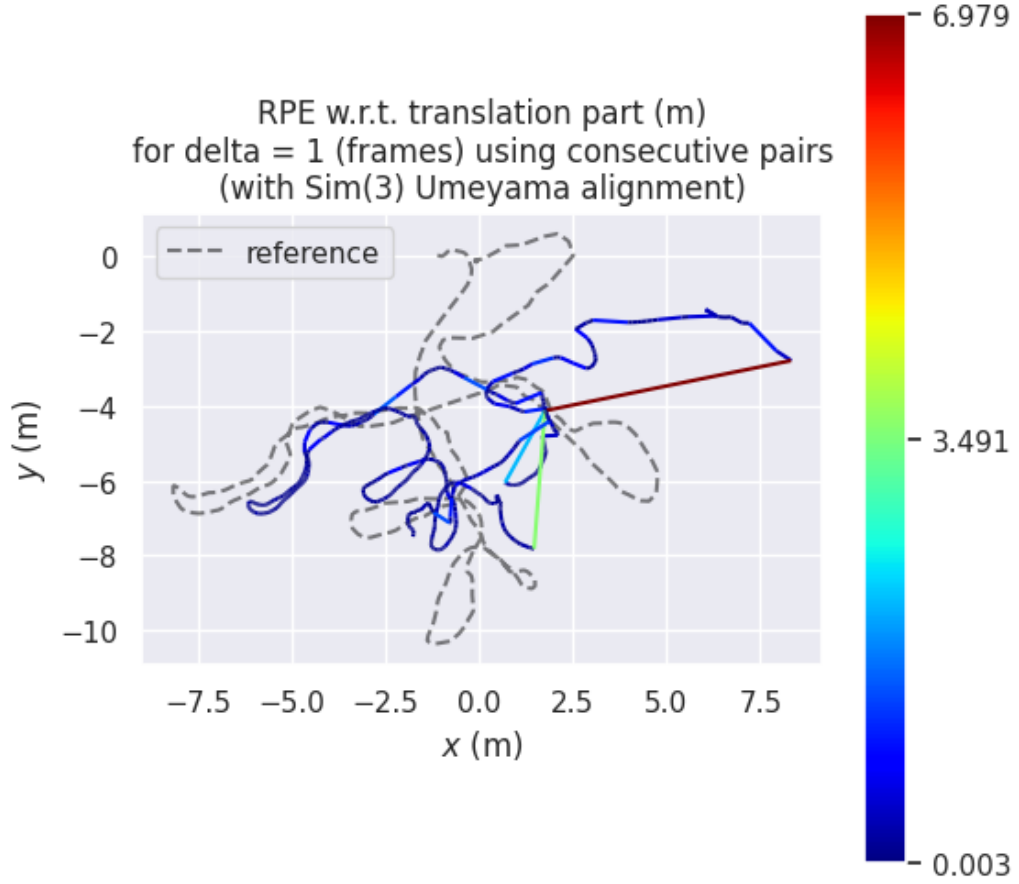
performans sergilediğini göstermektedir. Bu durum, modelin genel yörünge izleme performansını iyileştirdiğini ve durumsal farkındalığın SLAM algoritmalarına entegre edilmesinin avantajlarını ortaya koymaktadır.



Şekil 6.7. ORB-SLAM3 yörünge APE hata sonuçları

ORB-SLAM3 APE Sonuçları

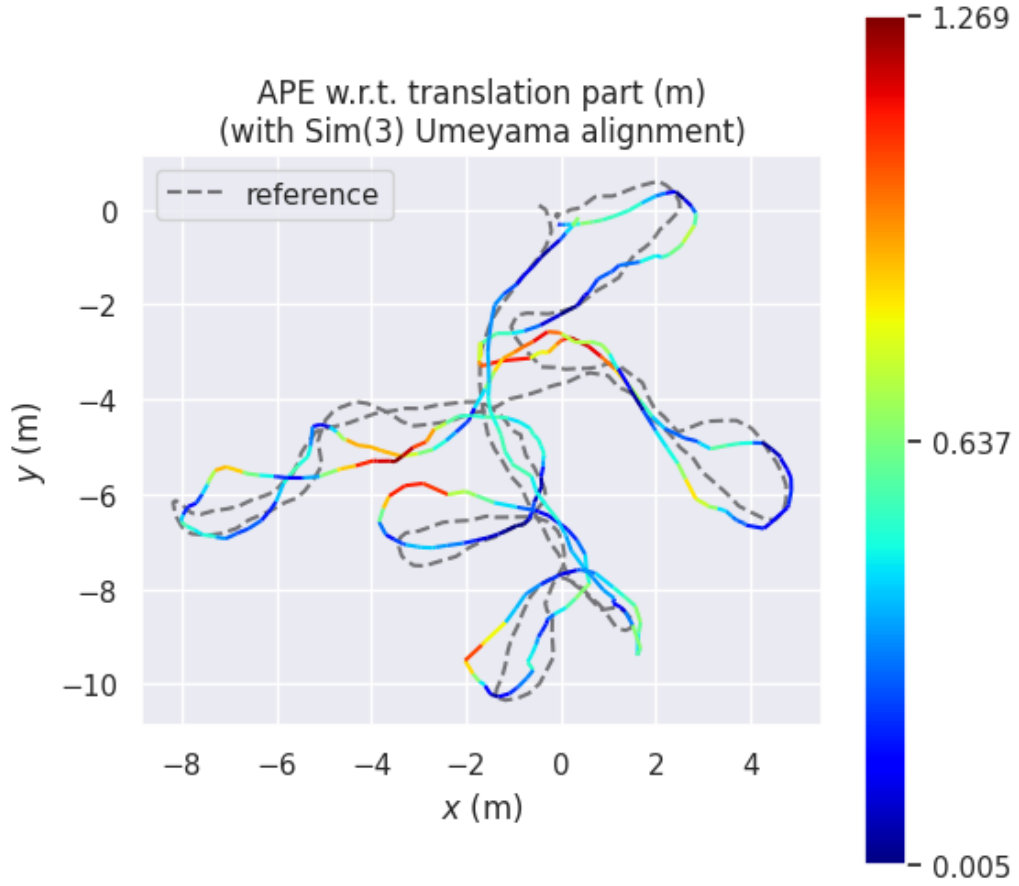
Maximum (max)	:	7.542640
Mean (mean)	:	2.269390
Median (median)	:	1.979212
Minimum (min)	:	0.171564
Root Mean Square Error (rmse)	:	2.757210
Sum of Squared Errors (sse)	:	5587.622721
Standard Deviation (std)	:	1.565910



Şekil 6.8. ORB-SLAM3 yörünge RPE hata sonuçları

ORB-SLAM3 RPE Sonuçları

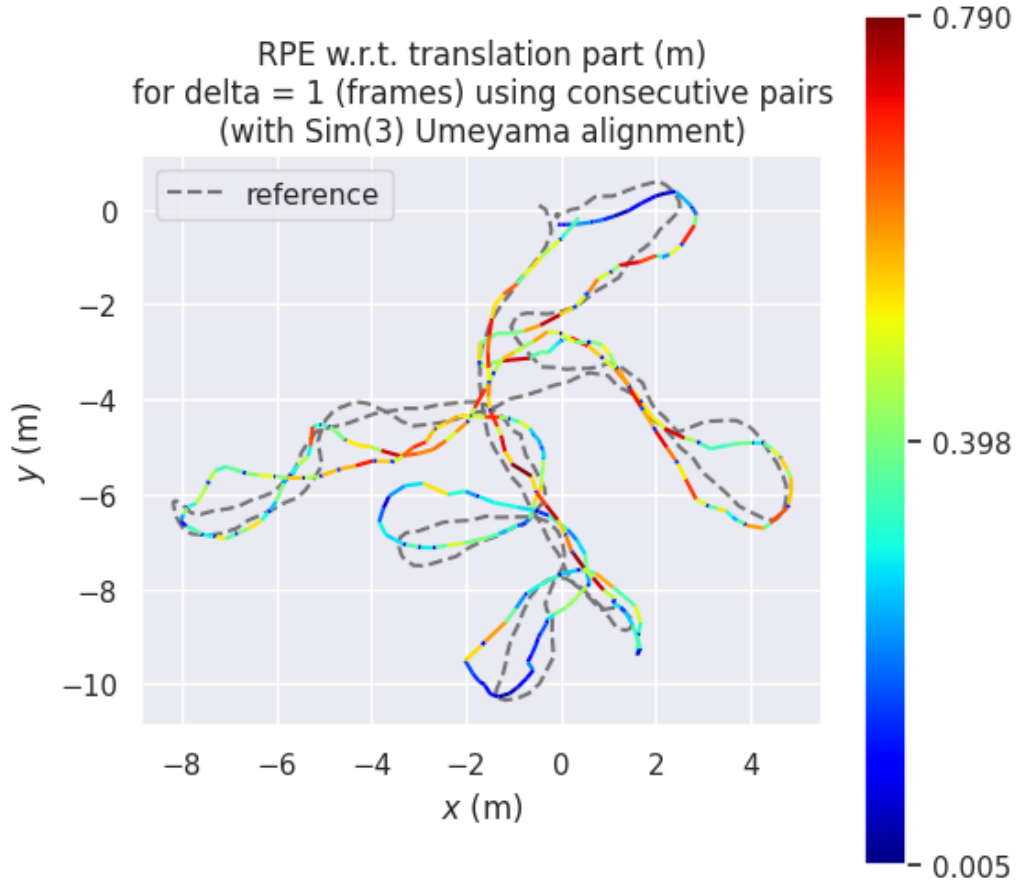
Maximum (max)	:	6.979453
Mean (mean)	:	0.152556
Median (median)	:	0.079867
Minimum (min)	:	0.002772
Root Mean Square Error (rmse)	:	0.374508
Sum of Squared Errors (sse)	:	102.947918
Standard Deviation (std)	:	0.342027



Şekil 6.9. EKF yörünge APE hata sonuçları

EKF APE Sonuçları

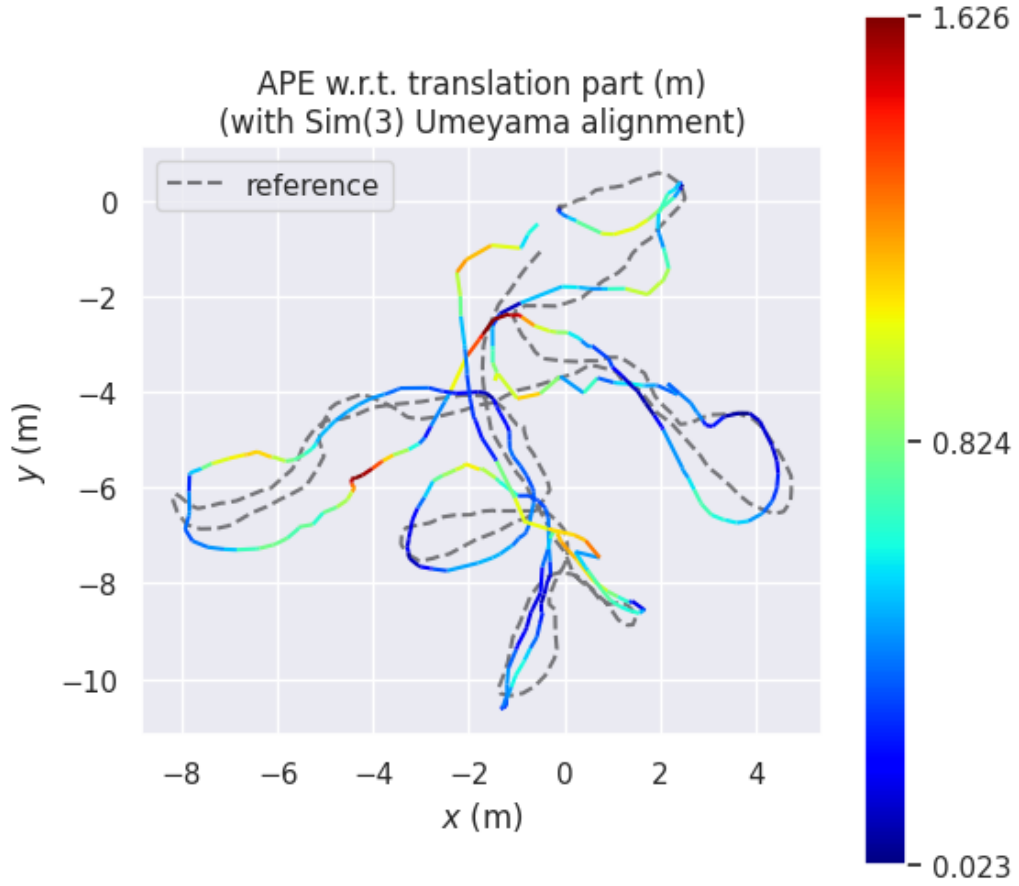
Maximum (max)	:	1.269212
Mean (mean)	:	0.464982
Median (median)	:	0.440703
Minimum (min)	:	0.005469
Root Mean Square Error (rmse)	:	0.545131
Sum of Squared Errors (sse)	:	106.386145
Standard Deviation (std)	:	0.284534



Şekil 6.10. EKF yörünge RPE hata sonuçları

EKF RPE Sonuçları

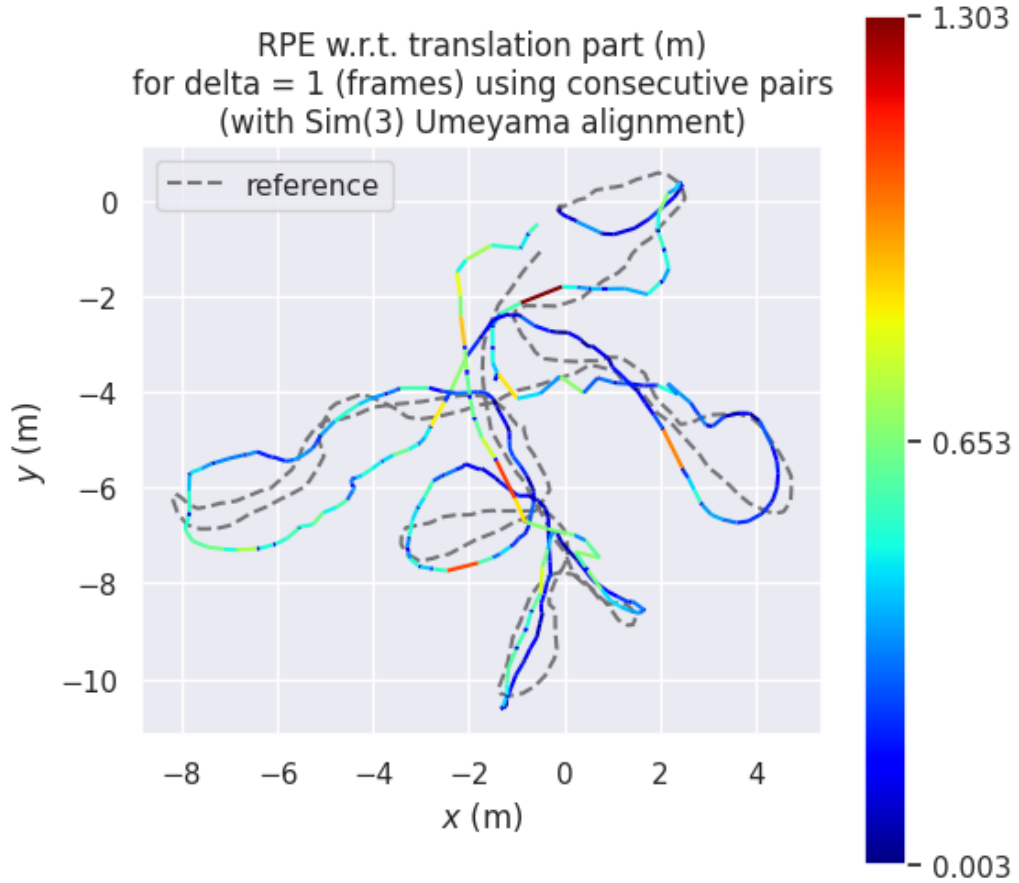
Maximum (max)	:	0.790164
Mean (mean)	:	0.321595
Median (median)	:	0.326627
Minimum (min)	:	0.005484
Root Mean Square Error (rmse)	:	0.383144
Sum of Squared Errors (sse)	:	52.407351
Standard Deviation (std)	:	0.208269



Şekil 6.11. Model ile elde edilen yörünge APE hata sonuçları

Model APE Sonuçları

Maximum (max)	:	1.626042
Mean (mean)	:	0.564261
Median (median)	:	0.481396
Minimum (min)	:	0.022793
Root Mean Square Error (rmse)	:	0.665729
Sum of Squared Errors (sse)	:	158.663666
Standard Deviation (std)	:	0.353276



Şekil 6.12. Model ile elde edilen yörünge RPE hata sonuçları

Model RPE Sonuçları

Maximum (max)	:	1.303265
Mean (mean)	:	0.266930
Median (median)	:	0.192189
Minimum (min)	:	0.003356
Root Mean Square Error (rmse)	:	0.350203
Sum of Squared Errors (sse)	:	43.783171
Standard Deviation (std)	:	0.226694

7. SONUÇ VE ÖNERİLER

Bu çalışmada geliştirilen Durumsal Farkındalık Temelli Adaptif SLAM Karar Modeli, SLAM algoritmalarının mevcut sınırlamalarını aşmayı hedefleyerek, daha sağlam, güvenilir ve çevre şartlarına uyum sağlayan bir yapı ortaya koymuştur. Çalışmamız, SLAM sistemlerinin performansını artırmak amacıyla, sağlam performans, yüksek seviye anlayış, kaynak farkındalığı ve görev odaklı çıkarım gibi çok yönlü zorlukları ele almıştır. Model, zorlu veri setleri üzerinde gerçekleştirilen testlerle etkinliğini kanıtlamış ve adaptif tepkiler üretebilme kapasitesi sayesinde SLAM algoritmalarına esneklik kazandırmıştır.

7.1. Sonuçlar

Geliştirilen model, SLAM algoritmalarına başarısızlığa karşı korunaklı ve kendini ayarlayabilen bir yapı sunarak performanslarını önemli ölçüde artırmıştır. Zorlu çevre koşulları ve ani değişimler karşısında, sistemin esnekliği ve dayanıklılığı belirgin bir şekilde iyileşmiştir. Modelin çevreyi sadece algılamakla kalmayıp, çevreyi kavrama yeteneği sayesinde SLAM algoritmaları daha derin bir anlayış ve çıkarım yeteneği kazanmıştır. Bu, özellikle zorlu şartlarda daha doğru ve güvenilir harita oluşturma süreçlerine olanak tanımıştır.

Modelin çevrimiçi ve ömür boyu öğrenme yetenekleri, sistemin uzun vadeli otonomi sağlamasına katkıda bulunmuştur. Yeni nesnelere ve sahnelerle karşılaştığında kendini adapte edebilme ve performansını sürekli iyileştirebilme yeteneği, SLAM sistemlerinin daha sağlam ve esnek hale gelmesine yardımcı olmuştur. Bu özellikler, SLAM algoritmalarının gerçek dünya şartlarında karşılaşılan zorluklara karşı hızlı ve etkili çözümler üretebilmesini sağlamıştır.

7.2. Öneriler

Gelecekteki çalışmalarda, modelin performansını daha da artırmak amacıyla derin öğrenme ve yapay zeka tekniklerinin modelle entegre edilmesi önerilmektedir.

Bu entegrasyon, sistemin çevresel farkındalık ve öğrenme süreçlerinin iyileştirilmesine katkı sağlayacaktır. Ayrıca, geliştirilen modelin farklı görevler ve ortamlar için uygulanabilirliğini artırmak amacıyla daha fazla test ve geliştirme yapılması önerilmektedir. Bu, modelin geniş ölçekli ve dinamik ortamlarda performansını artıracak ve daha geniş bir uygulama alanı bulmasını sağlayacaktır.

Modelin ömür boyu öğrenme ve adaptasyon süreçlerinin daha da geliştirilmesi için uzun vadeli araştırmalar yapılması gerekmektedir. Bu çalışmalar, sistemin zaman içinde daha karmaşık çevresel faktörlere karşı kendini adapte etmesine olanak tanıyacaktır.

Bu tez, SLAM algoritmalarını sadece bir algılama ve anlamlandırma mekanizması olarak değil, adaptif, öğrenen ve dayanıklı bir teknoloji olarak yeniden tanımlamış ve bu alanda önemli bir adım atmıştır. Gelecekte yapılacak çalışmalarla modelin daha da geliştirilmesi ve yaygınlaştırılması, otonom sistemlerin güvenilirliğini ve performansını önemli ölçüde artıracaktır.

Çalışma ayrıca, toplulukla açık kaynak olarak paylaşılan ve farklı kapalı ortamlarda çeşitli zorlukları barındıran Konya-Dataset adındaki veri seti ile de katkıda bulunmuştur. Bu veri seti, son yıllarda keşfedilen yeni zorlukları ve destekleyici nitelikte diziler barındırmaktadır. Veri setiyle yapılan testler, mevcut SLAM algoritmalarının sınırlarını zorlayarak, bu zorluklar karşısında gösterdikleri performansın açık alanların keşfedilmesinde ve gelecekteki çalışmaların odaklanılmasında önemli bir katkı sağladığını göstermiştir.

KAYNAKLAR

- Ahmed, S., Huda, M. N., Rajbhandari, S., Saha, C., Elshaw, M., ve Kanarachos, S. (2019). Pedestrian and cyclist detection and intent estimation for autonomous vehicles: A survey. *Applied Sciences*, 9(11):2335.
- Akai, N., Morales, L. Y., ve Murase, H. (2018). Simultaneous pose and reliability estimation using convolutional neural network and Rao–Blackwellized particle filter. *Advanced Robotics*, 32(17):930–944.
- Armeni, I., He, Z.-Y., Gwak, J., Zamir, A. R., Fischer, M., Malik, J., ve Savarese, S. (2019). 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, s. 5664–5673.
- Bar-Shalom, Y., Li, X. R., ve Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.
- Barnes, D., Gadd, M., Murcutt, P., Newman, P., ve Posner, I. (2020). The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE international conference on robotics and automation (ICRA)*, s. 6433–6438. IEEE.
- Bavle, H., Sanchez-Lopez, J. L., Schmidt, E. F., ve Voos, H. (2023a). From SLAM to Situational Awareness: Challenges and Survey. *Sensors*, mayıs:1–34.
- Bavle, H., Sanchez-Lopez, J. L., Shaheer, M., Civera, J., ve Voos, H. (2022). Situational Graphs for Robot Navigation in Structured Indoor Environments. *IEEE Robotics and Automation Letters*, 7(4):9107–9114.
- Bavle, H., Sanchez-Lopez, J. L., Shaheer, M., Civera, J., ve Voos, H. (2023b). S-Graphs+ : Real-time Localization and Mapping leveraging Hierarchical Representations . *IEEE Robotics and Automation Letters*, 8(8):1–8.
- Ben Ghezala, M. W., Bouzeghoub, A., ve Leroux, C. (2014). RSAW: A situation awareness system for autonomous robots. *2014 13th International*

Conference on Control Automation Robotics and Vision, ICARCV 2014, 2014(December):450–455.

Biber, P. ve Duckett, T. (2009). Experimental analysis of sample-based maps for long-term SLAM. *International Journal of Robotics Research*, 28(1):20–33.

Blanco-Claraco, J.-L., Moreno-Duenas, F.-A., ve González-Jiménez, J. (2014a). The Málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario. *The International Journal of Robotics Research*, 33(2):207–214.

Blanco-Claraco, J. L., Moreno-Dueñas, F. Á., ve González-Jiménez, J. (2014b). The Málaga urban dataset: High-rate stereo and LiDAR in a realistic urban scenario. *International Journal of Robotics Research*, 33(2):207–214.

Bowman, S. L., Atanasov, N., Daniilidis, K., ve Pappas, G. J. (2017). Probabilistic data association for semantic slam. In *2017 IEEE international conference on robotics and automation (ICRA)*, s. 1722–1729. IEEE.

Bresson, G., Alsayed, Z., Yu, L., ve Glaser, S. (2017). Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., ve Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. *International Journal of Robotics Research*, 35(10):1157–1163.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., ve Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332.

Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M., ve Tardós, J. D. (2021). Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890.

Carlevaris-Bianco, N., Ushani, A. K., ve Eustice, R. M. (2016). University of

- Michigan North Campus long-term vision and lidar dataset. *International Journal of Robotics Research*, 35(9):1023–1035.
- Caruso, D., Engel, J., ve Cremers, D. (2015). Large-scale direct SLAM for omnidirectional cameras. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:141–148.
- Ceriani, S., Fontana, G., Giusti, A., Marzorati, D., Matteucci, M., Migliore, D., Rizzi, D., Sorrenti, D. G., ve Taddei, P. (2009). Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, 27(4):353–371.
- Chang, Z., Wu, H., ve Li, C. (2022). YOLOv4-tiny-based robust RGB-D SLAM approach with point and surface feature fusion in complex indoor environments. *Journal of Field Robotics*, (June):1–14.
- Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., ve Stachniss, C. (2017a). Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *International Journal of Robotics Research*, 36(10):1045–1052.
- Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., ve Stachniss, C. (2017b). Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research*, 36(10):1045–1052.
- Chen, W., Zhou, C., Shang, G., Wang, X., Li, Z., Xu, C., ve Hu, K. (2022). SLAM Overview : From Single Sensor to Heterogeneous Fusion. *Remote Sensing*, s. 1–53.
- Chen, X., Milioto, A., Palazzolo, E., Giguere, P., Behley, J., ve Stachniss, C. (2019). Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, s. 4530–4537. IEEE.
- Cieslewski, T., Lynen, S., Dymczyk, M., Magnenat, S., ve Siegwart, R. (2015). Map

- api-scalable decentralized map building for robots. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, s. 6241–6247. IEEE.
- Cohen, M. S. (1993). Metacognitive strategies in support of recognition. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 37, s. 1102–1106. SAGE Publications Sage CA: Los Angeles, CA.
- Considine, D. M. ve Considine, G. D. (2012). *Standard handbook of industrial automation*. Springer Science & Business Media.
- Cossitt, J. E., W. Carruth, D., II Swan, J. E., ve Jingdao Chen (2022). Developing a Model of Driver Performance, Situation Awareness, and Cognitive Load Considering Different Levels of Partial Vehicle Autonomy.
- Cui, Y., Dai, X., Dai, L., ve Wang, K. (2022a). An Overview on the Development and Application of Cognitive Robot. *Frontiers in Artificial Intelligence and Applications*, 358:359–368.
- Cui, Y., Dai, X., Dai, L., ve Wang, K. (2022b). An overview on the development and application of cognitive robot. *FSDM*, s. 359–368.
- Davison, A. J., Reid, I. D., Molton, N. D., ve Stasse, O. (2007). Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067.
- Dayoub, F., Cielniak, G., ve Duckett, T. (2011). Long-term experiments with an adaptive spherical view representation for navigation in changing environments. *Robotics and Autonomous Systems*, 59(5):285–295.
- De Croce, M., Pire, T., ve Bergero, F. (2019). Ds-ptam: distributed stereo parallel tracking and mapping slam system. *Journal of Intelligent & Robotic Systems*, 95(2):365–377.
- Dellaert, F., Fox, D., Burgard, W., ve Thrun, S. (1999). Monte carlo localization for mobile robots. In *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, volume 2, s. 1322–1328. IEEE.

- Edwards, W. (1962). Dynamic decision theory and probabilistic information processings. *Human factors*, 4(2):59–74.
- Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1):32–64.
- Eustice, R., Walter, M., ve Leonard, J. (2005). Sparse extended information filters: Insights into sparsification. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, s. 3281–3288. IEEE.
- Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., ve Stewart, B. (2006). Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7):1325–1339.
- Geiger, A., Lenz, P., Stiller, C., ve Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32(11):1231–1237.
- Geneva, P., Eckenhoff, K., ve Huang, G. (2018). Asynchronous multi-sensor fusion for 3d mapping and localization. In *2018 IEEE international conference on robotics and automation (ICRA)*, s. 5994–5999. IEEE.
- Gonzalez, C. (2012). Cognitive models of training principles and the instance-based learning tool. *Training Cognition*, s. 181–200.
- Gonzalez, C. (2017). Oxford Handbooks Online Decision-Making : A Cognitive Science Perspective. (April):1–27.
- Gonzalez, C., Dutt, V., ve Lejarraga, T. (2011). A loser can be a winner: Comparison of two instance-based learning models in a market entry competition. *Games*, 2(1):136–162.
- Gonzalez, C., Lerch, J. F., ve Lebiere, C. (2003). Instance-based learning in dynamic decision making. *Cognitive Science*, 27(4):591–635.
- Gonzalez, C., Thomas, R. P., ve Vanyukov, P. (2005). The relationships between cognitive ability and dynamic decision making. *Intelligence*, 33(2):169–186.

- Grisetti, G., Stachniss, C., ve Burgard, W. (2007). Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34–46.
- Haag, M. U. D. ve Huschbeck, S. Cognitive and Collaborative sUAS Swarms in Urban Environments. s. 1–18.
- Hess, W., Kohler, D., Rapp, H., ve Andor, D. (2016). Real-time loop closure in 2d lidar slam. In *2016 IEEE international conference on robotics and automation (ICRA)*, s. 1271–1278. IEEE.
- Huang, A. S., Antone, M., Olson, E., Fletcher, L., Moore, D., Teller, S., ve Leonard, J. (2010). A high-rate, heterogeneous data set from the DARPA urban challenge. *International Journal of Robotics Research*, 29(13):1595–1601.
- Hwang, S.-Y. ve Song, J.-B. (2011). Monocular vision-based slam in indoor environment using corner, lamp, and door features from upward-looking camera. *IEEE Transactions on Industrial Electronics*, 58(10):4804–4812.
- Indelman, V., Nelson, E., Dong, J., Michael, N., ve Dellaert, F. (2016). Incremental distributed inference from arbitrary poses and unknown data association: Using collaborating robots to establish a common reference. *IEEE Control Systems Magazine*, 36(2):41–74.
- Jeong, J., Cho, Y., Shin, Y. S., Roh, H., ve Kim, A. (2019). Complex urban dataset with multi-level sensors from highly diverse urban environments. *International Journal of Robotics Research*, 38(6):642–657.
- Julier, S. J. ve Uhlmann, J. K. (1997). New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, s. 182–193. Spie.
- Julier, S. J. ve Uhlmann, J. K. (2001). A counter example to the theory of simultaneous localization and map building. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*,

volume 4, s. 4238–4243. IEEE.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

Kalman, R. E. ve Bucy, R. S. (1961). New results in linear filtering and prediction theory.

Kaveti, P., Gupta, A., Giaya, D., Karp, M., Keil, C., Nir, J., Zhang, Z., ve Singh, H. (2023). Challenges of Indoor SLAM: A Multi-Modal Multi-Floor Dataset for SLAM Evaluation. *IEEE International Conference on Automation Science and Engineering*, 2023-Augus:1–8.

Kim, P. ve Huh, L. (2011). Kalman filter for beginners: with matlab examples. (*No Title*).

Klein, G. ve Murray, D. (2007). Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, s. 225–234. IEEE.

Klingensmith, M., Dryanovski, I., Srinivasa, S. S., ve Xiao, J. (2015). Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields. In *Robotics: science and systems*, volume 4.

Kohlbrecher, S., Von Stryk, O., Meyer, J., ve Klingauf, U. (2011). A flexible and scalable slam system with full 3d motion estimation. In *2011 IEEE international symposium on safety, security, and rescue robotics*, s. 155–160. IEEE.

Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B., ve Vincent, R. (2010). Efficient sparse pose adjustment for 2d mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, s. 22–29. IEEE.

Korrapati, H., Courbon, J., Alizon, S., ve Marmoiton, F. (2013). "the institut pascal data sets": un jeu de données en extérieur, multicateurs et datées avec réalité terrain, données d'étalonnage et outils logiciels. In *Orasis, Congrès des jeunes chercheurs en vision par ordinateur*.

- Krajník, T., Fentanes, J. P., Mozos, O. M., Duckett, T., Ekekrantz, J., ve Hanheide, M. (2014). Long-term topological localisation for service robots in dynamic environments using spectral maps. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, s. 4537–4542. IEEE.
- Leonard, J. J. ve Durrant-Whyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. In *IROS*, volume 3, s. 1442–1447.
- Li, C., Wang, S., Zhuang, Y., ve Yan, F. (2019). Deep sensor fusion between 2d laser scanner and imu for mobile robot localization. *IEEE Sensors Journal*, 21(6):8501–8509.
- Li, R., Wang, S., ve Gu, D. (2020). Deepslam: A robust monocular slam system with unsupervised deep learning. *IEEE Transactions on Industrial Electronics*, 68(4):3577–3587.
- Liu, T. ve Shen, S. (2017). High altitude monocular visual-inertial state estimation: Initialization and sensor fusion. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, s. 4544–4551. IEEE.
- Liu, Y., Fu, Y., Chen, F., Goossens, B., Tao, W., ve Zhao, H. (2021). Simultaneous Localization and Mapping Related Datasets: A Comprehensive Survey. *arXiv*.
- Liu, Y. ve Thrun, S. (2003). Results for outdoor-slam using sparse extended information filters. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 1, s. 1227–1233. IEEE.
- Lynen, S., Sattler, T., Bosse, M., Hesch, J. A., Pollefeys, M., ve Siegwart, R. (2015). Get out of my lab: Large-scale, real-time visual-inertial localization. In *Robotics: Science and Systems*, volume 1, s. 1.
- Maddern, W., Pascoe, G., Linegar, C., ve Newman, P. (2017). 1 year, 1000 km: The Oxford RobotCar dataset. *International Journal of Robotics Research*, 36(1):3–15.
- Maybeck, P. S. (1982). *Stochastic models, estimation, and control*. Academic press.

- McCormac, J., Handa, A., Leutenegger, S., ve Davison, A. J. (2017). SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation? *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:2697–2706.
- Mu, L., Yao, P., Zheng, Y., Chen, K., Wang, F., ve Qi, N. (2020). Research on slam algorithm of mobile robot based on the fusion of 2d lidar and depth camera. *Ieee Access*, 8:157628–157642.
- Mur-Artal, R., Montiel, J. M. M., ve Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163.
- Newcombe, R. A., Fox, D., ve Seitz, S. M. (2015). Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, s. 343–352.
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., ve Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, s. 127–136. Ieee.
- Nilwong, S., Hossain, D., Kaneko, S.-i., ve Capi, G. (2019). Deep learning-based landmark detection for mobile robot outdoor localization. *Machines*, 7(2):25.
- Nitzan, D. (1985). Development of intelligent robots: achievements and issues. *IEEE Journal on Robotics and Automation*, 1(1):3–13.
- Ohya, I., Kosaka, A., ve Kak, A. (1998). Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing. *IEEE transactions on robotics and automation*, 14(6):969–978.
- Pandey, G., McBride, J. R., ve Eustice, R. M. (2011). Ford Campus vision and lidar data set. *International Journal of Robotics Research*, 30(13):1543–1552.
- Panzieri, S., Pascucci, F., ve Ulivi, G. (2002). An outdoor navigation system

- using gps and inertial platform. *IEEE/ASME transactions on Mechatronics*, 7(2):134–142.
- Peynot, T., Scheduling, S., ve Terho, S. (2010). The marulan data sets: Multi-sensor perception in a natural environment with challenging conditions. *International Journal of Robotics Research*, 29(13):1602–1607.
- Pronobis, A. ve Caputo, B. (2009). COLD: The CoSy localization database. *International Journal of Robotics Research*, 28(5):588–594.
- Qin, T., Li, P., ve Shen, S. (2018). Relocalization, global optimization and map merging for monocular visual-inertial slam. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, s. 1197–1204. IEEE.
- Qiu, K., Qin, T., Gao, W., ve Shen, S. (2019). Tracking 3-d motion of dynamic objects using monocular visual-inertial sensing. *IEEE Transactions on Robotics*, 35(4):799–816.
- Ros, G., Sellart, L., Materzynska, J., Vazquez, D., ve Lopez, A. M. (2016). The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:3234–3243.
- Rui Huang, Y. Z. (2021). An adaptive scheme for degradation suppression in Lidar based SLAM. *Sensor Review*, 41(4):361–367.
- Ruiz-Sarmiento, J. R., Galindo, C., ve Gonzalez-Jimenez, J. (2017). Robot@Home, a robotic dataset for semantic mapping of home environments. *International Journal of Robotics Research*, 36(2):131–141.
- Scharfe-Scherf, M. S. L., Wiese, S., ve Russwinkel, N. (2022). A Cognitive Model to Anticipate Variations of Situation Awareness and Attention for the Takeover in Highly Automated Driving. *Information (Switzerland)*, 13(9).
- Shamwell, E. J., Lindgren, K., Leung, S., ve Nothwang, W. D. (2019).

- Unsupervised deep visual-inertial odometry with online error correction for rgb-d imagery. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2478–2493.
- Shan, T. ve Englot, B. (2018). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, s. 4758–4765. IEEE.
- Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., ve Rus, D. (2020). Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, s. 5135–5142. IEEE.
- Sharma, N., Pandey, J. K., ve Mondal, S. (2023). A Review of Mobile Robots: Applications and Future Prospect. *International Journal of Precision Engineering and Manufacturing*, 24(9):1695–1706.
- Shi, X., Li, D., Zhao, P., Tian, Q., Tian, Y., Long, Q., Zhu, C., Song, J., Qiao, F., Song, L., Guo, Y., Wang, Z., Zhang, Y., Qin, B., Yang, W., Wang, F., Chan, R. H., ve She, Q. (2020). Are We Ready for Service Robots? the OpenLORIS-Scene Datasets for Lifelong SLAM. *Proceedings - IEEE International Conference on Robotics and Automation*, s. 3139–3145.
- Smith, M., Baldwin, I., Churchill, W., Paul, R., ve Newman, P. (2009). The new college vision and laser data set. *International Journal of Robotics Research*, 28(5):595–599.
- Smith, R. C. ve Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56–68.
- Sousa, R. B., Sobreira, H. M., ve Moreira, A. P. (2023). A systematic literature review on long-term localization and mapping for mobile robots. *Journal of Field Robotics*, 40(5):1245–1322.
- Sturm, J., Engelhard, N., Endres, F., Burgard, W., ve Cremers, D. (2012). A

- benchmark for the evaluation of RGB-D SLAM systems. *IEEE International Conference on Intelligent Robots and Systems*, s. 573–580.
- Sun, Q., Yuan, J., Zhang, X., ve Duan, F. (2021). Plane-Edge-SLAM: Seamless Fusion of Planes and Edges for SLAM in Indoor Environments. *IEEE Transactions on Automation Science and Engineering*, 18(4):2061–2075.
- Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, 45(3):52–57.
- Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., ve Durrant-Whyte, H. (2004). Simultaneous localization and mapping with sparse extended information filters. *The international journal of robotics research*, 23(7-8):693–716.
- Tong, C. H., Gingras, D., Larose, K., Barfoot, T. D., ve Dupuis, É. (2013). The Canadian planetary emulation terrain 3D mapping dataset. *International Journal of Robotics Research*, 32(4):389–395.
- Tong, Z., Shi, D., ve Yang, S. (2017). Sceneslam: A slam framework combined with scene detection. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, s. 487–494. IEEE.
- Turnage, D. M. (2016). Simulation results for localization and mapping algorithms. In *2016 Winter Simulation Conference (WSC)*, s. 3040–3051. IEEE.
- Vidal, A. R., Rebecq, H., Horstschaefer, T., ve Scaramuzza, D. (2018). Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001.
- Welch, G., Bishop, G., ve others (1995). An introduction to the kalman filter.
- Westfechtel, T., Ohno, K., Neto, R. P. B., Kojima, S., ve Tadokoro, S. (2019). Fusion of camera and lidar data for large scale semantic mapping. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, s. 257–264. IEEE.
- Whelan, T., Leutenegger, S., Salas-Moreno, R. F., Glocker, B., ve Davison, A. J. (2015). Elasticfusion: Dense slam without a pose graph. In *Robotics: science*

and systems, volume 11, s. 3. Rome, Italy.

- Wood, R., Nagpal, R., ve Wei, G.-Y. (2013). Flight of the robobees. *Scientific American*, 308(3):60–65.
- Wu, S.-C., Wald, J., Tateno, K., Navab, N., ve Tombari, F. (2021). Scenegrphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, s. 7515–7525.
- Xing, K., Zhang, X., Lin, Y., Ci, W., ve Dong, W. (2022). Simultaneous localization and mapping algorithm based on the asynchronous fusion of laser and vision sensors. *Frontiers in Neurorobotics*, 16:866294.
- Ye, H., Chen, Y., ve Liu, M. (2019). Tightly coupled 3d lidar inertial odometry and mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, s. 3144–3150. IEEE.
- Yin, H. ve Berger, C. (2018). When to use what data set for your self-driving car algorithm: An overview of publicly available driving datasets. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2018-March:1–8.
- Yin, J., Li, A., Li, T., Yu, W., ve Zou, D. (2022). M2DGR: A Multi-Sensor and Multi-Scenario SLAM Dataset for Ground Robots. *IEEE Robotics and Automation Letters*, 7(2):2266–2273.
- Yuan, S., Zhang, J., Pan, C., Zhang, C., Wan, M., Zheng, T., ve Gu, J. (2023). A Comprehensive SLAM Dataset for Indoor Exhibition Environments: Data Collection, Processing, and Comparative Evaluation. *2023 IEEE International Conference on Robotics and Biomimetics, ROBIO 2023*, s. 1–6.
- Zghair, N. A. K. ve Al-Araji, A. S. (2021). A one decade survey of autonomous mobile robot systems. *International Journal of Electrical and Computer Engineering*, 11(6):4891–4906.
- Zhang, J., Kaess, M., ve Singh, S. (2016). On degeneracy of optimization-based

- state estimation problems. In *2016 IEEE international conference on robotics and automation (ICRA)*, s. 809–816. IEEE.
- Zhang, J. ve Singh, S. (2015). Visual-lidar odometry and mapping: Low-drift, robust, and fast. In *2015 IEEE international conference on robotics and automation (ICRA)*, s. 2174–2181. IEEE.
- Zhang, J. ve Singh, S. (2017). Low-drift and real-time lidar odometry and mapping. *Autonomous robots*, 41:401–416.
- Zhang, J. ve Singh, S. (2018). Laser–visual–inertial odometry and mapping with high robustness and low drift. *Journal of field robotics*, 35(8):1242–1264.
- Zhang, J., Singh, S., ve others (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and systems*, volume 2, s. 1–9. Berkeley, CA.
- Zhang, L., Helmberger, M., Fu, L. F. T., Wisth, D., Camurri, M., Scaramuzza, D., ve Fallon, M. (2022). Hilti-Oxford Dataset: A Millimetre-Accurate Benchmark for Simultaneous Localization and Mapping. 8(1):408–415.
- Zhang, Y., Carballo, A., Yang, H., ve Takeda, K. (2023). Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196(January):146–177.
- Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10.
- Zhu, A. Z., Thakur, D., Özaslan, T., Pfrommer, B., Kumar, V., ve Daniilidis, K. (2018). The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception. *IEEE Robotics and Automation Letters*, 3(3):2032–2039.
- Zou, Y., Eldemiry, A., Li, Y., ve Chen, W. (2020). Robust rgb-d slam using point and line features for low textured scene. *Sensors*, 20(17):4984.
- Zuo, X., Geneva, P., Lee, W., Liu, Y., ve Huang, G. (2019). Lic-fusion:

Lidar-inertial-camera odometry. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, s. 5848–5854. IEEE.