

**4 BACAQLI ROBOT SİSTEMİ ÜZERİNDE 3D LİDAR İLE
SENSÖR FÜZYONU VE HARİTALAMA**

**SENSOR FUSION AND MAPPING ON A 4 LEGGED ROBOT
SYSTEM WITH 3D LIDAR**

ONURCAN YILMAZ

Danışman

DOÇ. DR. İSMAIL UYANIK

Hacettepe Üniversitesi

Lisansüstü Eğitim - Öğretim ve Sınav Yönetmeliğinin

Elektrik Elektronik Mühendisliği Anabilim Dalı için Öngördüğü

YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

Ocak 2025

ÖZET

4 BACAĞI ROBOT SİSTEMİ ÜZERİNDE 3D LİDAR İLE SENSÖR FÜZYONU VE HARİTALAMA

Onurcan Yılmaz

Yüksek Lisans, Elektrik Elektronik Mühendisliği

Danışman: Doç. Dr. İsmail UYANIK

Ocak 2025, 125 sayfa

Bu tez çalışması, dört bacaklı bir robot üzerinde 3D LiDAR tabanlı sensör füzyonu ve haritalama sistemlerinin geliştirilmesini hedeflemektedir. Çalışmanın temel odak noktası, LiDAR Odometrisi ve Haritalama (LOAM) algoritmasının varyans kestirimi yapamamasından kaynaklanan eksiklikleri gidermek ve bu algoritmayı IMU ve GPS gibi diğer sensörlerle entegre ederek daha güvenilir bir konumlandırma sistemi oluşturmaktır. Bu doğrultuda, LOAM algoritmasının dönüşüm matrislerinin doğruluğunu belirlemek için Hausdorff Mesafesi tabanlı yeni bir varyans kestirim yöntemi önerilmiş ve yöntemin performansı kapalı ve açık alan veri setleri üzerinde analiz edilmiştir.

Tez kapsamında, LOAM algoritmasının varyans kestirimi için ortalama maliyet ve Hausdorff Mesafesi metrikleri karşılaştırılmış, Hausdorff Mesafesi'nin ölçüm ve harita nokta bulutları arasındaki maksimum mesafeyi hesaplayarak varyans tahmini yapmada daha etkili olduğu gösterilmiştir. Elde edilen varyans kestirimi, LiDAR, IMU ve GPS verilerini En Küçük Varyanslı Yansız Kestirici (MVUE) metoduna dayalı tekniklerle birleştirilerek hassas bir pozisyon kestirimi sağlamak amacıyla kullanılmıştır. Ayrıca, Kalman Filtresinin iki tahmini

arasındaki dönüşümün varyansını hesaplamak için doğrudan yapılan işlemler yerine daha hızlı ve etkin bir yöntem önerilmiş ve dönüşüm kestirimi sürecine entegre edilmiştir.

Çalışma kapsamında, kapalı ve açık alanlarda veri toplayabilen bir donanım sistemi tasarlanmış ve Sensör Sistemi v1 (SSv1) ve Sensör Sistemi v2 (SSv2) olarak adlandırılan iki platform geliştirilmiştir. Bu sistemlerde Jetson serisi tek kart bilgisayarlar, Velodyne VLP-16 LiDAR, ZED stereo kameralar ve INS sensörleri kullanılmıştır. Kapalı alan veri seti, ArUco işaretleyiciler ile referans alınarak toplanırken, açık alan testleri için RTK sistemini referans alan sensör sisteminin geliştirilmesine başlanmıştır. Çalışmada, 1 boyutlu ve 3 boyutlu sensör füzyonu deneylerinde MVUE temelli global pozisyon ortalaması ve dönüşüm ortalaması gibi çeşitli metotlar test edilmiştir. Sonuçlar, girdi olarak verilen odometri verilerinden daha düşük hatalar üretmiştir. Bu durum, Hausdorff Mesafesinin, varyans kestirimi için uygun bir metrik olduğunu göstermektedir.

Bağlaık metotlarda, Kalman filtresi ile LOAM'ın haritalama adımı birleştirilerek optimizasyon sürecine pozisyon düzeltmeleri dahil edilmiştir. Bu yaklaşım, hata birikimini azaltmada kısmen başarılı olmuş, ancak IMU tabanlı kayma giderme yöntemlerinin performansını tam olarak geçememiştir. Sonuç olarak, Hausdorff Mesafesi tabanlı varyans kestirimi, LiDAR odometrisinin güvenilirliğini artırarak sensör füzyonunda tutarlılığı sağlamıştır. Geliştirilen donanım altyapısı, kapalı ve açık alanlarda çoklu sensör verisi toplama kapasitesiyle genişletilebilir bir sistem sunmaktadır. İleriki çalışmalar için farklı hata metriklerinin incelenmesi ve makine öğrenmesi tabanlı yöntemlerin entegrasyonu önerilmektedir.

Keywords: LiDAR, LOAM, Varyans Kestirimi, Sensör Füzyonu, Kalman Filtresi, SLAM, Konumlandırma, Haritalama

ABSTRACT

SENSOR FUSION AND MAPPING ON A 4 LEGGED ROBOT SYSTEM WITH 3D LIDAR

Onurcan Yılmaz

Master of Science, Department of Electrical and Electronics Engineering

Supervisor: Doç. Dr. İsmail UYANIK

January 2025, 125 pages

This thesis aims to develop 3D LiDAR-based sensor fusion and mapping systems on a quadruped robot. The primary focus of the study is to address the deficiencies of the LiDAR Odometry and Mapping (LOAM) algorithm, which lacks variance estimation, and to integrate this algorithm with other sensors such as IMU and GPS to create a more reliable localization system. To this end, a new variance estimation method based on the Hausdorff Distance has been proposed to assess the accuracy of LOAM's transformation matrices, and its performance has been analyzed using both indoor and outdoor datasets.

Within the scope of this thesis, the variance estimation capabilities of the LOAM algorithm have been evaluated by comparing the mean cost and Hausdorff Distance metrics. The results demonstrate that the Hausdorff Distance, which computes the maximum distance between measurement and map point clouds, is more effective for variance estimation. The estimated variance has been utilized to achieve precise position estimation by fusing LiDAR, IMU, and GPS data using techniques based on the Minimum Variance Unbiased Estimator (MVUE). Additionally, a faster and more efficient method has been proposed for estimating

the variance of transformations between two predictions of a Kalman Filter, replacing direct computation methods, and integrating this process into transformation estimation.

As part of the study, a hardware system capable of collecting data in both indoor and outdoor environments has been designed, leading to the development of two platforms named Sensor System v1 (SSv1) and Sensor System v2 (SSv2). These systems incorporate Jetson-series single-board computers, Velodyne VLP-16 LiDAR, ZED stereo cameras, and INS sensors. The indoor dataset was collected using ArUco markers as reference points, while for outdoor tests, the development of a sensor system referencing an RTK system was initiated. Various methods, such as MVUE-based global position averaging and transformation averaging, were tested in both one-dimensional and three-dimensional sensor fusion experiments. The results produced lower errors compared to the input odometry data, indicating that the Hausdorff Distance is a suitable metric for variance estimation.

In coupled methods, the mapping step of LOAM was integrated with the Kalman filter to incorporate position corrections into the optimization process. While this approach was partially successful in reducing error accumulation, it did not fully surpass the performance of IMU-based motion correction methods. Consequently, the Hausdorff Distance-based variance estimation has enhanced the reliability of LiDAR odometry and ensured consistency in sensor fusion. The developed hardware infrastructure provides an extensible system with multi-sensor data collection capabilities for both indoor and outdoor environments. Future work should explore different error metrics and integrate machine learning-based approaches.

Keywords: LiDAR, LOAM, Variance Estimation, Sensor Fusion, Kalman Filter, SLAM, Localization, Mapping

TEŞEKKÜR

Öncelikle danışman hocam Doç. Dr. İsmail UYANIK'a tez sürecimdeki katkılarından dolayı, ancak bundan daha çok sağladığı imkanlar ve sayesinde girdiğim bu akademik yol için teşekkür ederim. O olmadan bütün bunlar mümkün olmazdı.

Ayrıca, değerli katkılarından dolayı jüri üyelerim; Prof. Dr. Berkan DÜLEK, Doç. Dr. Şölen Kumbay YILDIZ, Doç. Dr. Mustafa Mert ANKARALI ve Dr. Öğr. Üyesi Gökhan Koray GÜLTEKİN'e de teşekkür ederim.

Robotiğe duyduğum sevginin temelini oluşturan IEEE ODTÜ RAS ailesine, başta Oğuz ÖZDEMİR, Atakan DURMAZ, Eren Emre AYDIN, Mustafa KILINÇ, Yunus Emre İkiz, Aybars AĞAYA, Başer KANDEHİR, Burak SEVSAY, Canberk Sönmez, Denge UZEL, Elif Damla GÜLTEKİN AYDIN, Ferhat GÖLBOL, Sami Alperen AKGÜN, Uğur AÇIKGÖZ, Yusuf KARABACAĞLU ve Seyit Yiğit SIZLAYAN olmak üzere hem onlardan edindiğim sınırsız bilgi ve tecrübe için, hem de birlikte paylaştığımız sayısız sabahlamalar için çok teşekkür ederim. Özellikle Oğuz Özdemir'e tez çalışmam sürecinde yaptığımız beyin fırtınaları, bir problemle karşılaştığımda yaptığı "Abi onun için şöyle bir kütüphane var." yorumları ve tezi yazarken yaptığı düzeltmeler için ayrıca teşekkür ederim.

Tezin yapım ve yazım sürecince mesaimi paylaşan ve yetiştirmem için desteklerini eksik etmeyen NeuRoLab üyelerine, başta Ahmet Safa ÖZTÜRK, Alp DEMİREL, Can TEKİN TEMUR, Emin Yusuf AYDIN, Eren Cem GÖKSÜLÜK, Eylül HAYDAROĞLU, Furkan Sabri DİPİ, İzel SOLMAZ, Mehmet MURATOĞLU, Mesut Taner ÇETİN ve Tuğberk Emre DEMİREZER'e tüm yardım ve destekleri için şükranlarımı sunarım.

Tüm bunların yanında bu zorlu süreçte doğrudan ya da dolaylı olarak tezime katkıda bulunmuş Dr. Ömer SEZGİN, Ramazan AKDOĞAN, Emine İKİZ, 233 numaralı oda sakinleri, Leandre 100. Yıl ahalisi ve Can BONOMO'ya teşekkür ederim.

Son olarak, ilk öğretmenlerim, ailem Serpil YILMAZ, Erol YILMAZ ve Oğuzhan YILMAZ'a gösterdikleri sevgi, destek ve her zaman yanımda oldukları için teşekkür ederim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER	vi
TABLolar	ix
ŞEKİLLER	x
KISALTMALAR	xi
1. GİRİŞ	1
1.1. Tezin Kapsamı	2
1.2. Katkılar	3
1.3. Organizasyon	3
2. ARKAPLAN ÖZETİ	5
2.1. Sensörler	5
2.1.1. IMU, GNSS ve RTK	5
2.1.1.1. IMU,	5
2.1.1.2. GNSS	6
2.1.1.3. RTK (Real-Time Kinematic)	6
2.1.1.4. Genel Karşılaştırma:	8
2.1.2. LiDAR	8
2.2. 3 Boyutlu LiDAR Odometrisi	11
3. LİTERATÜR TARAMASI	14
3.1. YALIN LiDAR ODOMETRİSİ YÖNTEMLERİ	14
3.1.1. LOAM	14
3.1.1.1. LOAM Öznitelik Çıkarımı	15
3.1.1.2. LOAM Karşılıklık Bulma ve LiDAR Odometrisi	16
3.1.1.3. LOAM Haritalama ve Kestirim İyileştirme	19
3.1.2. LeGO-LOAM	20

3.1.3. KISS-ICP	22
3.2. FÜZYONLU LiDAR ODOMETRİSİ YÖNTEMLERİ	23
3.2.1. P ³ -LOAM	23
3.2.2. KINEMATIC-ICP	24
3.2.3. LIO-EKF	26
3.2.4. LIO-Fusion	27
3.3. LiDAR VERİSETİ ÇALIŞMALARI	28
3.3.1. MULRAN VERİSETİ	28
3.3.2. M2DGR VERİSETİ	28
4. DONANIM ve VERİ TOPLAMA	30
4.1. Donanım Elemanları	30
4.1.1. Tek Kart Bilgisayar	32
4.1.2. LiDAR	32
4.1.3. Stereo Kamera	33
4.1.3.1. ZED ve ZED 2	34
4.1.4. INS Sensörleri	34
4.2. Veri Toplama	36
4.2.1. Kapalı Alan Veri Seti	36
5. LOAM KOVARYANS KESTİRİMİ	39
5.1. Kullanılan Hata Metrikleri	41
5.1.1. Ortalama Maliyet	41
5.1.2. Hausdorff Mesafesi	41
5.2. Dönüşüm Varyansından Global Varyans Hesaplama	42
6. 1 BOYUTTA SENSÖR FÜZYONU	44
6.1. Kullanılan Metotlar	44
6.1.1. Çevrimdışı Metotlar	44
6.1.1.1. Sabit α	44
6.1.1.2. Dinamik α	44
6.1.2. Çevrimiçi Metotlar	45
6.1.2.1. Global ortalama	45

6.1.2.2. Dönüşüm ortalama	46
6.2. Sonuçlar	46
7. 3 BOYUTTA SENSÖR FÜZYONU	50
7.1. Tek Yönlü Hausdorff Mesafesi	50
7.2. Kullanılan Metotlar	51
7.2.1. Optimal Ortalama	52
7.2.2. Global Ortalama.....	53
7.2.3. Dönüşüm Ortalama	54
7.2.3.1. Kalman Filtresi dönüşüm varyansı alt limiti	55
7.2.3.2. Kalman Filtresi dönüşüm varyansı üst limiti	55
7.2.4. Düzeltilmiş Ortalama	58
7.3. Sonuçlar	59
8. BAĞLAŞIK METOTLAR ve İLERİ ÇALIŞMA	67
8.1. Kalman Filtresi ile İklendirilmiş Haritalama	67
8.2. Kalman Filtresi Sonucu Destekli Maliyet Fonksiyonu	69
8.3. İleri Çalışma	71
9. SONUÇ	73

TABLULAR

	<u>Sayfa</u>
Tablo 4.1 SSV1 ve SSV2 Birimlerinin Karşılaştırılması	31
Tablo 4.2 Jetson Nano, Xavier NX ve Orin NX Karşılaştırması	33
Tablo 4.3 Velodyne VLP-16 Lidar Sensörü Teknik Özellikleri	34
Tablo 4.4 ZED ve ZED2 Kameralarının Karşılaştırması	35
Tablo 4.5 MTi-7 ve ArNav S1G Sensörlerinin Karşılaştırması	36
Tablo 6.1 Kapalı Alan Veriseti için Ortalama Görece Hata Tablosu	47
Tablo 6.2 Kapalı Alan Veriseti için Ortalama Mutlak Hata Tablosu	47
Tablo 6.3 Kapalı Alan Veriseti deney 11_07 için Görece Hata Tablosu	48
Tablo 6.4 Kapalı Alan Veriseti deney 11_07 için Mutlak Hata Tablosu	48
Tablo 7.1 Riverside 3 Veriseti için Mutlak Hata Tablosu	62
Tablo 7.2 Riverside 3 Veriseti için Görece Hata Tablosu	62
Tablo 7.3 DCC2 Veriseti için Mutlak Hata Tablosu	64
Tablo 7.4 DCC2 Veriseti için Görece Hata Tablosu	64
Tablo 8.1 Mutlak Hata Değerleri	71
Tablo 8.2 Görece Hata Değerleri	71

ŞEKİLLER

	<u>Sayfa</u>
Şekil 3.1	Farklı geometrilere sahip şekiller üzerindeki noktaların düzgünlük dağılımları. 16
Şekil 3.2	MULRAN Riverside 2 Veriseti üzerinde Öznitelik Çıkarımı sonucu... 17
Şekil 4.1	SSv1 (sol) ve SSv2 (sağ)..... 30
Şekil 4.2	SSv2 blok diyagramı. 31
Şekil 4.3	Kapalı Alan Deney Ortamı 37
Şekil 5.1	2024 Paris Olimpiyatları 10 metre Havalı Tabanca kategorisi ön eleme müsabakası sonuçları. 39
Şekil 5.2	LOAM Hareket Güncellemesi 42
Şekil 6.1	Kapalı Alan Veriseti Deney 11_07 Kestirim Sonucu 49
Şekil 7.1	Aynı doğru üzerinde bulunan Harita ve Ölçüm kenar nokta bulutları üzerinden Hausdorff Mesafesinin hesaplanması. 50
Şekil 7.2	X_{Opt} 'un Hesaplanması 53
Şekil 7.3	Global Ortalama metodu sonuç uzayı 54
Şekil 7.4	Riverside 3 Veriseti için Global ve Dönüşüm Ortalama Metotlarında kullanılan LiDAR Sensörü Ağırlıkları 59
Şekil 7.5	Riverside 3 Veriseti için Global ve Dönüşüm Ortalama Metotlarında kullanılan LiDAR Sensörü Ağırlıklarının Spektrumu 60
Şekil 7.6	Riverside 3 Veriseti Sonucunun Kuşbakışı Görüntüsü 63
Şekil 7.7	Riverside 3 Veriseti için Mutlak (sol) ve Görece (sağ) Pozisyon Hatası Sonuçları 63
Şekil 7.8	DCC 2 Veriseti 3B Odometri Sonucu ve Sonucun Kuşbakışı Görüntüsü 65
Şekil 7.9	DCC 2 Veriseti için Mutlak ve Görece Pozisyon Hatası Sonuçları 66
Şekil 8.1	Doğru ve Dağınık pozisyanlarda iklendirilmiş ICP kestirimleri 68
Şekil 8.2	Örnekleme Sayısının Yerel Minimumlara Etkisi 69

KISALTMALAR

EKF	:	Extended Kalman Filter - Geniřletilmiř Kalman Filtresi
GLONASS	:	GLO balnaya NA vigatsionnaya S putnikovaya S istema - Kresel Uydu Navigasyon Sistemi
GNSS	:	Global Navigation Satellite Systems - Kresel Konum Belirleme Sistemi
GPS	:	Global Positioning System - Kresel Konum Belirleme Sistemi
ICP	:	Iterative Closest Point - İteratif En Yakın Nokta
IMU	:	Inertial Measurement Unit - Atalet lm Birimi
INS	:	Inertial Navigation System - Ataletsel Navigasyon Sistemi
LIO	:	LiDAR-Inertial Odometry - LiDAR Ataletsel Odometri
LiDAR	:	Light Detection And Ranging - Iřık Algılama ve Menzil Belirleme
LOAM	:	LiDAR Odometry And Mapping - LiDAR Odometrisi ve Haritalama
LM-Optimizasyonu	:	Levenberg – Marquardt Optimizasyonu
MMD	:	Maximum Mean Discrepancy - Maksimum Ortalama Farklılık
MSE	:	Mean Squared Error - Ortalama Kare Hatası
MVUE	:	Minimum Variance Unbiased Estimator - En Kk Varyanslı Yansız Kestirici
PPP	:	Precise Point Positioning - Hassas Nokta Konum Belirleme
RAIM	:	Receiver Autonomous Integrity Monitoring - Alıcı Otonom Btnlk İzleme
RANSAC	:	RAN dom SA mple C onsensus - Rastgele rnek Mutabakatı
RMSE	:	Root Mean Squared Error - Karekk Ortalama Kare Hatası
ROS	:	Robot Operating System - Robot İřletim Sistemi

- RTCM** : **R**adio **T**echnical **C**ommission for **M**aritime **S**ervices
- Denizcilik Teknik Radyo Komisyonu
- RTK** : **R**eal-**T**ime **K**inematic - Gerçek Zamanlı Kinematik
- SLAM** : **S**imultaneous **L**ocalization **A**nd **M**apping
- Eşzamanlı Konumlama ve Haritalama
- TUSAGA** : **T**ürkiye Ulusal **S**Abit **G**NSS **A**ğı

1. GİRİŞ

Elmer ve Elsie, bir insan tarafından kontrol edilmeden sadece kendi sensör verilerini kullanarak yollarını bulabilen ilk otonom mobil robotlardır[1]. Bu robotların William Grey Walter tarafından 1940'lı yılların sonunda yapılmalarından sonra, daha çeşitli sensörleri kullanarak, daha karmaşık görevleri gerçekleştirebilen birçok robot yapıldı. Bu görevlerden biri de robotun daha önceden hiç bilmediği bir ortamda, bu ortamın haritasını çıkarması ve bu harita üzerinde kendini düzgün bir şekilde konumlandırması, yada daha yaygın bilinen adıyla, eşzamanlı konumlandırma ve haritalama'dır (Simultaneous Localization And Mapping - SLAM). 1990'lı yıllarda üzerinde aktif olarak çalışılmaya başlanan SLAM, robot süpürge gibi ilk örneklerinin gündelik hayata girmesi ile yaygınlık kazanmıştır. Günümüzde otonom araçlar, servis, güvenlik ve denetim robotları gibi birçok alanda kullanılmaktadır.

SLAM, kamera, radar ve LiDAR (Light Detection And Ranging - Işık Algılama Ve Menzil Belirleme) gibi sensörlerden elde edilen verilerin, GNSS (Global Navigation Satellite System - Küresel Konum Belirleme Sistemi), IMU (Inertial Measurement Unit - Atalet Ölçüm Sensörü) ve odometre gibi sensörlerin de füzyonu ile birlikte bir haritaya eklemeli olarak kaydedilmesi ve robotun konumunun bu harita üzerinde kestirilmesini amaçlar. Problem gereği, konumlandırma ne kadar doğru ise, haritaya eklenen verinin doğruluğu, dolayısıyla haritanın doğruluğu o kadar yüksektir. Öte yandan, haritanın doğruluğu ne kadar yüksek ise, sensörlerin kendilerini bu harita üzerinde konumlandırma doğruluğu da o kadar yüksektir. Bu açıdan SLAM, tavuk - yumurta problemi ile benzerlik göstermektedir.

Bu alanda sıkça kullanılan sensör füzyonu yöntemlerinden biri Kalman Filtresi'dir. 1960'ların başında geliştirilen bu yöntem ilk olarak Apollo görevlerinde roketin yörünge tahmini için kullanılmasıyla dikkat çekmiştir. Kalman Filtresi, sistem durumlarını, sistem modelinden elde ettiği tahmin ve sensörlerden gelen ölçümleri, sensör ve sistem gürültülerine göre ters orantılayarak güncelleyen stokastik bir yöntemdir.

LOAM[2] (LiDAR Odometry and Mapping - LiDAR Odometrisi ve Haritalama) ise Ji Zhang tarafından 2014 yılında sadece LiDAR verisi kullanarak SLAM yapan bir yöntemdir. Bunun

için LiDAR verisinin özniteliklerini kullanır. An itibari ile, görsel ve LiDAR odometri algoritmalarını kıyaslamak için yaygın olarak kullanılan KITTI kıyaslama paketinde tüm algoritmalar arasında üçüncü, sadece LiDAR verisi kullanan algoritmalar arasında birinci sıradadır.

LOAM'ın bir eksiği, odometri verisini nümerik hesaplamalar sonucu elde etmesi sonucunda yaptığı kestirim için bir varyans değeri elde etmemesidir ki bu durum diğer ICP (Iterative Closest Point - İteratif En Yakın Nokta)[3] temelli LiDAR odometrisi algoritmalarının[4–7] da muzdarip olduğu bir durumdur. Bu duruma yönelik, ICP temelli metotlar için önerilmiş çeşitli yöntemler[8, 9] mevcuttur. Ayrıca Li ve ark. P³-LOAM[10] çalışmalarında hata yayılımını tekil değer çözümününün (singular value decompozition) Jakobiye'n'i ile modelleyerek varyans kestiriminde bulunmuştur.

Bu tez kapsamında, LOAM algoritmasına Hausdorff Mesafesi'ne dayalı bir varyans kestirimi metodu önerilmiştir. Bu varyans kestirimine sahip LiDAR odometrisi sonucu, diğer sensör ve kestirim sonuçları ile birleştirilmiştir. Böylece, hem Hausdorff Mesafesi yönteminin bu probleme uygunluğu test edilmiş, hem de hata birikiminden muzdarip, parakete hesabı ile ilerleyen LOAM algoritmasının, mutlak pozisyon bilgisi veren GPS (Global Positioning System - Küresel Konum Belirleme Sistemi), sensörü gibi sensörler ile uzun vadedeki sapmasının önüne geçilmeye çalışılmıştır.

1.1. Tezin Kapsamı

Bu tezin amacı, açık alanda hareket eden bir robottan toplanan LiDAR, IMU ve GPS verilerinin stokastik bir düzlemde birleştirilmesi ve bu sensörlerden elde edilen veriler ile hassas bir pozisyon kestirimi gerçekleştirilmesidir. Çalışmada, Hausdorff Mesafesi'ne dayalı bir varyans kestirim yöntemi önerilmiştir. Bu yöntem ile LiDAR odometrisi sonucunda elde edilen veriler diğer sensörlerden gelen bilgilerle birleştirilerek, özellikle uzun vadeli hata birikimlerinin giderilmesi hedeflenmiştir. Ayrıca, GPS gibi mutlak konum bilgisi sağlayan sensörlerin LiDAR verileri ile entegrasyonu sağlanmıştır.

1.2. Katkılar

Bu tez çalışmasının literatüre sağladığı katkılar şu şekilde özetlenebilir:

- Kurulan sensör sistemi ile LiDAR ve Stereo Kamera ile veri toplanmış, bu sistemin IMU, GNSS, RTK ve bacak odometrisi verilerini de içerecek şekilde geliştirilmesi çalışmalarına başlanmıştır.
- Hausdorff Mesafesi'ne dayalı bir varyans kestirim yöntemi önerilmiştir. Bu yöntem ile LiDAR odometrisi sırasında elde edilen sonuçların güvenilirliği belirlenmiştir. Böylece LOAM algoritmasına varyans çıktısı kazandırılmıştır.
- Kalman Filtresinin son iki tahmini arasındaki ötelemenin kovaryansını elde edebilmek için matematiksel bir yöntem önerilmiştir.
- Kullanılan metotlar, GNSS, IMU ve LiDAR verilerini dinamik bir çoklu oranlı yapıda birleştirerek, daha önceki çalışmalarda gözlemlenen uzun vadeli sapmaların önüne geçmiştir.
- Gerçek sensörlerden toplanan verisetleri üzerinde çevrim dışı olarak yapılan denemeler sonucunda, önerilen metodun ayrık sensör verilerini daha düşük hata oranları ve daha stabil pozisyon kestirimi sağlayacak şekilde birleştirebildiği gösterilmiştir.
- Kalman Filtresi ile LOAM algoritması füzyonu için iki farklı metot denenmiş ve sonuçları sunulmuştur.

1.3. Organizasyon

Tezin organizasyonu aşağıdaki gibidir:

- Kısım (1)'de konuya giriş yapılmıştır. Ayrıca tez konusundaki amacımız ve motivasyonumuz yine burada belirtilmiştir.

- Kısım (2)'de tezde kullanılan matematik ve algoritmalar ile ilgili genel bir arka plan bilgisi verilmiştir.
- Kısım (3)'te literatürde bu konu ile ilgili daha önce yapılan çalışmalar anlatılmış, bu çalışmaların varsa tez ile ilgili kısımları detaylandırılmıştır.
- Kısım (4)'te tez sürecinde donanım üzerinde yapılan çalışmalar ve bununla elde edilen veri seti anlatılmıştır.
- Kısım (5)'te varyans kestirimi yaparken kurduğumuz mantık ve bunun için kullandığımız metotlar açıklanmıştır.
- Kısım (6)'da topladığımız veriseti üzerinde 1 boyutta uyguladığımız sensör füzyonu metotları anlatılmış ve elde edilen sonuçlar gösterilmiştir.
- Kısım (7)'de MulRan veriseti üzerinde Hausdorff Mesafesinin varyans değeri olarak kullanılabilirliği ters varyans ile ortalama temelli yöntemler üzerinden karşılaştırılmış ve elde edilen sonuçlar gösterilmiştir.
- Kısım (8)'de denenen bağlaşıklık metotlar açıklanmış ve sonuçları paylaşılmıştır. Ayrıca ileride bu konuda yapılabilecek çalışmalar bu bölümde tartışılmıştır.
- Kısım (9)'da çalışma sonucunda elde edilen sonuçlar değerlendirilmiştir.

2. ARKAPLAN ÖZETİ

2.1. Sensörler

2.1.1. IMU, GNSS ve RTK

2.1.1.1. IMU, yani Atalet ölçüm Birimi, bir cismin açısal ve doğrusal hareketlerini ölçmek için kullanılan bir sensördür. Genellikle üç eksenle ivmelenme (akselerometre), açısal hız (jiroskop) ve bazen manyetik alan (manyetometre) ölçümleri yapar. Bu sensörlerin birleşimi, bir cismin konumu, yönelimi ve hareket durumu hakkında detaylı bilgi sağlar.

Çalışma Prensipleri IMU'nun temel bileşenleri şunlardır:

1. **İvme Ölçer:** Doğrusal ivmelenmeyi ölçer. Bir cismin hızlanmasını ve yerçekimi kuvvetini algılayabilir.
2. **Jiroskop:** Açısal hızları ölçerek cismin dönme hareketlerini belirler.
3. **Manyetometre:** Manyetik alan ölçümü yaparak yön tayini sağlar (pusula gibi çalışır).

Bu sensörlerden elde edilen veriler, genellikle filtreleme algoritmaları (örneğin, Kalman veya Mahony filtresi) kullanılarak işlenir ve cisim hareketlerinin kesin ve güvenilir bir şekilde takip edilmesi sağlanır [11].

Kullanım Alanları

- **Otonom Araçlar:** IMU, araçların hareketlerini algılayarak doğru bir şekilde konumlandırılmasını sağlar.
- **Robotik:** Hareket ve denge kontrolünde kullanılır.

- **Mobil Cihazlar:** Akıllı telefonlar ve tabletlerde ekran döndürme ve hareket algılama gibi işlevler için gereklidir.
- **Havacılık ve Uzay:** Uçaklar, roketler ve uyduların stabilitesini ve yönelimini korumak için kullanılır.

2.1.1.2. GNSS , uydulu tabanlı bir konumlandırma sistemidir ve dünya üzerindeki herhangi bir noktanın kesin konumunu (enlem, boylam ve yükseklik) belirlemek için kullanılır. GNSS sistemi, Dünya'nın yörüngesinde dönen uydular ve bu uydularla iletişim kuran alıcı cihazlardan oluşur.

Çalışma Prensipleri GNSS alıcıları, uydulardan gelen sinyalleri kullanarak konumlarını üçgenleme yöntemiyle hesaplar. Bu yöntem, bir kullanıcının konumunu belirlemek için üç veya daha fazla uydudan gelen sinyallerin zaman farklarını ölçer [12].

Kullanım Alanları

- **Navigasyon:** Araçlar, gemiler ve uçaklar için yol bulma.
- **Arazi ölçümü:** Haritalama ve coğrafi bilgi sistemlerinde kullanılır.
- **Mobil Cihazlar:** Haritalar ve konum tabanlı hizmetler sunar[13].
- **Acil Durumlar:** Kaybolan kişilerin veya araçların konumlarının tespiti.

2.1.1.3. RTK (Real-Time Kinematic) , GNSS teknolojisinin hassasiyetini artırmak için kullanılan bir yöntemdir. Bu yöntem, bir referans istasyonu ve bir mobil alıcı arasında gerçek zamanlı diferansiyel düzeltmeler yaparak santimetre düzeyinde doğruluk sağlar.

Çalışma Prensipleri RTK sistemi, bir baz istasyonu ve bir veya daha fazla hareketli alıcıdan oluşur. Baz istasyonu, uzun süre boyunca aynı pozisyondan ölçüm yapması nedeni ile yüksek hassasiyette konum bilgisine sahiptir. Bu konum bilgisini kullanarak algıladığı GNSS sinyallerindeki hata kaynaklarını (atmosferik gecikmeler, uydu saat hataları gibi) tespit eder. Bu düzeltme verileri, bir radyo bağlantısı veya internet üzerinden mobil alıcıya iletilir. Mobil alıcı, bu verileri kullanarak hassas bir konum belirler. Mobil alıcı hassasiyeti baz istasyonundan uzaklaştıkça azalır. [14].

TUSAGA-Aktif (Türkiye Ulusal Sabit GNSS Ağı - Aktif) Sistemi Türkiye’de RTK sistemini de içeren bir yapı olan TUSAGA-Aktif sistemi kullanılmaktadır. Bu sistem, Türkiye genelinde kurulmuş referans istasyonlarından oluşur ve GNSS (Küresel Navigasyon Uydu Sistemi) sinyallerini kullanarak kullanıcıların yüksek doğruluklu konum bilgisi elde etmesini sağlar. TUSAGA-Aktif, coğrafi bilgi sistemlerinden tarıma, inşaat projelerinden altyapı çalışmalarına kadar birçok alanda yaygın olarak kullanılmaktadır [15, 16].

Hacettepe Üniversitesi, Elektrik ve Elektronik Mühendisliği RTK Sistemi Bu çalışma sürecinde, ArdicLabs tarafından geliştirilen bir RTK sistemi bölüme kurulmuştur.

TUSAGA RTK yayını, RTK gözlem (observation) mesajları olarak RTCM 1004 ve RTCM 1012 mesajlarını içermektedir. Bunlar, sırasıyla GPS ve GLONASS uyduları için çift bant düzeltme bilgileri içerir.

Bizim RTK istasyonumuzun yaptığı yayında ise RTCM 1077, RTCM 1087, RTCM 1097, RTCM 1127 mesajları bulunmaktadır. Bunlar, GPS, GLONASS, Galileo ve BeiDou sistemlerinin uyduları için düzeltme bilgileri içermektedir. Bu mesajlar çok kanallı mesajlar (Multiple Signal Message / MSM) olarak geçmekte olup, şu sinyaller için bilgi taşıyabilmektedir:

- **GPS:** L1, L2, L5
- **GLONASS:** G1, G2

- **Galileo:** E1, E5a, E5b, E6
- **BeiDou:** B1, B2, B3

Bizim istasyonumuzda kullanılan alıcı gereği bu sinyallerden L1, L2, G1, G2, E1, E5b, B1, B2 desteklenmektedir.[17].

Bizim sistemimizin en büyük farkı ise Galileo ve BeiDou sistemlerini de içerdği için kullanılan uydu sayısının iki katından fazla olmasıdır. Bu farklılık şu açıdan önemlidir: RTK'nın çalışabilmesi için istasyon ve alıcının aynı anda aynı uyduları görmesi gerekmektedir. özellikle alıcının az uydu gördüğü senaryolarda, daha fazla GNSS sistemi kullanılabilmesi hem RTK yapılamıyorsa yapabilmemesini sağlayabilmekte, hem de RTK yapılabiliyorsa isabetliliğini arttırmaktadır.

2.1.1.4. Genel Karşılaştırma: IMU, GNSS ve RTK birbirlerini tamamlayıcı teknolojilerdir. IMU hızlı ve bağımsız bir şekilde hareket algılamak, GNSS geniş alanlarda konum bilgisi sağlar. RTK ise hassasiyet gerektiren durumlarda GNSS'in doğruluğunu artırır. Bu sensörlerin birlikte kullanımı, otonom araçlar, robotik sistemler ve haritalama gibi uygulamalarda optimal sonuçlar elde edilmesini sağlar [18].

2.1.2. LiDAR

İlk lazer mesafe ölçüm cihazı, lazerin icadından hemen sonra, 1961 yılında Buddenhagen ve arkadaşları tarafından geliştirilmiştir. Bundan sonraki 5 sene içerisinde ise LiDAR'ın atası denebilecek CoLiDAR ve LiDAR geliştirilmiştir. O dönemde mesafe ölçümünü Radar ile aynı fiziksel prensibi kullanarak yapması nedeniyle lazer ya da ışık Radar'ı olarak da anılan bu sistemin[19] -ki LiDAR (Light Detection And Ranging) ile Radar (Radyo Tespit ve Mesafe Ölçümü - Radio Detection And Ranging) arasındaki isim benzerliği de bu durumla ilişkilidir- Radar'dan en temel farkı kullandığı elektro manyetik dalgaların dalga boyudur. LiDAR radyo dalgası yerine görünür ışık ya da kızılötesi ışık spektrumunu kullanır. Bunun

sonucunda, LiDAR'ın maksimum ölçüm mesafesi Radar'a göre çok daha kısa kalırken, bu kısa mesafelerde Radar'dan çok daha hassas ve doğru ölçüm yapabilir.

LiDAR, üzerinde bulunan lazerden kendi gönderdiği lazer ışınının sensörden çıkış anı ile, ışının cisimden yansıyor yine kendi üzerinde bulunan lazer algılayıcı ile algılandı an arasındaki zaman farkını ölçerek mesafe ölçümü yapmaktadır. Mesafe ile geçen süre arasındaki ilişki, Denklem (1) ile belirlenmektedir. Denklemdaki R elde edilen uzaklık bilgisi, c ışık hızı yani yaklaşık 3×10^8 km/s'dir

$$R = \frac{c \times Zaman}{2} \quad (1)$$

LiDAR'ın birim zamanda yaptığı ölçüm sayısı ve görüş açısı arttıkça, LiDAR'ın içinde bulunduğu çevre daha iyi anlamlandırılabilir ve haritalanabilir. Bu nedenle LiDAR mobil araçlar haricinde uçak, drone gibi uçan araçlarla birlikte orman ya da inşaat sahası gibi geniş alanların haritalandırılmasında da kullanılmaktadır. Ölçüm sayısı ve görüş açısını arttırmak için katıhal tabanlı çözümler olmakla birlikte, mekanik tabanlı çözümler daha eski ve genel olarak daha ucuz olmaları ve katıhal tabanlı LiDAR'ların şu anda mekanikler kadar güçlü lazerler kullanamamaları ve bu durumun sonucunda uzak mesafe performanslarının düşük olması nedeni ile mekanik LiDAR'lar daha yaygın olarak kullanılmaktadır. Mekanik LiDAR'lar ise, döner aynalı ve döner lazerli varyasyonlara sahiptir.

Döner aynalı versiyonlarda sabit lazer ve alıcının karşısına konulan ayna ile gönderilen ışının açısı değiştirilerek, ortamdaki farklı noktalardan ölçüm alınır. Ölçüm mesafesi ve aynanın ölçüm anındaki açısı kullanılarak bu noktanın 2 ya da 3 boyutlu uzaydaki sensör merkezli konumu bulunur.

Döner sensörlü LiDAR'larda ise, kontrollü bir şekilde döndürülen şaft üzerinde bulunan, aynı yöne bakan bir lazer ve sensör grubu, şaftın dönüşü sayesinde dönüş ekseninde ölçüm yapar. 2 boyutlu LiDAR'larda şafta dik olarak bulunan sensör grubu sayesinde, ölçüm anındaki dönüş açısı ve ölçüm mesafesi ile, noktanın LiDAR merkezli 2 boyutlu uzaydaki konumu bulunur. 3 boyutlu LiDAR'larda ise, şaft üzerine, yükseklik açısına sahip sensör

grupları eklenmesi ile, nokta konumu hesabına yükseklik açısı da dahil edilerek 3 boyutlu tarama yapılabilir.

Şu an piyasada bu yöntemleri tek başına kullanan LiDAR'lar olduğu gibi farklı eksenler için farklı yöntemleri kullanan varyasyonları da mevcuttur. Örneğin, döner sensörlü LiDAR'lar, azimut ekseninde yüksek çözünürlüklü ve homojen sonuçlar vermekle birlikte, dikey ekseninde ayrıntı sensörler kullanmaları nedeni ile bu eksenindeki performansları görece zayıf kalmaktadır. Bu nedenle yatay ekseninde şaft üzerinde dönerken, dikey ekseninde çözünürlüğü arttırmak için katıhal teknolojisini kullanan çalışmalar mevcuttur.[20]

LiDAR'lar genel olarak sabit bir frekansta çalışırlar ve yapılan bir tam tarama ile elde edilen görüntüye çerçeve (frame) denir. Bir çerçevenin verisi 2 boyutta uzaklık imgesi (range image) olarak ifade edilebilir. LiDAR yapısı gereği noktasal bir kaynaktan uzaklık ölçümü alan bir sistem olduğu için, bu formatta genel tercih yatay ve dikey eksenleri küresel koordinat sisteminin açı değerleri ile ifade etmektir ancak buradan diğer koordinat sistemlerine de çevrilebilir. Küresel koordinat sisteminden kartezyen koordinat sistemine yapılan dönüşüm için Denklem (2)(3) ve (4) kullanılır. Denklemdeki X, Y, Z sırasıyla sağ elli kartezyen koordinat sisteminde ileri, sol ve yukarı yöndeki konumlara, α ve ω sırasıyla sağ elli küresel koordinat sisteminde yatay ve dikey eksenindeki açılara denk gelmektedir.

$$X = R \times \cos(\omega) \times \sin(\alpha) \quad (2)$$

$$Y = R \times \cos(\omega) \times \cos(\alpha) \quad (3)$$

$$Z = R \times \sin(\omega) \quad (4)$$

Çerçeve verisi 3 boyutta ise genellikle nokta bulutu (Point Cloud) olarak ifade edilir. Bu nokta bulutunda her bir nokta LiDAR ile yapılan bir ölçüme denk gelir. Bir nokta bulutu bir çerçeveden daha küçük ya da birden fazla çerçeveyi içeriyor olabilir. Hatta bir bölgenin haritası, birçok taramanın birleştirilmesiyle oluşturulmuş tek bir nokta kümesi olabilir.

2.2. 3 Boyutlu LiDAR Odometrisi

LiDAR odometrisinin amacı, Denklem (5)'de gösterildiği gibi, ölçüm sonunda elde edilen LiDAR nokta bulutu \mathcal{P}_k 'nin pozisyonu ve referans LiDAR nokta bulutu \mathcal{P}_{ref} 'in pozisyonu arasındaki dönüşümü

$$T : \mathcal{P}_k \rightarrow \mathcal{P}_{ref} \quad (5)$$

bulmaktır. \mathcal{P}_{ref} , Denklem (6)'de olduğu gibi bir önceki LiDAR nokta bulutu ölçümü \mathcal{P}_{k-1} ya da denklem (7)'de olduğu gibi harita nokta bulutu \mathcal{P}_{map} olabilir.

$$T_k : \mathcal{P}_k \rightarrow \mathcal{P}_{k-1} \quad (6)$$

$$T_k^m : \mathcal{P}_k \rightarrow \mathcal{P}_{map} \quad (7)$$

Ancak hata birikimi nedeniyle T_k^m ile başlangıçtan beri elde edilen dönüşümlerin çarpımı zamanla ıraksar. (Denklem (8)) Bu problemin etkisini azaltmak için, harita referansı ile karşılaştırma[2], ya da döngü kapama[21] gibi yöntemler kullanılır.

$$\lim_{k \rightarrow \infty} \|T_k^m - \prod_{i=0}^k T_k\| \rightarrow \infty \quad (8)$$

Jonnavithula ve ark.[22] LiDAR odometrisinde kullanılan veri işleme hattını (pipeline) 5 aşamaya ayırmaktadır. Bunlar, (1) ön işleme (pre-processing), (2) öznitelik (feature) çıkarımı, (3) karşılıklılık (correspondence) bulma, (4) dönüşüm kestirimi ve (5) art işleme (post-processing)'dir.

Ön işleme aşamasında LiDAR nokta bulutları, filtreleme, gruplara ayırma ve dönüşüm gibi aşamalardan geçerek algoritma ile birlikte çalışabilecek şekilde düzenlenir. Eğer başka sensörler ile çalışılıyorsa, diğer sensörlerden elde edilen veriler de genellikle bu aşamada algoritmaya dahil edilir.

LiDAR tüm ölçümlerini aynı anda yapmadığı için, gövde hareketi sonucunda çerçevenin ilk ve son ölçümleri arasında kaymalar (skewing) oluşabilir. Bu kaymaları en aza indirmeyi önceleyerek doğruluğu arttırmayı amaçlayan daha karmaşık çalışmalar[7] olsa da, taramanın başı ve sonu arasındaki hareket küçük olduğu sürece Denklem (9)'de verilen sabit hız modeli de kayma giderme için yeterli olmaktadır. Hatta Vizzo ve ark. çalışmalarında[5] yeterince küçük hareketler için sabit hız modelinin daha iyi sonuç verdiğini söylemektedir.

$$\mathcal{P}_{k,i}^L = \frac{t_{k,i} - t_k}{t - t_k} T_k^L \mathcal{P}_i^{L_{skewed}} \quad (9)$$

Öznitelik çıkarımı aşamasında nokta bulutu içindeki noktaların birbirlerine göre konumları ya da bulut içerisinde oluşturdukları gruplar göz önünde bulundurularak bulutun öznitelikleri bulunur. Bu öz nitelikler, noktaların oluşturduğu düzlem ve doğru parçaları[2], dağılımlar[4][23] ya da çevredeki maksimum yükseklikler[21] gibi özellikler olabilir.

Karşılıklık bulma adımında, bir önceki adımda bulunan öznitelikler ile referans nokta kümesindeki öznitelikler eşleştirilir. Bu eşler kullanılan eşleştirme algoritması (RANSAC[24] (Random Sample Consensus - Rasgele Örneklem Mutabakatı), ICP[3], Levenberg–Marquardt Optimizasyonu (LM-Optimizasyonu)[25] gibi) ve maliyet fonksiyonuna göre değişebilir. Maliyet fonksiyonu, en yakın noktaya olan uzaklık gibi basit bir fonksiyon[5][6] olabileceği gibi daha karmaşık formlarda da olabilir. Örneğin LOAM[2] ölçüm nokta bulutundaki öznitelik noktalarının referans nokta bulutundaki doğru ve düzlem parçalarına dik uzaklıklarını kullanmaktadır.

Dönüşüm kestirimi aşamasında, eşleştirme algoritması, maliyet fonksiyonu ile elde edilen hatanın en küçük değerini veren dönüşümü bulmak için kullanılır. Bu süreçte tüm eksenler aynı anda optimize edilmek zorunda değildir, LeGO-LOAM[26] gibi parçalı optimizasyon yapan yöntemler de mevcuttur.

Art işleme adımında, Dönüşüm kestirimi sonucunda elde edilen dönüşüm (T_k^{k-1}), daha önce elde edilen dönüşümler ile birleştirilerek global pozisyon elde edilir. Bu dönüşüm bir önceki pozisyondan şu anki pozisyona gelmek için yapılması gereken optimum dönüşümdür. Ayrıca bu aşamada global pozisyon kestirimini iyileştirmek için döngü kapama[21], harita ile karşılaştırma [2, 26], çözünürlük artırma [27, 28] veya diğer sensörlerle füzyonlama gibi işlemler yapılabilir.

3. LİTERATÜR TARAMASI

3.1. YALIN LiDAR ODOMETRİSİ YÖNTEMLERİ

Bu kısımda sadece LiDAR verisi ile yüksek hassasiyette sonuç veren algoritmalara yer verilmiştir. LOAM[2] ve LeGO-LOAM[26] opsiyonel IMU füzyonu ve ya döngü kapama becerilerine de sahipken, literatüre asıl katkıları LiDAR odometrisi konusunda olduğu ve bu beceriler aktif değilken de başarılı sonuç verdikleri için bu başlık altında değerlendirilmişlerdir.

3.1.1. LOAM

LOAM algoritmasının literatüre iki büyük katkısı vardır. Bunlardan birincisi, son LiDAR çerçevesinde elde edilen tüm noktalar yerine, çevrede bir düzlem ya da doğru parçasına ait olabilecek noktaları, önceki çerçeve ya da haritada belirlenmiş düzlem ya da doğru parçasına yerleştirmeye çalışarak çözülmesi gereken optimizasyon probleminin uzayını büyük oranda küçültmesidir.

İkincisi ise problemi ilk olarak, son ve sondan bir önceki LiDAR nokta kümelerini (\mathcal{P}_k ve \mathcal{P}_{k-1}) karşılaştırarak yüksek frekans ve düşük doğruluğa sahip ön dönüşüm kestirimi yapar (Denklem (16)). Daha sonra ise daha düşük frekansta son nokta bulutunu (\mathcal{P}_k) oluşturduğu harita nokta bulutu \mathcal{P}_{map} ile karşılaştırarak o anki nokta bulutu için yaptığı dönüşüm kestirimini günceller ve güncel nokta bulutunu bu kestirimi (T_k^{map}) kullanarak haritaya ekler. Ayrıca bu dönüşümü bir dahaki haritalama işlemine kadar Denklem (20) ön dönüşüm kestiriminin doğruluğunu arttırmak için kullanır.

Özellikle makalenin çıktığı dönemde bilgisayarların işlem kapasitesinin tüm nokta bulutunu gerçek zamanlı olarak optimize etmek için yeterli olmadığı ve yüksek hassasiyetli nokta bulutu haritalarının çevrim dışı olarak yapıldığı dönemde gerçek zamanlı bir algoritma olarak başarılı sonuçlar vermiş sonrasında birçok varyasyonu ortaya çıkmıştır.

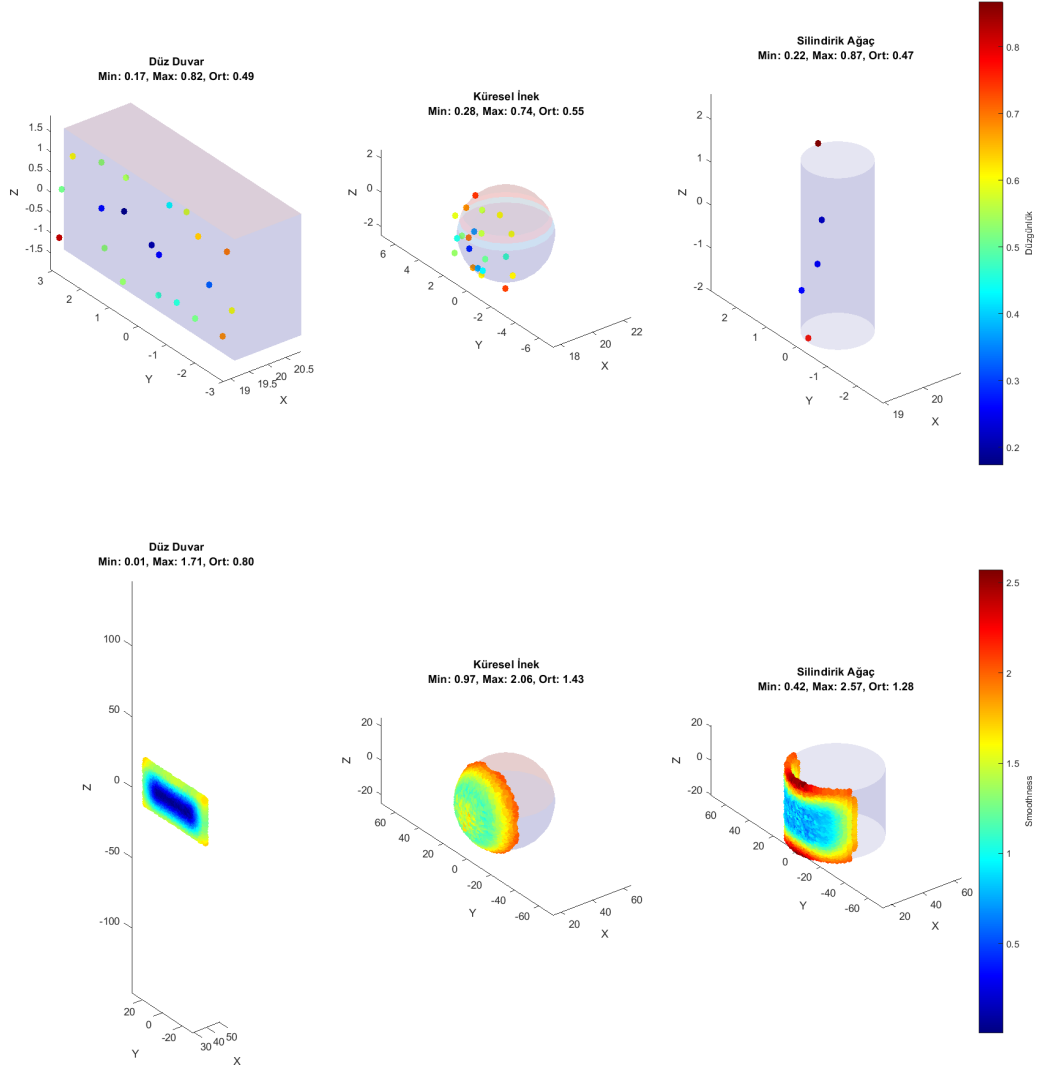
3.1.1.1. LOAM Öznitelik Çıkarımı LOAM, yeni bir LiDAR nokta bulutu mesajı geldiğinde (\mathcal{P}_k), önce bu nokta bulutundaki her bir nokta için düzgünlük (smoothness) puanlaması yapar. Bu puanlama için Denklem (10)'i kullanır. Denklemdaki c düzgünlük puanını, $X_{(k,i)}^L$ puanı hesaplanan noktanın LiDAR koordinat sistemine göre konumunu, S bu noktanın komşuluğundaki diğer noktaların oluşturduğu kümeyi ve $X_{(k,j)}^L$, S içerisindeki j 'inci noktanın LiDAR koordinat sistemine göre koordinatlarını ifade etmektedir.

$$c = \frac{1}{|S| \cdot \|X_{(k,i)}\|} \cdot \sum_{j \in S, j \neq i} \| (X_{(k,i)} - X_{(k,j)}) \| \quad (10)$$

Bu puanlama sonucunda, duvar gibi bir düzlemin merkezinde yer alan noktalar daha düşük puan alırken düzlemlerin kenarında ya da ağaç, sütun gibi tek eksenle uzayan objeler üzerinde bulunan noktalar yüksek puan almaktadır. Bu puanlamayı daha iyi görmek için çevrede düz bir duvar, bir ağaç ve iki duruma da uymayan bir inek olduğu bir senaryo hayal edelim. Ancak problemi daha da basitleştirmek için, duvarın bir dikdörtgen prizması, ağacın bir silindir, ineğin ise bir küre olduğunu farz edelim. Şekil (3.1)'de bu senaryoyu görebiliriz. Buradaki noktalar, objelerin yüzeyinde eşit aralıklarla seçilen pozisyonlara normal dağılımlı gürültü eklenmesi ile elde edilmiştir.

Şekil (3.1)'de de görülebileceği üzere objelerin merkezinde yani daha düzgün yerlerde yer alan noktalar daha düşük düzgünlük puanına sahipken, kenarlarında yer alan noktaların puanı daha yüksektir. Öte yandan küresel ineğimizde görülebileceği üzere diğerlerine göre düzensiz objelerde bulunan noktalar daha ortalama değerler alma eğilimindedir. Objelerin boyutları ya da obje üzerinden alınan ölçüm sayısı arttıkça bu durum daha net olarak gözlemlenebilir.

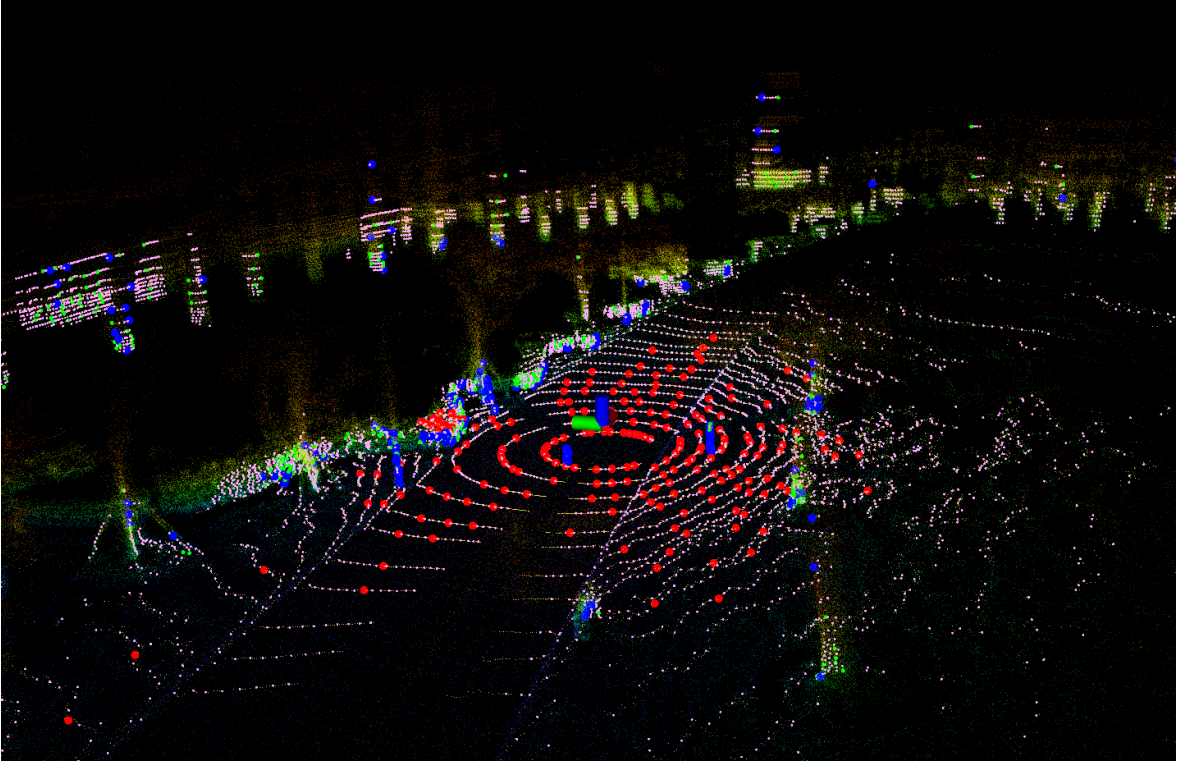
Tüm noktaların puanlaması yapıldıktan sonra belirli bir eşik değerinin altında kalan noktalar düzlemsel nokta bulutu \mathcal{H}_k^L , daha büyük bir eşik değerinin üstünde kalan noktalar ise kenar nokta bulutu \mathcal{E}_k^L içerisine eklenir. Şekil (3.2)'de bu nokta bulutlarının RViz'de görselleştirilmiş halini görebiliriz. Eğer noktanın ölçümü sırasındaki yansıma açısı çok büyükse (bulunduğu düzlem ölçüm yönüne neredeyse paralel ise) ya da nokta, daha öndeki



Şekil 3.1 Farklı geometrilere sahip şekiller üzerindeki noktaların düzgünlük dağılımları.

bir objenin hemen arkasında kaldığı için noktanın komşuluğunun bir kısmı öndeki objenin gölgesinde kalıyorsa bu noktalar bulutlara dahil edilmez.

3.1.1.2. LOAM Karşılıklık Bulma ve LiDAR Odometrisi Bu adımda, k 'inci taramada elde edilen \mathcal{H}_k^L ve \mathcal{E}_k^L 'deki her bir nokta için, \mathcal{H}_{k-1}^L ve \mathcal{E}_{k-1}^L 'deki karşılıklar bulunur. Daha sonra ise bu noktalar üzerinden hesaplanan maliyet fonksiyonunu minimize eden dönüşüm T_k^{k-1} LM-Optimizasyonu kullanılarak hesaplanır. Bu adım iteratif bir adımdır ve n 'inci



Şekil 3.2 MULRAN Riverside 2 Veriseti üzerinde Öznitelik Çıkarımı sonucunu RViz görüntüsü.

En yüksek puana sahip kenar noktaları mavi, daha düşük puana sahip kenar noktaları ise yeşil kürelerle gösterilmiştir. En düşük puana sahip düzlem noktaları kırmızı, daha yüksek puana sahip yüzey noktaları ise pembe renktedir.

iterasyonun sonunda \mathcal{H}_k^L ve \mathcal{E}_k^L bulutlarındaki noktaların konumu Denklem (11) ve Denklem (12) kullanılarak güncellenir.

$$\tilde{\mathcal{H}}_{k,n}^L = \tilde{T}_{k,n}^{k-1} \mathcal{H}_k^L \quad (11)$$

$$\tilde{\mathcal{E}}_{k,n}^L = \tilde{T}_{k,n}^{k-1} \mathcal{E}_k^L \quad (12)$$

Kenar noktası için karşılık bulma ve maliyet fonksiyonu hesaplama adımında karşılık bulmak istediğimiz nokta $\tilde{X}_{k,i}^L$ olsun. İlk olarak \mathcal{E}_{k-1}^L içerisindeki $X_{k,i}^L$ 'e en yakın nokta $X_{k-1,j}^L$ bulunur. Daha sonra ise bu noktanın elemanı olduğu kenarın doğrultusunu bulmak için \mathcal{E}_{k-1}^L içerisindeki $X_{k-1,j}^L$ 'e en yakın nokta $X_{k-1,l}^L$ bulunur.

Noktalar bulunduğundan sonra bunlar Denklem (13)'ye yerleştirilerek $\tilde{X}_{k,i}^L$ noktasının maliyeti hesaplanır. Bu denklem, $\tilde{X}_{k,i}^L$ noktasının $(X_{k-1,j}^L, X_{k-1,l}^L)$ doğrusuna olan dik uzaklığını vermektedir ve amacımız da zaten bu noktanın bu doğruya olan uzaklığını azaltmaktır.

$$d_{\mathcal{E}} = \frac{\| (\tilde{X}_{(k,i)} - X_{(k-1,j)}) \times (\tilde{X}_{(k,i)} - X_{(k-1,l)}) \|}{\| (X_{(k-1,j)} - X_{(k-1,l)}) \|} \quad (13)$$

Düzlem noktası için karşılık bulma ve maliyet fonksiyonu hesaplama adımında karşılık bulmak istediğimiz nokta yine $\tilde{X}_{k,i}^L$ olsun. İlk olarak \mathcal{H}_{k-1}^L içerisindeki $X_{k,i}^L$ 'e en yakın nokta $X_{k-1,j}^L$ bulunur. Daha sonra ise bu noktanın elemanı olduğu düzlemi bulmak için \mathcal{H}_{k-1}^L içerisindeki $X_{k-1,j}^L$ 'e en yakın iki nokta $X_{k-1,l}^L$ ve $X_{k-1,m}^L$ bulunur.

Noktalar bulunduğundan sonra bunlar Denklem (14)'ye yerleştirilerek $\tilde{X}_{k,i}^L$ noktasının maliyeti hesaplanır. Denklemin sağ tarafı $(X_{k-1,j}^L, X_{k-1,l}^L, X_{k-1,m}^L)$ düzleminin normal birim vektörünü vermektedir. Bu normalin $\langle X_{k-1,j}^L, X_{k,i}^L \rangle$ vektörü ile iç çarpımı alınarak, $\tilde{X}_{k,i}^L$ noktasının $(X_{k-1,j}^L, X_{k-1,l}^L, X_{k-1,m}^L)$ düzlemine olan dik uzaklığı elde edilmektedir ve amacımız da zaten bu noktanın bu düzleme olan uzaklığını azaltmaktır.

$$d_{\mathcal{H}} = (\tilde{X}_{(k,i)} - X_{(k-1,j)}) \cdot \frac{(X_{(k-1,j)} - X_{(k-1,l)}) \times (X_{(k-1,j)} - X_{(k-1,m)})}{\| ((X_{(k-1,j)} - X_{(k-1,l)}) \times (X_{(k-1,j)} - X_{(k-1,m)})) \|} \quad (14)$$

LiDAR Odometrisi, bir önceki adımda tüm öz nitelik noktalarının maliyeti bulunduğundan sonra, LM-Optimizasyonu kullanılarak elde edilir (Denklem (15)). n 'inci iterasyonda bu noktalar için en düşük maliyeti veren dönüşüm $\tilde{T}_{k,n}^{k-1}$ 'dir. Burada her bir öz nitelik noktası, problem matrisinin bir sırasına denk gelmektedir. Denklem (11) ve (12) kullanılarak, nokta kümesinin pozisyonu güncellenir. Eğer bu adımda hesaplanan dönüşüm yakınsamışsa (mutlak yerdeğiştirme mesafesi ve dönme açısı bir değerin altındaysa) ya da belirlenen iterasyon limitine ulaşılmışsa, en son elde edilen dönüşüm \hat{T}_k^{k-1} , Denklem (16) kullanılarak önceki çerçevede elde edilen pozisyonun üzerine eklenir ve güncel çerçevenin pozisyonu \hat{T}_k^0

bulunur. Bu kestirim, LiDAR'ın çalışma frekansıyla aynı frekansta, düşük doğruluğa sahip bir kestirimdir.

$$\min_{\mathbf{T}_k} \sum (d_{\mathcal{E}}^2 + d_{\mathcal{H}}^2) \quad (15)$$

$$\hat{T}_k^0 = \hat{T}_{k-1}^0 \hat{T}_k^{k-1} \quad (16)$$

3.1.1.3. LOAM Haritalama ve Kestirim İyileştirme LOAM'ın haritalama adımı, odometri adımından daha düşük frekansta gerçekleşir. Haritaya eklenecek LiDAR çerçevesi, 3.1.1.1. Öznitelik Çıkarımı adımında düzlemsel (\mathcal{H}_k^L) ve kenar (\mathcal{E}_k^L) nokta bulutu olarak kümelere ayrılmış bir şekilde ele alınır. Ancak bu adımda 3.1.1.2. Karşılıklık Bulma ve LiDAR Odometrisi adımındaki gibi referans olarak bir önceki nokta bulutu yerine, daha önceki haritalama adımlarında oluşturulan tüm harita (\mathcal{P}^{map} , \mathcal{H}^{map} ve \mathcal{E}^{map}) kullanılır. Bu nedenle, \mathcal{H}_k^L ve \mathcal{E}_k^L bir önceki kestirim iyileştirme aşamasında Denklem (20) ile elde edilen pozisyon $\hat{T}_{k+i(i < p)}^{map}$ 'ye taşınır. (p değeri burada bir periyottaki LiDAR çerçevesi sayısını ifade etmektedir.) Bunun sonucunda Denklem (11) ve (12), Denklem (17) ve (18) formunu alır. Yeni başlangıç noktasına göre olan dönüşüm, yine Denklem (13), (14) ve LM-Optimizasyonu kullanılarak kestirilir.

$$\tilde{\mathcal{H}}_{k,n}^{map} = T_{k-p}^{map} \tilde{T}_{k,n}^{k-p} \mathcal{H}_k^L \quad (17)$$

$$\tilde{\mathcal{E}}_{k,n}^{map} = T_{k-p}^{map} \tilde{T}_{k,n}^{k-p} \mathcal{E}_k^L \quad (18)$$

Eğer dönüşüm kestirimi $\tilde{T}_{k,n}^{k-p}$ bir değere yakınsarsa, bu dönüşüm değeri Denklem (16)'e benzer şekilde haritalamanın sonucu olarak Denklem (19) kullanılarak hesaplanır. T_k^{map} hesaplandıktan sonra \mathcal{P}_k^{map} , \mathcal{H}_k^{map} ve \mathcal{E}_k^{map} haritaya eklenir. Haritada homojen nokta dağılımı sağlamak amacı ile eğer bir vokselle içerisinde birden fazla nokta bulunuyorsa bunların ağırlıklı ortalaması alınır.

$$T_k^{map} = T_{k-p}^{map} T_k^{k-p} \quad (19)$$

Kestirim iyileştirme adımında, k 'inci haritalama adımında elde edilen dönüşüm, bir sonraki haritalama adımına ($k + p$) kadar LiDAR Odometrisi adımı sonucunda elde edilen sonucu iyileştirmek için Denklem (20)'deki gibi kullanılır. Burada yapılan işlem, k 'inci adımı başlangıç kabul ettiğimizde, LiDAR Odometrisi adımının hesapladığı dönüşümü, yine bu adımın hesapladığı, \hat{T}_k^0 pozisyonu yerine daha doğru olan T_k^{map} pozisyonunu referans alarak hesaplamaktır.

$$\hat{T}_{k+i}^{map} = T_k^{map} \hat{T}_k^{0^{-1}} \hat{T}_{k+i}^0, \quad i < p \quad (20)$$

3.1.2. LeGO-LOAM

LeGO-LOAM[26] LOAM'ın en yaygın kullanılan varyasyonlarından biridir ve bu tez sürecinde yapılan çalışmalar bu varyasyona dayanmaktadır. Orijinal LOAM'ın üzerine eklediği 3 temel katkısı vardır. Bunlar;

- Öznitelik çıkarımı adımında, yer düzlemini ayırarak bu düzlemi yuvarlanma (roll) ve yunuslama (pitch) yönündeki kestirimde kullanması
- LiDAR Odometrisi adımında dönüşüm kestirimi yaparken tüm pozisyon değerlerini tek seferde hesaplamak yerine, bu problemi parçalara bölerek küçük bir kestirim doğruluğu kaybı karşısında işlem hızında büyük oranda kazanç elde etmesi
- Haritalama adımında algoritmaya döngü kapama becerisi kazandırması

olarak tanımlanabilir.

Yer düzlemini ayırma adımında, kara araçlarının seyir sürecinde genel olarak yuvarlanma ve yunuslama yönlerinde düşük açı değerlerine sahip hareket yapıyor olmalarından faydalanılır. Yuvarlanma ve yunuslama açıları düşük olduğunda LiDAR'ın tarama düzlemi aracın hareket düzlemine paralel olacaktır. Buradan hareketle, nokta bulutu içerisindeki noktalardan dikey açısı belirli bir negatif değer altında olanlar eğer eğer düşük düzgünlük puanına sahiplerse,

bu noktalar büyük ihtimalle yer düzlemine aittir. Bu nedenle Bu iki kriteri de sağlayan noktalar yer nokta bulutu \mathcal{G}_k^L olarak ayrıca sınıflandırılır.

LiDAR odometrisi adımında Optimizasyon problemi 2 aşamada çözülür. İlk olarak yer düzlemini tespit etmek için \mathcal{G}_k^L ve \mathcal{E}_k^L nokta bulutları, Denklem (14) ve (13) ile birlikte kullanılır. Bu bulutlardaki tüm noktalar için maliyet hesaplandıktan sonra elde edilen Jakobiyen matrisi ve maliyet vektörü, tüm dönüşüm değerleri yerine sadece yuvarlanma, yunuslama ve z eksenlerindeki dönüşüm değerlerini kestirmek için kullanılır. Daha sonra ise $\hat{\mathcal{H}}_k^L$ ve Denklem (14) kullanılarak elde edilen maliyet matrisi, x , y ve sapma (roll) eksenlerindeki hareketi kestirmek için kullanılır.

Orijinal LOAM makalesi haritalama adımında sadece dönüşümün sonucunu hafızasında tutmaktadır. Ancak döngü kapama becerisinin için, haritalama adımında elde edilen tüm dönüşümlerin hafızada tutulup, döngü kapama tespiti durumunda belirlenen hatayı, bu dönüşümlere dağıtılması gerekmektedir. LeGO-LOAM bu işlem için faktör graf (factor graph) [29][30] yapısını kullanır. Burada her bir düğüm (node) \mathcal{N}_m , $m \approx k/p$ haritalama adımında kestirimi yapılan pozisyona karşılık gelir. Bir döngü kapama tespit edildiğinde, o ana kadar birikmiş hata o döngü içindeki düğümlere dağıtılır. Bu durumda Denklem (19) ve (20), Denklem (21) ve (22) formunu alır. m ie k/p ifadeleri arasında eşitlik yerine yakınsama kullanılma nedeni, haritalama adımının periyodunun her zaman sabit olmayıp, kestirimin yakınsaması ve, belirli bir limitin üzerinde rotasyon veya yer değiştirme kriterlerinin sağlanması gibi şartlara da bağımlı olmasıdır.

$$T_k^{map} = \prod_{n=1}^m T_{\mathcal{N}_n}^{\mathcal{N}_{n-1}}, \quad m \approx k/p \quad (21)$$

$$\hat{T}_{k+i}^{map} = T_{\mathcal{N}_m}^{map} \hat{T}_k^{0^{-1}} \hat{T}_{k+i}^0, \quad i < p, m \approx k/p \quad (22)$$

Ayrıca, haritalama için referans nokta bulutu oluşturulurken, başlangıçtan beri toplanan tüm nokta bulutları yerine, o anki pozisyona uzaklığı LiDAR'ın görüş mesafesinde olan (örneğin VLP-16 için 100m) düğümler, referans kümesi uzayını küçültüp algoritma hızını

arttırmak ve hafıza ihtiyacını azaltmak amacıyla, her vokselde tek bir nokta bulunacak şekilde birleştirilerek kullanılır.

3.1.3. KISS-ICP[5]

KISS-ICP, sade ve etkili bir Iterative Closest Point (ICP) algoritması sunarak gerçek zamanlı SLAM uygulamalarında performans ve sadelik arasında bir denge kurmayı hedefler. Bu algoritma, yalnızca temel ICP adımlarını kullanarak, optimizasyon sürecini mümkün olduğunca basitleştirirken doğruluğu da korur. Makalenin ana katkıları şu şekilde özetlenebilir:

- **Basitlik:** Algoritma, karmaşık optimizasyon tekniklerinden ve ek parametrisasyonlardan kaçınarak yalnızca temel ICP adımlarını uygular. Bu, hem anlaşılabilirliği artırır hem de uygulamayı kolaylaştırır.
- **Hız:** Yüksek işlem hızına sahip olan KISS-ICP, gerçek zamanlı haritalama ve yerleştirme uygulamalarında etkili bir şekilde çalışır.
- **Genel Performans:** KISS-ICP, hem iç hem de dış mekanlarda geniş bir veri seti üzerinde test edilmiş ve doğruluk, hız ve kararlılık açısından başarılı sonuçlar elde etmiştir.

KISS-ICP algoritması, iki temel aşamadan oluşur:

1. **Nokta Eşleştirme:** Algoritma, kaynak noktalarının hedef noktalara en yakın komşu eşleştirmelerini bulur. Bu süreçte, noktaların öklid mesafesi kullanılarak en uygun eşleşmeler tespit edilir:

$$\min_i \| \mathbf{p}_i - \mathbf{q}_j \|_2, \quad (23)$$

burada \mathbf{p}_i kaynak noktasını ve \mathbf{q}_j hedef noktasını temsil eder.

2. **Rijit Dönüşüm Hesaplama:** Eşleşen noktalar kullanılarak optimal bir dönüşüm matrisi \mathbf{T} hesaplanır. Bu dönüşüm matrisi, hem bir döndürme matrisi \mathbf{R} hem de bir öteleme vektörü \mathbf{t} içerir:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}. \quad (24)$$

Dönüşüm, eşleşen noktalar arasındaki hata fonksiyonunu minimize ederek elde edilir:

$$\min_{\mathbf{R}, \mathbf{t}} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_j\|_2^2. \quad (25)$$

KISS-ICP'nin başarısı, bu basit fakat etkili adımların optimizasyonunda yatmaktadır. Ek olarak, algoritmanın uygulaması hem yazılım hem de donanım açısından düşük maliyetlidir, bu da onu geniş bir kullanım yelpazesi için ideal hale getirir. Algoritmanın SLAM ve otonom navigasyon gibi alanlardaki potansiyeli, doğruluğu ve hızından kaynaklanmaktadır. Ayrıca, KISS-ICP'nin açık kaynaklı olarak sunulması nedeniyle algoritma kısa sürede geniş bir topluluğa ulaşmıştır. Bu sayede, algoritma daha çeşitli senaryolarda test edilmiş ve topluluk tarafından geliştirilmeye devam edilmiştir.

3.2. FÜZYONLU LiDAR ODOMETRİSİ YÖNTEMLERİ

3.2.1. P³-LOAM[10]

P³-LOAM, LiDAR-SLAM ve PPP tekniklerini gevşek (loosely coupled) bir şekilde birleştirerek, özellikle kentsel kanyon ortamlarında küresel konumlandırma doğruluğunu artıran bir SLAM sistemi sunar. Sistem, LiDAR-SLAM ve PPP arasındaki entegrasyonu güçlendirmek için iki ana katkı sağlar:

- **LiDAR-SLAM Kovaryans Tahmini:** Sistem, ICP algoritmasının SVD tabanlı bir hata yayılım modeliyle LiDAR-SLAM'in konumlandırma kovaryansını tahmin eder. Bu yaklaşım, çoklu sensör entegrasyonu için daha hassas ağırlıklandırma sağlar.

- **GNSS RAIM Destekli LiDAR-SLAM:** GNSS gözlemlerindeki yanlış verileri elimine etmek için, LiDAR-SLAM'den alınan sonuçlarla GNSS RAIM (Receiver Autonomous Integrity Monitoring) algoritması desteklenir. Böylece, GNSS'nin güvenilirliği ve doğruluğu artırılır.

P³-LOAM, bir ön yüz (frontend) ve bir arka yüz (backend) yapısına sahiptir. Ön yüzde, LiDAR nokta bulutlarının kaydı ve PPP sonuçları işlenirken, arka yüzde 15 serbestlik dereceli bir değişken optimizasyonuyla tüm sensörlerden gelen veriler entegre edilir. Deneyler, P³-LOAM'ın diğer yöntemlere kıyasla daha yüksek doğruluk ve kullanılabilirlik sağladığını göstermiştir.

Sonuç olarak, P³-LOAM, LiDAR ve PPP'nin özelliklerini etkili bir şekilde birleştirerek, büyük ölçekli ve karmaşık çevrelerde güvenilir bir küresel konumlandırma ve haritalama çözümü sunar.

3.2.2. KINEMATIC-ICP[6]

Kinematic-ICP, tekerlekli mobil robotlar için LiDAR odometri algoritmalarını iyileştirmek amacıyla kinematik kısıtlamaları geleneksel ICP (Iterative Closest Point) algoritmasına entegre eden bir yöntem sunar. Bu yöntem, robot kinematikini hesaba katarak daha doğal ve doğru bir hareket tahmini sağlar. Algoritmanın öne çıkan katkıları şunlardır:

- **Kinematik Entegrasyon:** Algoritma, tekerlekli mobil robotların kinematik modelini kullanarak, LiDAR nokta bulutlarının hizalanmasını optimize eder. Bu, robot hareketinin doğal ve fiziksel olarak anlamlı bir şekilde tahmin edilmesini sağlar.
- **Uyarlanabilirlik:** LiDAR ölçümleri ve tekerlek odometrisinin ağırlıklarını dinamik olarak ayarlayarak, özellik açısından zayıf ortamlarda dahi doğru sonuçlar üretir.
- **Üstün Performans:** Kinematic-ICP, hem iç hem de dış mekanlarda yapılan deneylerde, mevcut yöntemlere kıyasla daha yüksek doğruluk ve kararlılık sağlamıştır.

Kinematic-ICP, hareket tahmini sürecinde iki ana aşama içerir:

1. **Başlangıç Tahmini:** Robotun tekerlek odometrisi, LiDAR verisinin işlenmesi için başlangıç pozisyon tahmini olarak kullanılır. Bu tahmin, aşağıdaki şekilde ifade edilir:

$$\hat{T}_t = T_{t-1}O_t, \quad (26)$$

burada T_{t-1} önceki pozisyonu, O_t ise tekerlek odometrisinden gelen görelî hareket bilgisini temsil eder.

2. **Kinematik Kısıtlı Optimizasyon:** ICP algoritması, robotun kinematik modeliyle uyumlu bir şekilde nokta bulutu hizalaması gerçekleştirir. Bu süreç, aşağıdaki optimizasyon problemiyle ifade edilir:

$$\min_{\Delta u} \chi(\hat{T}_t \oplus \Delta u), \quad (27)$$

burada Δu kinematik düzeltme vektörünü, \oplus ise SE(3) uzayında düzeltme uygulamasını temsil eder.

Algoritmanın başarısı, hem kinematik modelin kullanılması hem de LiDAR verilerinin ve tekerlek odometrisinin dinamik olarak optimize edilmesine dayanmaktadır. Örneğin, tekerlek kayması gibi durumlarda algoritma, LiDAR verisine daha fazla ağırlık vererek bu zorlukları aşabilir. Ek olarak, aşağıdaki regularizasyon terimi, translasyonel kısmı kontrol altında tutar:

$$G(\hat{T}_t \oplus \Delta u) = \chi(\hat{T}_t \oplus \Delta u) + \frac{1}{\beta_t} \Delta x^2, \quad (28)$$

burada β_t , LiDAR ve tekerlek odometrisinin güvenilirliğine bağlı olarak ayarlanır.

Deneysel sonuçlar, Kinematic-ICP'nin hem açık alanlarda hem de büyük ölçekli depo ortamlarında tekerlek odometrisine kıyasla daha doğru sonuçlar ürettiğini göstermektedir. Algoritmanın kodu, topluluk tarafından geliştirilmek üzere açık kaynak olarak sunulmuştur.

Sonuç olarak, Kinematic-ICP, mobil robotların doğal hareketlerini kinematik modelle uyumlu bir şekilde tahmin ederek, LiDAR odometrisi için önemli bir ilerleme sunmaktadır.

3.2.3. LIO-EKF[31]

LIO-EKF, LiDAR ve IMU'yu klasik bir EKF (Extended Kalman Filter - Genişletilmiş Kalman Filtresi) şeması ile birleştirerek yüksek frekanslı ve hassas poz tahmini sağlayan bir LiDAR-inertial odometri sistemidir. Bu sistem, LiDAR taramalarını ve IMU ölçümlerini sıkı bir şekilde birleştirerek düşük sürüklenmeli hareket tahmini sunar. Ana katkılar şu şekilde özetlenebilir:

- **Adaptif Veri Eşleme:** LiDAR gürültüsü, harita kesiklilik hataları ve IMU'dan gelen hareket belirsizliği göz önüne alınarak adaptif bir eşik belirleme yöntemi sunar. Bu yaklaşım, veri eşleme için gereken parametreleri büyük ölçüde azaltır.
- **Basitleştirilmiş Hesaplama:** Klasik EKF şeması, diğer karmaşık optimizasyon yöntemlerine kıyasla daha hızlıdır ve IMU çerçeve oranına yakın bir hızda çalışır.

LIO-EKF'nin metodolojisi üç ana bileşenden oluşur:

1. **Hata Durum Modeli:** IMU ölçümlerini kullanarak robotun pozisyon, hız ve yönelim hatalarını tahmin eden bir hata durumu vektörü tanımlar.
2. **LiDAR Gözlem Modeli:** LiDAR taramalarından gelen nokta bulutu verilerini lokal haritalarla eşleştirir ve poz tahminini günceller.
3. **Adaptif Eşik:** Veri eşleşmeleri için kullanılan eşiği, sensör gürültüsü ve hareket belirsizliklerini dikkate alarak otomatik olarak ayarlar.

Deneysel sonuçlar, LIO-EKF'nin mevcut LiDAR-inertial odometri sistemleriyle benzer doğruluk sağlarken önemli ölçüde daha hızlı olduğunu göstermektedir. Ayrıca, karmaşık ortamlarda ve farklı hareket profillerinde yüksek performans sunarak sistemin genelleme kabiliyetini kanıtlamıştır.

3.2.4. LIO-Fusion[32]

LIO-Fusion, GNSS, yeniden konumlandırma (relocalization) ve tekerlek odometrisi verilerini etkili bir şekilde birleştirerek LIO'yu (LiDAR-Inertial Odometry - LiDAR - Ataletsel Odometri) güçlendiren bir sistem sunar. Bu yöntem, zorlu algı koşullarında robotların 6 serbestlik derecesine sahip hareketlerini güvenilir bir şekilde tahmin etmeyi amaçlar. Algoritma, bir faktör grafik çerçevesi kullanarak farklı sensörlerden gelen görel ve mutlak ölçümleri entegre eder. Bu yaklaşım, sensör verilerindeki bozulmalara ve eksikliklere karşı dirençli bir sistem oluşturur.

Sistemin temel bileşenleri şunlardır:

- **Global Güçlendirme Modülü:** GNSS ve yeniden konumlandırma faktörlerinin güvenilirlik durumunu değerlendirir ve yalnızca sağlıklı olanları entegre ederek küresel hata birikimini azaltır.
- **Yerel Güçlendirme Modülü:** Tekerlek odometrisini desteklemek için yerel odometri faktörünü güçlendirir ve LiDAR bozulmalarına karşı dayanıklılığı artırır.

Algoritmanın çalışmasında öne çıkan aşamalar şunlardır:

1. **Başlangıç:** LiDAR nokta bulutlarından özellik çıkarımı, IMU ön entegrasyonu ve GNSS kalibrasyonu gibi işlemleri içerir.
2. **Faktör Grafik Optimizasyonu:** LiDAR, IMU, GNSS ve tekerlek odometrisi ölçümleri arasında sıkı bir entegrasyon sağlar ve düşük sürüklenmeli odometri sonuçları elde eder.

DeneySEL sonuçlar, LIO-Fusion'ın kentsel ve tehlikeli ortamlar gibi zorlu senaryolarda yüksek doğruluk ve sağlamlık sağladığını göstermiştir. Algoritma, mevcut yöntemlere kıyasla daha düşük hata oranlarıyla birlikte daha doğru haritalama ve konumlandırma sunmaktadır. LIO-Fusion, otonom robotların karmaşık ortamlarda güvenilir bir şekilde çalışmasını sağlamak için önemli bir katkı sağlamaktadır.

3.3. LiDAR VERİSETİ ÇALIŞMALARI

3.3.1. MULRAN VERİSETİ[33]

MulRan, kentsel alanlarda yer tanıma arařtırmaları için radar ve LiDAR verilerinden oluşan çok modlu bir veri seti sunar. Veri seti, uzun süreli yer tanıma ve SLAM algoritmalarının doęrulamasına olanak tanıyacak řekilde, yapısal ve zamansal çeřitlilięi içeren veri örnekleriyle tasarlanmıřtır. Bařlıca katkılar řunlardır:

- **Çoklu Çevre ve Seans:** MulRan, DCC, KAIST, Riverside ve Sejong City gibi farklı çevrelerden, tekrarlı rotaları ve farklı zaman dilimlerini kapsayan veri dizileri içerir.
- **Radar ve LiDAR Entegrasyonu:** 3D LiDAR ve dönen radar sensörlerinden gelen veriler, yer tanıma performansını deęerlendirmek için birlikte sunulmuřtur.
- **Yeniden Ziyaret ve Döngü Algılama:** Çeřitli yapısal senaryolarda yer tanıma ve döngü kapatma doęruluęunu test etmek amacıyla aylar arası yeniden ziyaret senaryoları tasarlanmıřtır.

Veri seti, 6D poz referanslarıyla birlikte, radar ve LiDAR verilerinin polar görüntü ve nokta bulutu formatlarında temsillerini saęlar. Önerilen Scan Context tabanlı radar tanıma yöntemi, radarın uzun menzilli algılama yeteneklerinin yer tanıma görevlerinde LiDAR'ı geride bırakabildięini göstermiřtir. MulRan veri seti, yer tanıma ve SLAM algoritmalarının geliřimi için kapsamlı bir platform sunmaktadır.

3.3.2. M2DGR VERİSETİ[34]

M2DGR, zemin robotları için çoklu sensör ve çoklu senaryo SLAM veri seti sunan, geniş kapsamlı bir veri koleksiyonudur. Bu veri seti, hem iç hem de dış mekanlarda yüksek çeřitlilikte ortamları kapsayarak SLAM algoritmalarının geliřtirilmesi ve deęerlendirilmesi için zengin bir kaynak saęlar. Öne çıkan katkılar řu řekildedir:

- **Zengin Sensör Çeşitliliği:** Veri seti, altı balık gözü kamera, bir gökyüzü kamerası, bir kızılötesi kamera, bir olay kamerası, bir Visual-Inertial (VI) sensör, bir LiDAR, bir IMU ve GNSS-IMU navigasyon sistemlerini içerir. Tüm sensörler kalibre edilmiş ve senkronize edilmiştir.
- **Çeşitli Senaryolar:** Veri seti, asansör kullanımı, tamamen karanlık ortamlar, açık hava-ev içi geçişleri gibi pratikte sıkça karşılaşılan zorlu durumları içeren 36 farklı veri dizisi sunar.
- **Yüksek Kaliteli Gerçek Gezinge Verileri:** Gerçek gezinge (trajectory) verileri, RTK/IMU sistemi, lazer 3D takip cihazı ve hareket yakalama sistemi kullanılarak elde edilmiştir.

M2DGR veri seti, SLAM algoritmalarını hem görsel hem de LiDAR tabanlı yöntemlerle değerlendirmek için bir temel sağlar. Deneyler, mevcut yöntemlerin belirli senaryolarda başarısız olduğunu ve bu nedenle daha sağlam çözümlerin geliştirilmesi gerektiğini ortaya koymaktadır.

Sonuç olarak, M2DGR, zengin sensör çeşitliliği ve senaryolarıyla SLAM araştırmaları için kapsamlı bir referans sunar ve bu alanda ilerlemeyi teşvik eder.

4. DONANIM ve VERİ TOPLAMA

Tez çerçevesinde kullanılan algoritmaları kendi veri setimiz üzerinden test edebilmek için bir veri toplama sistemi kurduk. Şekil (4.1)'de solda bu istemin ilk versiyonu Sensör Sistemi v1 (SSv1), sağda ise ikinci versiyonu Sensör Sistemi v2 (SSv2) görülebilir. SSv2'nin mekanik tasarımı bitmiş olup üretimi devam etmektedir.

SSv1 ile kapalı alan verisetini toplamakla birlikte TUSAGA-Aktif sistemini kullanarak referans pozisyon elde etmeye çalıştığımız ilk açık alan testlerini de gerçekleştirdik. Bu testler sırasında, bina ve ağaçların gökyüzünü kapatması nedeni ile sık sık bağlantı kopması problemleri ile karşılaşmamız nedeni ile daha fazla uydu ve ve bantta düzeltme yapan kendi RTK sistemimize geçene kadar açık alan veri seti hazırlamayı erteleme kararı aldık.



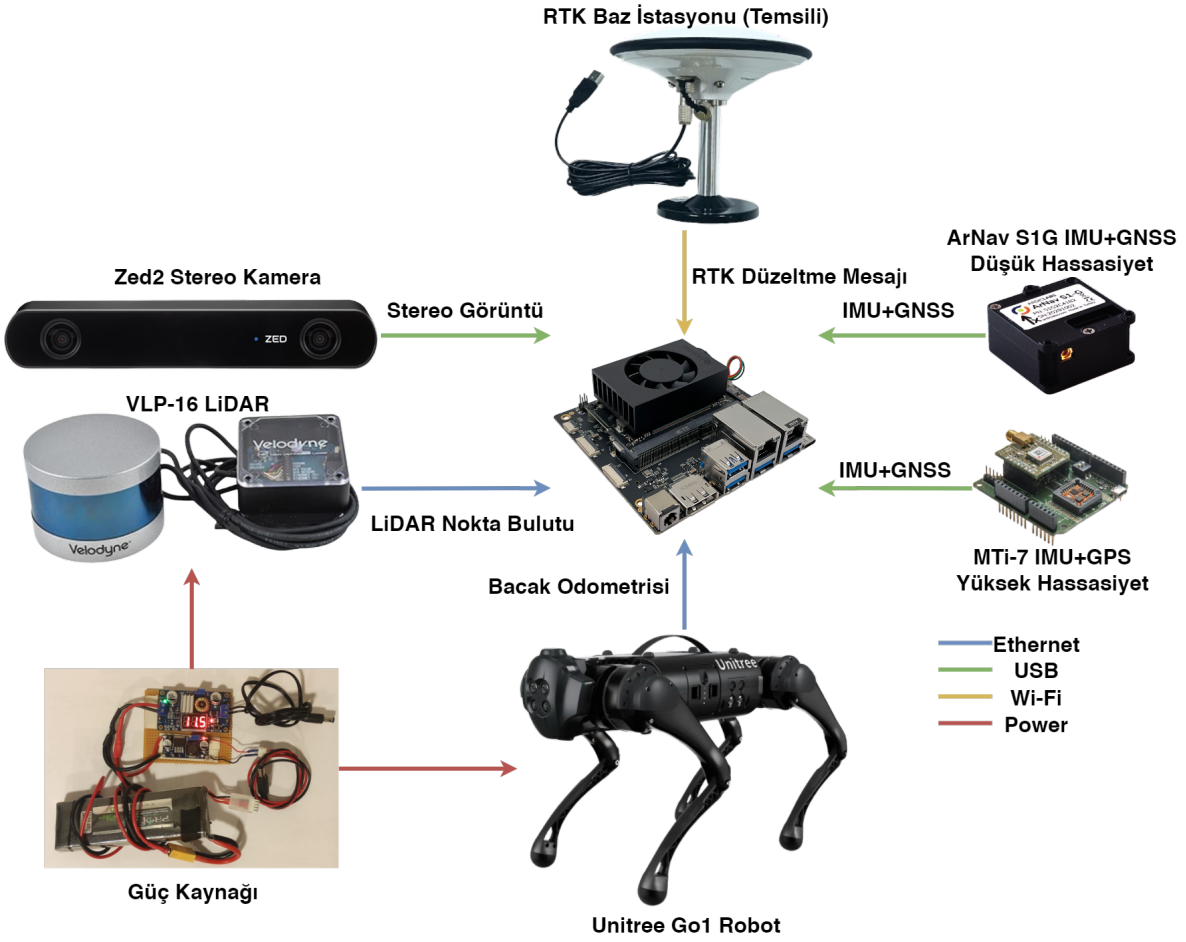
Şekil 4.1 SSv1 (sol) ve SSv2 (sağ)

4.1. Donanım Elemanları

SSv1 ve SSv2, benzer görevleri yapan birimlere sahipken, versiyonlar arasında model değişiklikleri de yapılmıştır. Bunlar Tablo (4.1)'de görülebilir. Şekil (4.2)'de ise SSv2'nin blok diyagramı görülebilir.

Özellikler	SSv1	SSv2
Kullanım Şekli	Elde, Robot Üzerinde	Robot Üzerinde
Tek Kart Bilgisayar	Jetson Nano 4GB	Jetson Orin Nx 16GB
LiDAR	Velodyne VLP-16	Velodyne VLP-16
Stereo Kamera	Zed	Zed2
INS1	ArNav S1A	ArNav S1A
INS2	-	XSens MTI-7
RTK	TUSAGA-Aktif	Hacettepe EE
Odometri	-	Unitree Go1 Edu

Tablo 4.1 SSv1 ve SSv2 Birimlerinin Karşılaştırılması.



Şekil 4.2 SSv2 blok diyagramı.

4.1.1. Tek Kart Bilgisayar

Sensör sisteminde, sensörlerin kontrolü ve sensörlerden gelen verileri kaydetmek için tek kart bilgisayar kullanılmıştır. Versiyon 1’de kapalı ortam veri seti toplanırken bu işlem Jetson Nano ile yapılmıştır. Ancak, daha sonra Jetson Xavier NX’e geçilmiştir. Bunun nedenleri arasında;

- Düşük bellek miktarı nedeniyle derleme sürecinde yaşanan sıkıntılar,
- microSD hafıza kartının veri yazma hızının düşük olması nedeni ile veri kaydı sırasında önbelleğin dolması sonucunda veri atlaması,
- microSD hafıza miktarının SSD’lere göre düşük kalması sonucunda tek seferde alınabilen veri miktarının sınırlı olması,
- Daha eski USB versiyonu nedeniyle, bir önceki madde sonucunda gerçekleşen sürekli hafıza boşaltma işleminin yavaş kalması,

gibi nedenler sayılabilir.

Geliştirmeler genel olarak Jetson Xavier NX üzerinde yapılmış olsa da, hem teknoloji olarak geride kalması, hem de Jetson Orin NX’in uyumlu olduğu Seeed Studio A608 geliştirme kartında bulunan ikinci Ethernet portu sayesinde bilgisayarın, robottun ürettiği bacak odometrisi, görsel odometri ve IMU gibi sensör bilgilerini de kaydedebilmesi amacı ile Jetson Orin NX’e geçilmiştir. Tablo (4.2)’de bu bilgisayarların özellikleri karşılaştırmalı olarak bulunabilir.

4.1.2. LiDAR

Sensör sisteminin tüm versiyonlarında LiDAR olarak Velodyne VLP-16 kullanılmıştır. Piyasadaki birçok 3 boyutlu LiDAR’dan daha az açıda ölçüm almasına rağmen, görece daha ucuz fiyatı nedeni ile çok farklı alanlara yönelik veri setlerinde yaygın olarak bulunan bir sensördür. Örneğin;

Özellikler	Jetson Nano 4GB	Jetson Xavier NX 16GB	Jetson Orin NX 16GB
GPU	128 CUDA Core Maxwell	384 CUDA Core Volta + 48 Tensor Core	1024 CUDA Core Ampere + 32 Tensor Core
CPU	Quad-Core ARM Cortex-A57	6-core ARMv8.2 Carmel	8-core ARM Cortex-A78AE
RAM	4 GB LPDDR4 @ 25.6 GB/s	16 GB LPDDR4x @ 51.2 GB/s	16 GB LPDDR5 @ 68 GB/s
AI Performansı	0.5 TOPS	21 TOPS	70 TOPS
Video Kodlama/ Dekodlama	4K 30 FPS (H.264/H.265)	4K 60 FPS (H.264/H.265)	4K 60 FPS (H.264/H.265)
Enerji Tüketimi	5-10 W	10-15 W	10-25 W
Depolama	microSD / eMMC (SOM varyantı)	16 GB eMMC (SOM)	16 GB eMMC (SOM)
Harici SSD Desteği	USB 3.0 ile uyumlu	NVMe M.2 + USB 3.1	NVMe M.2 + USB 3.2 Gen 2
Ethernet	10/100/1000 Mbps Gigabit LAN	10/100/1000 Mbps Gigabit LAN	10/100/1000 Mbps Gigabit LAN
USB Portları	4x USB 3.0, 1x Micro-USB	1x USB 3.1, 4x USB 2.0	1x USB 3.2 Gen 2, 4x USB 2.0
Diğer Portlar	HDMI, DisplayPort, GPIO, UART	HDMI, DisplayPort, GPIO, UART, I2C	HDMI, DisplayPort, GPIO, UART, I2C
Boyutlar	100 mm x 80 mm	70 mm x 45 mm	70 mm x 45 mm
Çıkış Yılı	2019	2020	2022

Tablo 4.2 Jetson Nano 4GB, Xavier NX 16GB ve Orin NX 16GB Karşılaştırması [35–37]

- Complex Urban Dataset[38] otonom araçlar için,
- ConSLAM[39] inşaat sahaları için,
- Ground-Challenge[40] kara araçlarında zorlu durum testleri için,
- Wild-Places[41] doğal ortamlar için,

VLP-16 LiDAR sensörünü kullanan veri setleridir. Sensörün özellikleri Tablo (4.3)'de görülebilir.

4.1.3. Stereo Kamera

Stereo kameralar, iki farklı perspektiften görüntü alarak derinlik bilgisi çıkaran optik sistemlerdir. İki lens ve sensör yardımıyla, aynı anda elde edilen görüntüler arasında görsel farklılık (disparity) hesaplanır ve bir derinlik haritası oluşturulur. Bu sistemler,

Özellik	Değer
Lazer Kanalları	16
Maksimum Menzil	100 metre
Minimum Menzil	1 metre
Yatay Görüş Açısı (FoV)	360°
Dikey Görüş Açısı (FoV)	30° (±15°)
Açısal Çözünürlük (H)	0.1° - 0.4°
Dikey Açısal Çözünürlük (V)	2°
Tarama Hızı	5 Hz - 20 Hz
Veri Çıkış Hızı	300,000 nokta/saniye
Lazer Dalga Boyu	903 nm
Çalışma Sıcaklığı	-10°C ile +60°C
Güç Tüketimi	8 W
Ağırlık	830 g
Boyutlar	103 mm (Çap) x 72 mm (Yükseklik)
Bağlantı Türü	Ethernet

Tablo 4.3 Velodyne VLP-16 Lidar Sensörü Teknik Özellikleri[42]

robotik, otonom araç navigasyonu ve 3D haritalama gibi alanlarda yaygın olarak kullanılır. Performansı, aydınlatma koşulları ve yüzey dokuları gibi faktörlere bağlıdır; düşük ışık ve tekrarlayan desenler gibi durumlarda sınırlı performans gösterebilir.

4.1.3.1. ZED ve ZED 2 ZED ve ZED 2, yüksek çözünürlükte stereo görüntüler sunarak 3D haritalama ve derinlik algılama sağlayan ileri teknoloji kameralardır. ZED, temel stereo görüntü işleme için kullanılırken, ZED 2 dahili sensörlerle (ör. IMU, barometre) donatılmış ve düşük ışık koşullarında daha hassas sonuçlar sunar. Firma tarafından doğrudan ROS (Robot Operating System - Robot İşletim Sistemi) desteği sağlamaları nedeniyle robotik alanında sıkça tercih edilmektedirler. İki kameranın özellikleri Tablo (4.4)'de görülebilir.

4.1.4. INS Sensörleri

Ataletsel Navigasyon Sistemleri (Inertial Navigation Systems - INS) hareketli bir cismin pozisyon, yön ve hız gibi durumlarını belirlemek için kullanılan sistemlerdir. Bunun için içlerinde bulunan İvme ölçer, Jiroskop, Manyetometre ve GNSS alıcıları gibi sensörleri

Özellikler	ZED	ZED2
Çözünürlük (Stereo)	2.2K: 1080p @ 30 FPS 720p @ 60 FPS WVGA @ 100 FPS	2.2K: 1080p @ 30 FPS 720p @ 60 FPS WVGA @ 100 FPS
Görüş Açısı (FoV)	90°	110°
Kamera Sensörleri	2x CMOS 1/3" sensör	2x CMOS 1/2.3" sensör
Derinlik Algılama Menzili	0.7 m - 20 m	0.3 m - 20 m
IMU Desteği	Yok	Var
Ortam Sensörleri	Yok	Sıcaklık, manyetometre, basınç
Bağlantı Arayüzü	USB 3.0	USB 3.0
Güç Tüketimi	1.2 W	1.7 W
Boyutlar	175 mm x 30 mm x 33 mm	142 mm x 29 mm x 30 mm
Ağırlık	159 g	120 g
Çıkış Yılı	2015	2019

Tablo 4.4 ZED ve ZED2 Kameralarının Karşılaştırması [43, 44]

kullanırlar. Genellikle havacılık, uzay, denizcilik, otomatik ve robotik gibi uygulama alanlarına sahiptir. İvme ölçer, Jiroskop ve Manyetometre sensörleri, başlangıç konumuna göre pozisyonu parakete hesabı ile bulur. Sensör gürültüsü ve yanlılığından (bias) hesaplamada doğal olarak oluşan sapmalar, modern INS sistemlerinde GNSS sinyalleri de füzyon algoritmalarına dahil edilerek giderilir.

SSv2'de bulunan INS sensörleri ve özellikleri Tablo (4.5)'de verilmiştir. Bu sensörlerden daha hassas sensörlere sahip olan Xsens MTi-7 sensör füzyonu algoritmalarında kullanılacak verileri toplamak için kullanılacaktır. ArNav S1G ise tek başına kullanıldığında sensör hassasiyeti daha düşük olmasına rağmen, (daha çeşitli uydu sinyal bandından veri toplayabildiği için) RTK düzeltmeleri ile birlikte referans pozisyon elde etmek için kullanılacaktır.

Özellikler	MTi-7	ArNav S1G
Sensör Paketi	Jiroskoplar, ivmeölçerler, barometre	Jiroskoplar, ivmeölçerler, manyetometre, barometre
GNSS Entegrasyonu	Harici GNSS alıcı arayüzü	Tüm büyük GNSS takım uydularını destekleyen entegre GNSS
Doğruluk	1.5° yön, 1 m konum DHİ	≤1 m konum
Çıkış Hızı	1 kHz'e kadar	230 Hz'e kadar
Uygulamalar	Gömülü sistemler, robotik	Kara, deniz ve hava sistemleri
Yazılım Desteği	SDK, ROS, MATLAB	Özel ArNavPro yazılımı ROS desteği labımız tarafından sağlanmıştır

Tablo 4.5 MTi-7 ve ArNav S1G Sensörlerinin Karşılaştırması [45, 46]

4.2. Veri Toplama

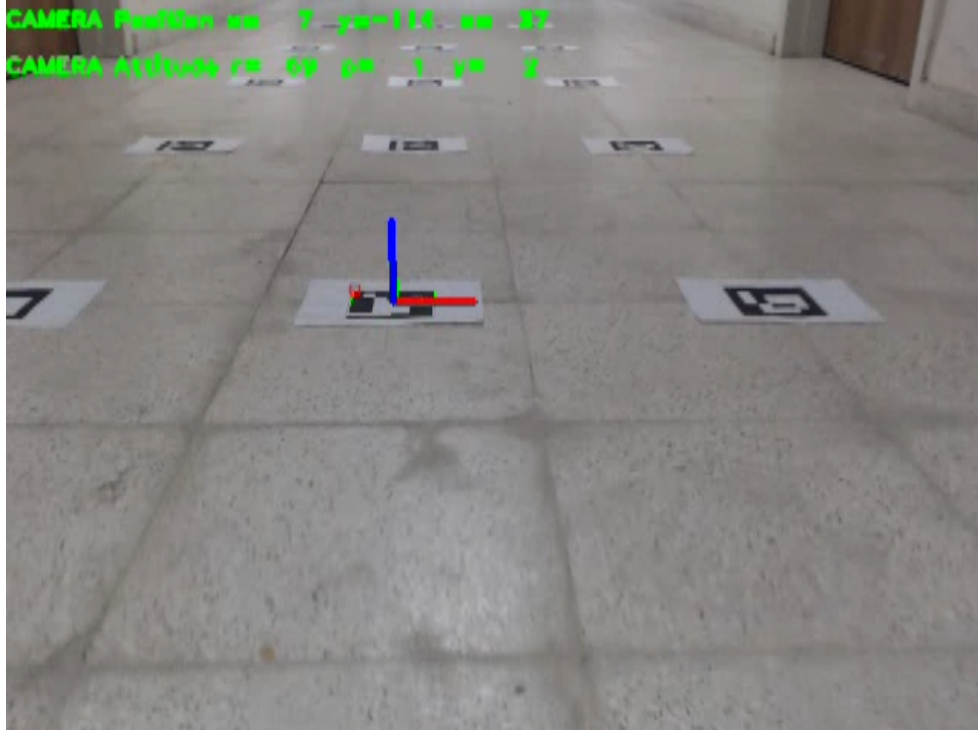
4.2.1. Kapalı Alan Veri Seti

Kapalı alan veri setini, Hacettepe Üniversitesi Elektrik ve Elektronik Mühendisliği Bölümü içerisinde SSV1'i kullanarak topladık. Deneyi bir koridorda yatay yönde 45 cm aralıklarla 3 sıra halinde, dikey yönde 90 cm aralıklarla 10 m boyunca sıraladığımız ArUco işaretleyicileri arasında uzaktan kontrol ettiğimiz bacaklı robotu kullanarak aldık. Deney ortamının Web Kamerası görüntüsü Şekil (4.3)'de görülebilir.

Deneyleri iki farklı günde aldık ve bunları gün ve alınma sırasına göre isimlendirdik. Ayın 7'sinde alınan deneyler uzaktan kumanda ile alındığı için hız komutlarında gürültüye sahiplerdir. Ayın 11'inde aldığımız deneyleri ise robota sabit bir hız komutu göndererek aldık, bu nedenle yapılan hareket daha düzgündür.

Toplanılan veriler;

- Stereo Kamera (ZED)
 - Sağ ve Sol RGBD Görüntü



Şekil 4.3 Kapalı Alan Deney Ortamı

- Sağ ve Sol Gri Ölçekli Görüntü
- Derinlik Görüntüsü
- Görsel Odometri
- LiDAR (VLP-16)
 - Nokta Bulutu
 - LiDAR Odometrisi
- Web Kamerası ve ArUco İşaretleyiciler
 - Referans Pozisyon
 - Deney Videosu

olarak sınıflandırılabilir.

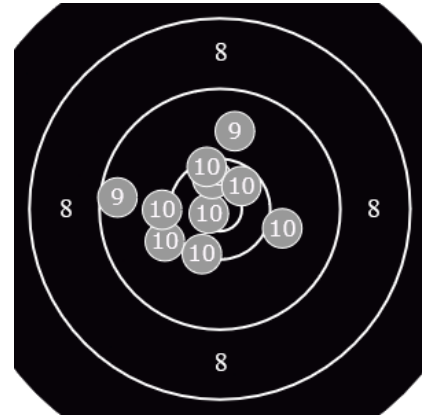
Verileri, ROS Melodic ortamında, Stereolabs'ın ZED kamera için yayınladığı ROS paketi[47], Velodyne'in VLP-16 LiDAR için yayınladığı ROS paketi[48], LeGO-LOAM[26]

makalesi ile birlikte yayınlanan ROS paketi[49] ve kendi yazdığımız ArUco ROS paketini[50] kullanılarak elde ettik. Bu verileri yine ROS ortamında .bag formatında kaydettik.

5. LOAM KOVARYANS KESTİRİMİ

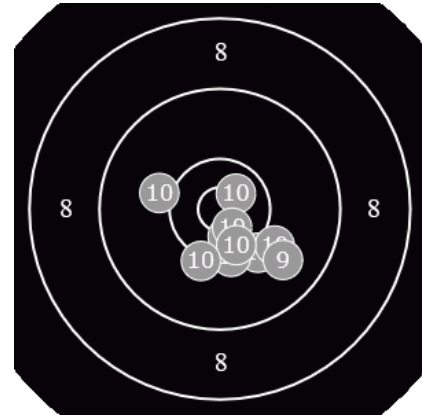
Varyans ve yanlılık (bias) bir kestirimin doğruluğunu ölçümlemek için kullanılan iki niteliktir. Varyans, yapılan her bir kestirimin ortalama etrafında ne kadar dağıldığını gösterir. Yanlılık ise ortalama bir kestirimin gerçek değerden sistematik olarak ne kadar saptığını gösterir. Şekil (5.1)'de bu iki özelliğin farkını görebiliriz. Şekil (5.1.a)'da yapılan atışlar merkez etrafında geniş bir dağılım gösterirken, Şekil (5.1.b)'de merkezin daha altında bir nokta etrafında birbirine çok daha yakın atışlar görülebilir. Bunlardan ilki yüksek varyans düşük yanlılığa, ikincisi ise düşük varyans yüksek yanlılığa örnektir. Tüm atışlar yapıldıktan sonra aldıkları puanlar ise serilerinin doğruluğu olarak değerlendirilebilir.

No	Skor	No	Skor
11	9	16	10
12	10	17	10
13	10	18	10
14	10	19	10
15	10	20	9



(a) D. Mikec Ön eleme S2 sonuçları. Toplam 98 puan.

No	Skor	No	Skor
1	10	6	10
2	10	7	9
3	10	8	10
4	10	9	10
5	10	10	10



(b) Y. Dikeç Ön Eleme S1 sonuçları. Toplam 99 puan.

Şekil 5.1 2024 Paris Olimpiyatları 10 metre Havalı Tabanca kategorisi ön eleme müsabakasında alınan iki sonuç.[51]

Doğruluk, yanlılık ve varyans arasındaki matematiksel ilişki ise Denklem (29)'de verilmiştir. Buradan da görülebileceği üzere, hataların karesinin ortalaması (MSE - Mean Squared Error), kestirimin varyansı ile koreledir ve kestirimin yanlılığı azaldıkça bu korelasyon artmaktadır. Ancak buradaki sorun, sistemin gerçek değeri, gerçek zamanlı olarak bilinemediği için hatayı tam olarak bilmek gerçek zamanlı olarak mümkün değildir. Öte yandan sistemde gerçek zamanlı olarak ölçebileceğimiz çeşitli hata metrikleri vardır ve bu metriklerin sonuçlarının, hataların karesinin ortalaması ile korelasyona sahip olması doğaldır. Bu nedenle çalışma süresince varyans kestirimi için iki farklı metriğin başarımı incelenmiştir.

$$\begin{aligned}\mathbf{MSE} &= \mathbb{E}[\|y - \hat{y}\|^2] \\ &= \mathbb{E}[\|(y - \mathbb{E}[\hat{y}]) + (\mathbb{E}[\hat{y}] - \hat{y})\|^2]\end{aligned}$$

Varsayalım ki, $\mathbf{A} = y - \mathbb{E}[\hat{y}]$, $\mathbf{B} = \mathbb{E}[\hat{y}] - \hat{y}$

$$\begin{aligned}\mathbf{MSE} &= \mathbb{E}[\|\mathbf{A} + \mathbf{B}\|^2] \\ &= \mathbb{E}[\|\mathbf{A}\|^2 + 2\mathbf{A}^T\mathbf{B} + \|\mathbf{B}\|^2] \\ &= \mathbb{E}[\|\mathbf{A}\|^2] + 2\mathbb{E}[\mathbf{A}^T\mathbf{B}] + \mathbb{E}[\|\mathbf{B}\|^2]\end{aligned}$$

$$\begin{aligned}\mathbb{E}[\mathbf{A}^T\mathbf{B}] &= \mathbb{E}[(y - \mathbb{E}[\hat{y}])^T(\mathbb{E}[\hat{y}] - \hat{y})] \\ &= \mathbb{E}[y^T\mathbb{E}[\hat{y}] - y^T\hat{y} - \mathbb{E}[\hat{y}]^T\mathbb{E}[\hat{y}] + \mathbb{E}[\hat{y}]^T\hat{y}] \\ &= \underbrace{\mathbb{E}[y^T\mathbb{E}[\hat{y}] - \mathbb{E}[y^T\mathbb{E}[\hat{y}]]}_{=0} - \underbrace{\mathbb{E}[\mathbb{E}[\hat{y}]^T\mathbb{E}[\hat{y}] - \mathbb{E}[\mathbb{E}[\hat{y}]^T\mathbb{E}[\hat{y}]]}_{=0} \\ &= 0\end{aligned}$$

$$\begin{aligned}\text{O zaman, } \mathbf{MSE} &= \mathbb{E}[\|\mathbf{A}\|^2] + \mathbb{E}[\|\mathbf{B}\|^2] \\ &= \mathbb{E}[\|y - \mathbb{E}[\hat{y}]\|^2] + \mathbb{E}[\|\mathbb{E}[\hat{y}] - \hat{y}\|^2] \\ \mathbf{MSE} &= \|\text{Yanlılık}(\hat{y})\|^2 + \text{Var}(\hat{y})\end{aligned}$$

(29)

5.1. Kullanılan Hata Metrikleri

5.1.1. Ortalama Maliyet

LiDAR odometrisi adımıyla yapılan kestirimin başarısı, bu adımda optimizasyonu yapılan maliyet fonksiyonunun en son değeri ile ilişkilidir. Bu nedenle buradaki hatayı varyans kestirimi yapmak için kullanabiliriz. Ancak ortamda bulunan karşılıklılık sayısı arttıkça toplam maliyet artar, bu nedenle toplam maliyetin normalize edilmesi gerekir. Sonuç olarak ortalama maliyet metriğinin sonucu Denklem (30)'deki gibi elde edilir. Bu yöntemin dezavantajı ise, yalnız LiDAR odometrisi için bir sonuç verse de, haritalama adımıyla yapılan iyileştirmeyi varyans kestirimi problemine doğrudan dahil edememektedir.

$$d_{\mathcal{O}} = \sqrt{\frac{\sum^N (d_{\mathcal{E}}^2 + d_{\mathcal{H}}^2)}{N}} \quad (30)$$

5.1.2. Hausdorff Mesafesi

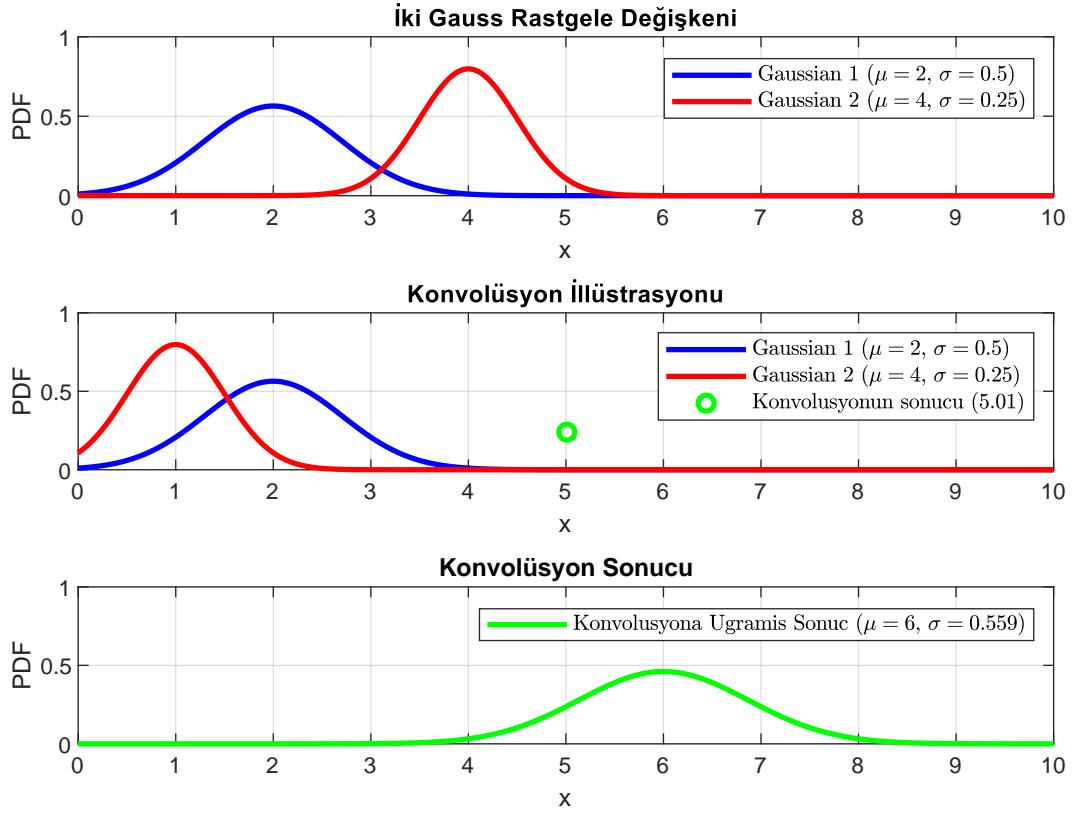
Hausdorff mesafesi, iki nokta bulutu arasındaki maksimum mesafeyi ölçer. Kısaca, önce her bir nokta için, diğer nokta kümesinde bu noktaya en yakın noktaya olan mesafeyi ölçer ve bu değerlerden en büyüğünün sonucunu alır. Denklem (31) şeklinde hesaplanır. $d(x, y)$, problem tanımına göre Euclid ya da Manhattan gibi herhangi bir mesafe fonksiyonu olabilir. Eğer iki nokta bulutu birbirinin aynısı ve mükemmel bir şekilde hizalanmışsa, tüm noktalar için en yakın nokta mesafesi 0 olduğu için, Hausdorff mesafesi 0 değerini verir.

$$d_H(\mathcal{X}, \mathcal{Y}) = \max \left\{ \sup_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}} d(x, y), \sup_{y \in \mathcal{Y}} \inf_{x \in \mathcal{X}} d(y, x) \right\} \quad (31)$$

5.2. Dönüşüm Varyansından Global Varyans Hesaplama

LOAM, Kısım 3.1.1.'de açıklandığı ve Denklem (16)'de de görülebileceği üzere, pozisyon kestirimini daha önceki kestirimleri referans alarak yapar ancak LiDAR odometrisi kısmı için hesaplanan dönüşüm, daha önceki dönüşüm kestirimlerinden tamamen bağımsızdır. Bu duruma olasılık perspektifinden bakarsak Denklem (32)'i elde ederiz. Denklem (32)'nin tek bir aşamasının 1 boyutta görselleştirilmiş hali Şekil (5.2)'de görülebilir.

$$p(\hat{T}_k^0) = *|_{n=1}^k p(\hat{T}_n^{n-1}) \quad (32)$$



Şekil 5.2 LOAM Hareket Güncellemesi. Gaussian 2'nin (kırmızı), \hat{T}_n^{n-1} dönüşüm kestirimi olduğunu varsayalım. bu kestirim, önceki kestirimlerden bağımsızdır. Konvolüsyon illüstrasyonunda, bu dönüşümün, sıfırdan farklı bir varyansa sahip pozisyona (mavi) eklenmesi sürecini canlandırabiliriz. Konvolüsyona Uğramış Sonuç (yeşil) ise kestirim eklendikten sonra elde edilen yeni pozisyon ve varyansını göstermektedir.

Eğer herhangi bir dönme hareketi yapmadan, sadece 3 Boyutlu Euclid uzayında çalışıyor olsaydık ($\widehat{T}_k^{k-1} = \begin{bmatrix} I_{3 \times 3} & \widehat{t}_k^{k-1} \\ 0_{1 \times 3} & 1 \end{bmatrix}$), güncel pozisyon varyansını Denklem (33)'deki gibi elde edebilirdik. Olasılık dağılımının Gauss dağılımı ($p(\widehat{T}_k^{k-1}) = \mathcal{N}(t_k, \sigma_k^2)$) olduğunu varsayarsak, pozisyon ve varyansı, Denklem (34) ve (35)'deki gibi hesaplayabiliriz.

$$p(\widehat{T}_k^0) = p(\widehat{T}_{k-1}^0) * p(\widehat{T}_k^{k-1}) \quad (33)$$

$$t_k^0 = t_{k-1}^0 + t_k \quad (34)$$

$$\sigma_k^{0^2} = \sigma_{k-1}^{0^2} + \sigma_k^2 \quad (35)$$

Eğer dönme hareketi mevcutsa ($\widehat{T}_k^{k-1} = \begin{bmatrix} R_k^{k-1} & \widehat{t}_k^{k-1} \\ 0_{1 \times 3} & 1 \end{bmatrix}$), dönme hareketinin doğrusal bir operasyon olmaması nedeni ile bu hesap çok daha karmaşık bir hal alacaktır. Ancak özellikle LiDAR sensör merkezine uzak noktalarda maliyet fonksiyonu dönme hareketindeki değişimlere karşı daha hassastır. Bu nedenle, özellikle düşük hızlarda (yer değiştirmenin düşük olduğu durumlarda), dönme kestiriminin daha hassas sonuç verdiğini söyleyebiliriz. Bunun sonucunda LiDAR odometrisi için deterministik bir dönüş kestirimi yaptığımızı varsayarsak, Denklem (34) ve (35), Denklem (36), (37) ve (38) formunu alır.

$$\widehat{t}_k^0 = \widehat{t}_{k-1}^0 + \widehat{R}_{k-1}^0 \widehat{t}_k \quad (36)$$

$$\widehat{R}_k^0 = \widehat{R}_{k-1}^0 \widehat{R}_k^{k-1} \quad (37)$$

$$\sigma_k^{0^2} = \sigma_{k-1}^{0^2} + \widehat{R}_{k-1}^0 \sigma_k^2 \widehat{R}_{k-1}^{0^T} \quad (38)$$

6. 1 BOYUTTA SENSÖR FÜZYONU

Çalışmanın bu kısmı, Sensör Sistemi Versiyon 1 ile toplanan veri seti üzerinde ileri yönde tek eksenle yapılan hareket baz alınarak yapılmıştır. LiDAR sensörü varyans kestirimi için Kısım (5.1.1.)’de anlatılan ortalama maliyet metodunu kullandık.

Referans pozisyon için kullanılan Aruco işaretleyici verisi, sensörler ile senkronize olmadığı için, sensör verilerinin, referans ölçümü alınan zamanlardaki değerlerini interpolasyon yaparak hesapladık. Yine bu zamanlardaki varyans değerlerini ise Denklem (38)’de birim rotasyon kullanarak elde ettik.

6.1. Kullanılan Metotlar

Sensör değerleri çeşitli çevrimiçi ve çevrimdışı sensör füzyonu metotları kullanılarak karşılaştırılmıştır. Bunlar;

6.1.1. Çevrimdışı Metotlar

6.1.1.1. Sabit α metodunda, sensörlerin füzyondaki ağırlığı sabittir. Bu ağırlık, tüm veri için en küçük kareler yöntemi ile optimum sonucu veren katsayıdır. Sonuç Denklem (39)’deki gibi hesaplanır.

$$X_{Sabit} = \alpha X_{LiDAR} + (1 - \alpha) X_{Stereo} \quad (39)$$

Sabit α metodu çevrim dışı bir metottur ve önerilen çevrimiçi metotların başarılarını kıyaslamak amacı ile eklenmiştir.

6.1.1.2. Dinamik α metodunda, her yeni pozisyon verisi için, sensörlerin ağırlıkları Denklem (40)’deki gibi tekrar hesaplanmaktadır. Eğer ilk hesaplanan değer (0, 1) aralığı

dışındaysa değeri 0 ya da 1 kabul edilir. Bu durum iki sensör de pozisyonun altında ya da üstünde sonuç verdiğinde ortaya çıkmaktadır. Efektif olarak sonuca daha yakın olan sensörün değeri kabul edilir. Füzyon sonucu Denklem (41)'deki gibi hesaplanır.

$$f(k) = \frac{X_{ArUco}(k) - X_{Stereo}(k)}{X_{LiDAR}(k) - X_{Stereo}(k)}$$

$$\alpha(k) = \begin{cases} 0, & \text{if } f(k) < 0, \\ 1, & \text{if } f(k) > 1, \\ f(k), & \text{if } 0 \leq f(k) \leq 1. \end{cases} \quad (40)$$

$$X_{Dinamik}(k) = \alpha(k)X_{LiDAR}(k) + (1 - \alpha(k))X_{Stereo}(k) \quad (41)$$

Dinamik α metodu çevrim dışı bir metottur ve sensör sonuçları ile yapılabilmesi mümkün en doğru mutlak pozisyon sonucunu vermektedir. Çünkü, eğer referans pozisyon iki sensör kestiriminin ortasında ise bu değeri, eğer dışındaysa referans pozisyona en yakın sensörün pozisyon değerini vermektedir. Bu nedenle önerilen çevrimiçi metotların başarılarını kıyaslarken bir alt limiti göstermesi amacı ile eklenmiştir.

6.1.2. Çevrimiçi Metotlar

6.1.2.1. Global ortalama metodunda, sensör pozisyonunun ağırlıklı ortalaması alınır. Sensörün ağırlığı, diğer sensörün global varyans değerinin kökü (σ) ile orantılı olacak şekilde belirlenir. Eğer bir sensörün varyansı ne kadar büyükse, yani güvenilirliği düşükse, diğer sensöre o kadar çok güvenilir. Sonuç Denklem (42)'deki gibi hesaplanır.

$$\alpha(k) = \frac{\sigma_{X_{Stereo}}(k)}{\sigma_{X_{Stereo}}(k) + \sigma_{X_{LiDAR}}(k)} \quad (42)$$

$$X_{Global}(k) = \alpha(k)X_{LiDAR}(k) + (1 - \alpha(k))X_{Stereo}(k)$$

6.1.2.2. Dönüşüm ortalama metodunda, sensör dönüşümlerinin ağırlıklı ortalaması alınır. Sensörün ağırlığı, diğer sensörün dönüşüm varyans değerinin kökü σ ile orantılı olacak şekilde belirlenir. Eğer bir sensörün dönüşüm varyansı ne kadar büyükse, yani güvenilirliği düşükse, diğer sensöre o kadar çok güvenilir. Sonuç Denklem (43)'deki gibi hesaplanır.

$$\alpha(k) = \frac{\sigma_{T_{Stereo}}(k)}{\sigma_{T_{Stereo}}(k) + \sigma_{T_{LiDAR}}(k)} \quad (43)$$

$$X_{Dönüşüm}(k) = X_{Dönüşüm}(k-1) + \alpha(k)T_{LiDAR}(k) + (1 - \alpha(k))T_{Stereo}(k)$$

6.2. Sonuçlar

Metotlardan elde edilen sonuçları mutlak ve görece hata metrikleri üzerinden karşılaştırdık.

Mutlak hata metriği adından da anlaşılacağı üzere, yapılan kestirimin referans pozisyona olan mutlak uzaklığıdır. Bu metrik çok temel bir karşılaştırma metriği olmakla beraber, deney süresi arttıkça, mutlak pozisyon bilgisi veren bir sensörün olmadığı durumda, bu metriğin ölçtüğü hata miktarı artmaktadır. Bu nedenle farklı senaryolara sahip deneyler, hatta aynı deneyin farklı kısımları arasında karşılaştırma yapmaya elverişli değildir.

Görece hata metriği ise, referans verideki belirli bir değer üzerinde yapılan her yer değiştirme için yapılan kestirimdeki hatayı ölçer. Bu nedenle zamandan bağımsız bir sonuç verebilir ya da farklı senaryoların kestirime etkisini karşılaştırabilir. Denklem (44) ile hesaplanır. Biz çalışmanın burasında referans pozisyonda yapılan her 15 santimetrelilik hareket için yapılan kestirimdeki hata yüzdesini ölçtük. Eğer son hesaplamadan beri yapılan yer değiştirme 25 santimetreden düşük ise, o pozisyon için hata hesabı yapılmaz.

$$\text{Görece Hata}(k) = |\Delta\text{Referans Pozisyon}(k) - \Delta\text{Pozisyon Kestirimi}(k)| \quad (44)$$

Yöntemlerin 16 deney sonucundaki ortalama performansı Tablo (6.2) ve (6.1)'de görülebilir. Ayrıca bu 16 deneyin burada listelemediğimiz sonuçları Ekler'in A Kısımında bulunabilir.

	Ort.(m)	Medyan(m)	Std.(m²)	RMSE(m)
Sabit	0.23	0.16	0.04	0.30
Dinamik	0.22	0.15	0.05	0.31
Global	0.25	0.20	0.04	0.32
Dönüşüm	0.25	0.19	0.04	0.33

Tablo 6.1 Kapalı Alan Veriseti için Ortalama Görece Hata Tablosu

	Ort.(m)	Medyan(m)	Std.(m²)	RMSE(m)
Sabit	0.21	0.15	0.04	0.28
Dinamik	0.20	0.14	0.04	0.28
Global	0.33	0.29	0.05	0.39
Dönüşüm	0.33	0.29	0.05	0.40

Tablo 6.2 Kapalı Alan Veriseti için Ortalama Mutlak Hata Tablosu

Sonuçları incelediğimizde mutlak hata için hem Dönüşüm hem de Global ortalama metodu Sabit ve Dinamik α metodlarının gerisinde kalmıştır. Özellikle bu hata metriğinde başarısının daha yüksek olmasını beklediğimiz Global ortalama metodunun Sabit α metodundan daha kötü performans göstermesi kullandığımız ortalama maliyet üzerinden varyans hesabının ya da bu değer üzerinden yaptığımız global varyans hesabının problemi çözmekte başarısız olduğu anlamına gelmektedir.

Görece hata metriği için ise, Dönüşüm ortalama metodu ortalamada Sabit α metodunu geride bırakmış hatta kimi zaman deney bazında Dinamik α metodundan bile daha başarılı sonuç göstermiştir. Deney 11_07 bunlardan biridir ve bu sonuçlar Tablo (6.3)'de görülebilir. Bu metrikte daha başarısız olmasını beklediğimiz Global Ortalama metodu ise beklenenin aksine Dönüşüm Ortalamayı geride bırakmıştır. Ancak Dönüşüm Ortalamasının Dinamik α metodundan daha başarılı sonuç verdiği deneylerde, Global ortalama Dönüşüm ortalamadan ya daha kötü sonuç vermiş ya da çok küçük farklarla daha başarılı olmuştur. Öte yandan

Dönüşüm ortalamasının çok fazla sapma gösterdiği deneylerde de hatasının görece düşük kalması nedeniyle ortalamada daha başarılı olmuştur. Bu durumun sonucu hatanın standart sapmasından da görülebilir.

	Ort.(m)	Medyan(m)	Std.(m²)	RMSE(m)
Sabit	0.1009	0.0834	0.0045	0.1204
Dinamik	0.0963	0.0852	0.0033	0.1117
Global	0.1074	0.1297	0.0034	0.1218
Dönüşüm	0.0723	0.0647	0.0017	0.0826

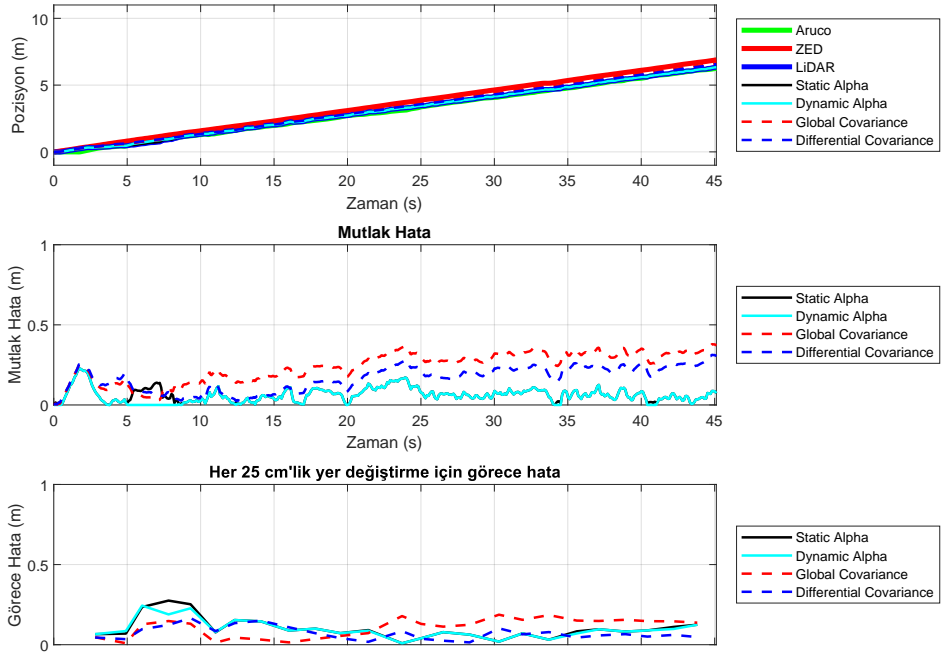
Tablo 6.3 Kapalı Alan Veriseti deney 11_07 için Görece Hata Tablosu. Tabloda da görüleceği üzere Dönüşüm ortalama metodu bu deney için en düşük ortalama ve RMSE hataya sahiptir.

	Ort.(m)	Medyan(m)	Std.(m²)	RMSE(m)
Sabit	0.0657	0.0606	0.0018	0.0784
Dinamik	0.0590	0.0559	0.0021	0.0745
Global	0.2382	0.2659	0.0086	0.2555
Dönüşüm	0.1643	0.1788	0.0058	0.1809

Tablo 6.4 Kapalı Alan Veriseti deney 11_07 için Mutlak Hata Tablosu

Ayrıca görece hata, tüm metotlar için başlangıçta yüksekken sonrasında düşme eğilimi göstermiştir. Bu durum Şekil (6.1) görece hata görselinde görülebilir. Bunun temel nedeni LiDAR odometrisinde IMU kullanılmadığı için kayma giderme (deskewing) işleminin bir önceki dönüşüm ile elde edilen hız bilgisi ile yapılmasıdır. Robot sabit bir hızla hareket ederken yapılan kayma giderme işlemi, başlangıçtaki ivmeli hareket sırasında yapılan kayma giderme işleminden daha başarılıdır. Bu nedenle bu süreçte yapılan dönüşüm kestirimi de daha başarılı olmaktadır.

Bununla birlikte, ortalama maliyet üzerinden yapılan varyans kestiriminin sonuçları genel olarak baktığımızda yeterince tatmin edici değildir. Bu durumun olası bir nedeni, aşırı uyum (overfit) ya da yerel minimuma takılma gibi problemler doğal olarak yapılan kestirimin varyansını artırırken, optimizasyon probleminde minimize edilmeye çalışılan bu değer bu problemler sonucunda düşmesi olabilir. Bu yöntemin bir diğer sıkıntısı ise, şu anki hali ile sadece LiDAR Odometrisi bölümü için bir sonuç vermekte, Haritalama ile yapılan düzeltmeleri hesabına dahil etmemektedir. Bu nedenle ilerleyen kısımlarda Hausdorff mesafesi üzerinden yapılan varyans kestirimi ile devam etmeye karar verdik.



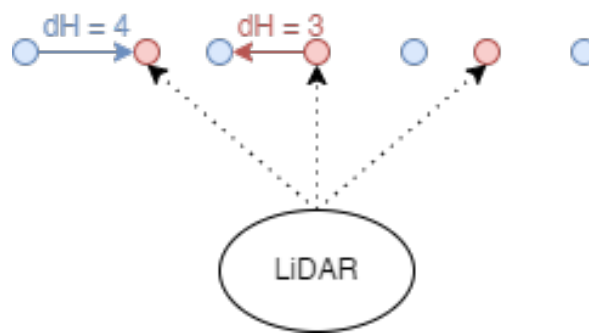
Şekil 6.1 Kapalı Alan Veriseti Deney 11_07 Kestirim Sonucu

7. 3 BOYUTTA SENSÖR FÜZYONU

Çalışmanın bu kısmında, hata metriği olarak Hausdorff Mesafesi'ni kullandık. Sensör füzyonu yöntemleri olarak ise, bir önceki kısımda denediğimiz yöntemlerin 3 boyuttaki karşılıkları denebilecek yöntemlerin performansını inceledik. Veriseti olarak ise MulRan[33] veri setini kullandık. Veri setinde yapılan tek değişiklik, 5 Hz olan GPS frekansının 1 Hz olarak alt örnekleminin alınmasıdır. Bu uygulamanın nedeni Kalman filtresinin performansını düşürerek LiDAR Odometrisinin etkisini daha net gözlemleyebilmektir.

7.1. Tek Yönlü Hausdorff Mesafesi

Orjinal Hausdorff Mesafesi metriği iki nokta bulutunda bulunan her nokta için hesaplanır. Ancak bizim problemimizde, ölçümden gelen nokta bulutu seyrekken (sparse), haritadan gelen nokta bulutu yoğundur (dense). Bu durum sonucunda, eğer harita nokta bulutundan ölçüm nokta bulutuna olan uzaklık hesaplandığı durumda, kenar yada düzlemin dış kısımlarında kalan noktalar için ölçümde tek bir karşılık bulunabilecek ve bu durum, kestirim başarıs yerine, örnekleme sıklığından kaynaklı yüksek değerler elde edilmesine neden olacaktır. Bu durum Şekil (7.1)'de gösterilmiştir.



Şekil 7.1 Aynı doğru üzerinde bulunan Harita (mavi) ve Ölçüm (Kırmızı) kenar nokta bulutları üzerinden Hausdorff Mesafesinin hesaplanması. Eğer Harita nokta bulutundan Ölçüm nokta bulutuna doğru olan uzaklık hesaba katılırsa, bu durum sonucu kestirim başarımından çok örnekleme sıklığına bağımlı hale getirecektir.

Ayrıca, hem problem uzayını küçültüp işlemi hızlandırmak, hem de daha uygun karşılıklılık bulmak adına, yüzey ve kenar nokta bulutları için Hausdorff Mesafesi ayrı ayrı olarak

hesaplayıp ve daha büyük değeri Denklem (47)'deki gibi sonuç olarak kabul ettik. Genel olarak yüzey nokta bulutları, kenar nokta bulutlarından daha büyüktür. Eğer yüzey ölçüm nokta bulutu boyutu n , referans nokta bulutu boyutu m ise, Hausdorff Mesafesi için hesaplamada zaman karmaşıklığımız (time complexity) $O(n \log m)$ olacaktır.

Hausdorff Mesafesini ölçmek için önce güncel yüzey ve kenar nokta bulutları \mathcal{H}_k^L ve \mathcal{E}_k^L 'nin orijinini, güncel pozisyon kestirimi \hat{T}_k^0 'ye taşıdık. Burada güncel nokta bulutları \mathcal{H}_k^L ve \mathcal{E}_k^L 'den güncel harita nokta bulutları \mathcal{H}_m^{map} ve \mathcal{E}_m^{map} 'ye göre tek yönlü Hausdorff mesafesi Denklem (45) ve (46)'deki gibi hesapladık.

$$d_{\hat{\mathcal{H}}_k^{map}}(\hat{\mathcal{H}}_k^{map}, \mathcal{H}_m^{map}) = \sup_{l \in \hat{\mathcal{H}}_k^{map}} \left\{ \inf_{m \in \mathcal{H}_m^{map}} \|l - m\| \right\} \quad (45)$$

$$d_{\hat{\mathcal{E}}_k^{map}}(\hat{\mathcal{E}}_k^{map}, \mathcal{E}_m^{map}) = \sup_{l \in \hat{\mathcal{E}}_k^{map}} \left\{ \inf_{m \in \mathcal{E}_m^{map}} \|l - m\| \right\} \quad (46)$$

$$\text{Var}(\hat{T}_k^{map}) = I_{3 \times 3} \times \max(d_{\hat{\mathcal{H}}_k^{map}}, d_{\hat{\mathcal{E}}_k^{map}}) \quad (47)$$

Bu metriğin karşılaştığı problem, aykırı verilerden çok fazla etkilenmesidir. Bu aykırı veri, yanlış bir ölçüm olabileceği gibi, üzerinden ilk defa ölçüm alınan bir cisim olabilir. Referans nokta bulutunda bu cisme dair bir ölçüm bulunmadığı için, bu noktanın en yakın uzaklığı, normalden çok daha fazla olacaktır. Bu nedenle Hausdorff Mesafesi'ni hesaplanan noktanın 5 metre uzağında, referans nokta bulutunda en az 4 nokta bulunmuyorsa, bu sonuç hesap dışı bırakılmıştır. Bu değerlerin seçilmesinin nedeni, çalışmada referans algoritma olarak kullanılan LeGO-LOAM[26] algoritmasının karşılıklılık bulurken yine bu değerleri kullanmasıdır. Bu nedenle bu değerler sisteme efektif olarak yeni parametre eklememektedir.

7.2. Kullanılan Metotlar

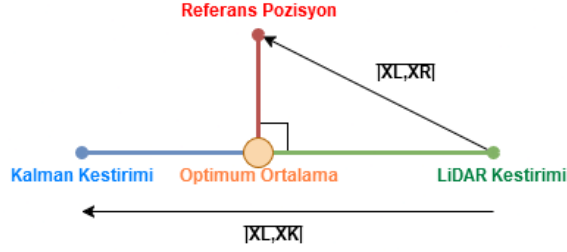
Kısım (2.)'de de anlatıldığı üzere, metriğimizin yeterince düşük yanlışlık altında, MSE ve dolayısıyla varyans değeri ile ilintili olmasını bekliyoruz. Tüm bu varsayımları zaten yaptığımız için, hipotezimizi sağlayan bir metrik bulduğumuz durumda en düşük

MSE'yi sağlayacak etkin kestirici (efficient estimator) MVUE (Minimum Variance Unbiased Estimator - En Küçük Varyanslı Yansız Kestirici) modelidir. Bu nedenle bu kısımda yaptığımız kestirimleri bu modeli kullanarak yaptık. Buradaki tek farkımız, pozisyon değerleri arasındaki çapraz varyans değerlerinin pozisyonların kendi varyans değerlerine göre çok daha küçük olması nedeni ile kestirime katkıları minimal seviyededir. Hesaplamaları kolaylaştırmak adına, burada kovaryans matrislerinin sadece çapraz elemanlarını alıp, diğer değerlerini 0 olarak kullandık.

Veri setindeki IMU ve GPS verisini, INS sensörleri için geliştirilmiş hazır bir Kalman Filtresi[52] kullanarak birleştirdik (T_K). LiDAR odometrisi (T_L) üretmek için ise, LeGO-LOAM[26] algoritmasının, Kısım (5.1.2.)'de anlattığımız Hausdorff Mesafesi yöntemi ile varyans kestirimi yapacak şekilde modifiye ettiğimiz bir versiyonunu[53] kullandık. Yapılan bu değişiklikler Kısım (7.1.)'de detaylı olarak anlatılmıştır. Ayrıca algoritmada opsiyonel olarak bulunan IMU entegrasyonu ve döngü kapama becerilerini devre dışı bıraktık. Daha sonra elde edilen bu iki odometri verisini Denklem (42) ve (43)'de verilen Global Ortalama ve Dönüşüm Ortalama metotlarını kullanarak birleştirdik. Son olarak, bu iki metodun da avantajlarını kullanan Düzeltilmiş Ortalama metodunda, Global Ortalama ve Dönüşüm Ortalama metotlarının sonuçları üzerinden hesapladık. Ayrıca, elde edilen sonuçları karşılaştırabilmek amacıyla (41)'de verilen Dinamik α metoduna benzer Optimal Ortalama metodu ile sensör verilerinin doğrusal kombinasyonu ile elde edilmesi mümkün en başarılı kestirim değerini bulduk.

7.2.1. Optimal Ortalama

Optimal Ortalama metodu çevrimdışı bir metottur ve bir önceki kısımdaki gibi yine elde edilen sonuçları karşılaştırabileceğimiz bir referans olması amacı ile eklenmiştir. Ancak burada 3 Boyutta çalışmamız nedeni ile tam doğru sonucu almamız her zaman mümkün değildir. Bu nedenle bu metot var olan sensör verisinin doğrusal ortalaması ile alınabilecek optimum sonucu vermektedir. Bu durumun nedeni Şekil (7.2)'de anlatılmıştır.



Şekil 7.2 X_{Opt} 'un Hesaplanması: 2 sensörden alınan pozisyon verisinin doğrusal ağırlıklı ortalaması ile elde edilebilecek tüm kestirimler, uzayda bu iki nokta arasında kalan doğru parçasının üzerinde bulunmaktadır. Bu noktalar arasındaki en başarılı sonuca X_{Opt} dersek, bu noktadan referans pozisyona çizilecek doğru, kestirim doğrusuna dik olmak zorundadır[10].

LiDAR, Kalman, Optimal ve Referans kestirimini sırası ile X_{LiDAR} , X_{Kalman} , X_{Opt} ve X_{Ref} olarak tanımlarsak, X_{Opt} değerini Denklem (48) ve (49)'deki gibi elde edebiliriz.

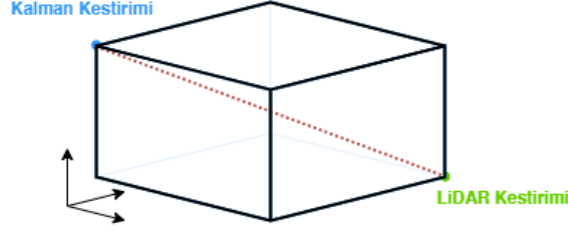
$$f(k) = \frac{\overrightarrow{X_{LiDAR}(k)X_{Ref}(k)} \cdot \overrightarrow{X_{LiDAR}(k)X_{Kalman}(k)}}{|\overrightarrow{X_{LiDAR}(k)X_{Kalman}(k)}|^2}$$

$$\alpha(k) = \begin{cases} 0, & \text{if } f(k) < 0, \\ 1, & \text{if } f(k) > 1, \\ f(k), & \text{if } 0 \leq f(k) \leq 1. \end{cases} \quad (48)$$

$$X_{Opt} = X_{LiDAR}(k) + \alpha(k) \cdot \overrightarrow{X_{LiDAR}(k)X_{Kalman}(k)} \quad (49)$$

7.2.2. Global Ortalama

Global Ortalama metodunda LiDAR odometrisinin pozisyon varyansını Denklem (32)'e benzer şekilde elde ettik. Yapılan hesaplamayı basitleştirmek amacı ile, Kalman Filtresi kovaryans matrisinin çapraz elemanlarını, karşılık geldikleri yöndeki LiDAR odometrisi ağırlığı, LiDAR odometrisi kovaryans matrisinin çapraz elemanlarını, karşılık geldikleri yöndeki Kalman Filtresi ağırlığı olarak belirledik. Burada anlatılan işlem Denklem (50)



Şekil 7.3 Global Ortalama metodu sonuç uzayı Siyah prizma ile, Optimal Ortalama metodunun sonuç doğrusu ise Kırmızı çizgi ile gösterilmiştir.

ve (51)'de görülebilir. Yapılan bu işlem, kestirim sonucunu, ayrıtları birim vektör olan ve iki kestirim sonucunun cisim köşegenini belirttiği dikdörtgenler prizmasının içinde kalan uzay ile sınırlar. Bu uzay Şekil (7.3)'de görülebilir. Şekildeki cisim köşegeni, Optimal Ortalama metodunda bahsedilen kestirim doğrusuna denk gelmektedir.

$$A(k) = (diag(P_{LiDAR}(k)) + diag(P_{Kalman}(k)))^{-1} \times diag(P_{LiDAR}(k)) \quad (50)$$

$$X_{Global}(k) = A(k) \times X_{Kalman}(k) + (1 - A(k)) \times X_{LiDAR}(k) \quad (51)$$

7.2.3. Dönüşüm Ortalama

Dönüşüm Ortalama metodunda LiDAR odometrisinin dönüşüm varyansı için o pozisyonda hesaplanan Hausdorff Mesafesini kullandık. Kullandığımız hazır Kalman Filtresi iki tahmin arasındaki dönüşümün varyansını vermemektedir, bu nedenle bu değerin ayrıca hesaplanması gerekir.

Bu değeri elde etmenin bir yolu her tahminden sonra filtreyi tekrar ilklendirmek (initialization) olabilir. Ancak bu yöntem, Kalman Filtresi için yapılacak her işlemi tekrardan yapmayı gerektirir ve tahmin kovaryansının hesaplandığı adımın hesaplamada zaman karmaşıklığı, n boyutlu durum vektörü için $O(n^3)$ olduğu düşünüldüğünde hızlı bir süreç değildir. Ayrıca, bu işlem her tahmin sonrası Kalman Filtresinin ve ölçümlerin tüm durum ve kovaryans değerlerinin bilinmesini gerektirir ve Kalman Filtresi ile sonuç üreten INS sensörlerinin kullanıldığı durumda, bu değerlerin hepsi her zaman mevcut değildir. Bu

nedenle, Kalman Filtresinin dönüşüm varyansını elde etmek için yeni bir metot önerdik. Bu metotta yaptığımız işlem kısaca, Kalman Filtresinin dönüşüm varyansı için bir alt limit ve bir üst limit belirleyip bunların ortalamasını almaktır.

7.2.3.1. Kalman Filtresi dönüşüm varyansı alt limiti Kalman Filtresi'nin dönüşüm varyansı için alt limit belirlerken Denklem (32)'deki mantığı ters konvolüsyon kullanarak tekrarladık. Bunun için $k - 1$ anındaki Kalman Filtresi tahmini P_{k-1}^0 'i P_k^0 'ye taşıyan dönüşüm $T_K(k)$ olsun ($P_{k-1}^0 * T_K(k) = P_k^0$). Bu durumda $T_K(k)$ ters konvolüsyonla Denklem (52) gibi hesaplanabilir.

$$T_K(k) = P_{k-1}^0{}^{-1} * P_k^0 + G_k \quad (52)$$

Burada G_k 0 ortalama değere sahip bir beyaz gürültüdür. Kovaryans matrisleri kesin pozitif (positive definite) oldukları için bu gürültü varyansını 0 kabul ettiğimizde elde edeceğimiz sonuç, dönüşüm varyansı için bir alt limit olacaktır. Ancak bu çıkarma işlemi sonucunda elde edilen matrisin yarı-kesin pozitif (positive semi-definite) olduğundan emin olmamız gerekir. Bunun için Özyayrışım (Eigendecomposition) ile elde edeceğimiz negatif özdeğerleri (eigenvalue) 0'a eşitleyebiliriz. Böylece varyans için alt limiti Denklem (53)'teki gibi elde edebiliriz.

$$\begin{aligned} Q\Lambda Q^T &= P_k^0 - P_{k-1}^0 \\ \Lambda^+ &= \max(\Lambda, 0) \\ P_{(k)}^{k-1} &> R_{k-1}^T Q\Lambda^+ Q^T R_{k-1} = R_{k-1}^T (P_k^0 - P_{k-1}^0)^+ R_{k-1} \end{aligned} \quad (53)$$

7.2.3.2. Kalman Filtresi dönüşüm varyansı üst limiti Kalman Filtresi'nin dönüşüm varyansı üst limitin belirlemek için ise Ayrık Doğrusal Olmayan Kalman Filtresi'nin denklemlerini (Denklem (54)-(60))[54] kullanabiliriz.

Eğer Ölçüm adımında hiç bir ölçüm alınmadığını var sayarsak, ki bu durumu sonsuz varyansa sahip bir ölçüm olarak kabul edebiliriz ($R_{k-1} \approx \text{inf}$), Denklem (55)'deki K değeri 0 olur. Bunun sonucunda ise Denklem (57) sadeleşerek $P_{k-1|k-1} = P_{k-1|k-2}$ formunu alır. Bu sonucu Denklem (59)'e yerleştirdiğimizde ise Denklem (62)'i elde ederiz.

Ölçüm Adımı:

$$H_{k-1} = \left. \frac{\partial \mathbf{h}(\mathbf{x}, t_{k-2})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-2}} \quad (54)$$

$$K_{k-1} = P_{k-1|k-2} H_{k-1}^T \left(H_{k-1} P_{k-1|k-2} H_{k-1}^T + R_{k-1} \right)^{-1} \quad (55)$$

$$\hat{\mathbf{X}}_{k-1|k-1} = \hat{\mathbf{X}}_{k-1|k-2} + K_{k-1} \left[\hat{\mathbf{Z}}_{k-1} - \mathbf{h}(\hat{\mathbf{x}}_{k-1|k-2}, k-1) \right] \quad (56)$$

$$P_{k-1|k-1} = (I - K_{k-1} H_{k-1}) P_{k-1|k-2} \quad (57)$$

Tahmin Adımı:

$$\hat{\mathbf{X}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{X}}_{k-1|k-1}, k-1) \quad (58)$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q \quad (59)$$

$$F_{k-1} = \left. \frac{\partial \mathbf{f}(\mathbf{x}, t)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1|k-1}} \quad (60)$$

$$(61)$$

$$P_{k|k-1} = F_k P_{k-1|k-2} F_k^T + Q \quad (62)$$

3 Boyutlu Euclid uzayında holonomik hareket yapan bir araç için state vektörümüz

$\mathbf{x} = [x_{(1 \times 3)}, \dot{x}_{(1 \times 3)}, \ddot{x}_{(1 \times 3)}]^T$ olur. Eğer bu state vektörünün State transition matrisini pozisyon (F_p), hız (F_v) ve ivme (F_a) için parçalara bölersek ve pozisyon, hız ve ivme durumları arasındaki çapraz varyansları 0 kabul edersek Denklem (63) ve (64)'i elde ederiz. Bu sonuçları Denklem (62)' içine yerleştirirsek pozisyon varyansı için Denklem (65)'i elde

ederiz.

$$F = \begin{bmatrix} F_{p(3 \times 9)} \\ F_{v(3 \times 9)} \\ F_{a(3 \times 9)} \end{bmatrix}$$

$$F_{p(k)} = \left[I_{(3 \times 3)}, dt \times I_{(3 \times 3)}, \frac{dt^2}{2} \times I_{(3 \times 3)} \right], \quad dt = t(k) - t(k-1) \quad (63)$$

$$P = \begin{bmatrix} P_{p(3 \times 3)} & 0_{(3 \times 3)} & 0_{(3 \times 3)} \\ 0_{(3 \times 3)} & P_{v(3 \times 3)} & 0_{(3 \times 3)} \\ 0_{(3 \times 3)} & 0_{(3 \times 3)} & P_{a(3 \times 3)} \end{bmatrix} \quad (64)$$

$$P_{p(k|k-1)} = P_{p(k-1|k-2)} + dt^2 \times P_{v(k-1|k-2)} + \frac{dt^4}{4} \times P_{a(k-1|k-2)} + Q$$

$$P_{p(k|k-1)} - P_{p(k-1|k-2)} = dt^2 \times P_{v(k-1|k-2)} + \frac{dt^4}{4} \times P_{a(k-1|k-2)} + Q \quad (65)$$

Denklem (65)'deki varyans değerleri dünya koordinat sistemini referans almaktadır, bu nedenle bunları gövde koordinat sistemine dönüştürürsek, Denklem (66)' elde ederiz. Bu sonuç, iki tahmin adımı arasında hiçbir ölçüm yapılmadığı takdirde elde edeceğimiz sonuçtur. IMU ölçümleri 100 Hz, LiDAR ölçümleri ise 10 Hz de alındığı için bu aralıkta bir ölçüm alınacaktır. Bu nedenle bu sonucu, dönüşümün varyansı için bir üst limit olarak kullanabiliriz.

$$P_{p(k)}^{k-1} < \widehat{R}_{k-1}^{0T} \left(dt^2 \times P_{v(k-1|k-2)} + \frac{dt^4}{4} \times P_{a(k-1|k-2)} + Q \right) \widehat{R}_{k-1}^0 \quad (66)$$

Alt ve üst limitini bildiğimiz dönüşüm varyansını, bu iki limitin ortalamasını alarak hesaplayabiliriz. Biz çalışmamızda ortalamayı alırken eşit ağırlık kullanmayı tercih ettik. Bu işlem sonucunda ise dönüşüm varyans değeri $\sigma_k^{k-1,2}$, yi, Denklem (38)'den yola çıkarak, Denklem (67)'deki gibi hesaplayabiliriz. Alt limit matrisinin yarı-kesin pozitif olması durumunda kesin pozitif bir matris ile ortalamasının kesin pozitif olacağı 2 boyutlu matris için Ekler'de kanıtlanmıştır.

$$P^{k-1}(k) \approx \frac{R(k-1)^T \left((P_k^0 - P_{k-1}^0)^+ + \left(dt^2 \times P_{v(k-1|k-2)} + \frac{dt^4}{4} \times P_{a(k-1|k-2)} + Q \right) \right) R(k-1)}{2} \quad (67)$$

Önerdiğimiz metodun hesaplamada zaman karmaşıklığı, matris çarpımı nedeniyle hala $O(n^3)$ olmakla birlikte, buradaki n değeri her zaman 3'tür.

İki sensör için de dönüşüm kestirimlerinin varyansı bulduktan sonra, iki kestirimin yaptığı öteleme, Denklem (68), (69) ve (70) kullanılarak hesaplanır. Başlangıçta Kalman Filtresi'nin oryantasyon verisi için de yukarıdakine benzer bir yöntem izlemekle birlikte, Kalman Filtresi'nin oryantasyon varyans değerleri, Hausdorf Mesafesi sonucundan 5 büyüklük mertesi küçük (10^{-5}) çıktığı için, işlemi hızlandırmak adına, Oryantasyon değeri olarak doğrudan Kalman Filtresi'nin sonucunu referans aldık.

$$A(k) = (diag(P_{LiDAR}^{k-1}(k)) + diag(P_{Kalman}^{k-1}(k)))^{-1} \times diag(P_{LiDAR}^{k-1}(k)) \quad (68)$$

$$\Delta X_{Dönüşüm}(k) = \Delta X_{Dönüşüm}(k-1) \times \left[A(k) \times \Delta X_{Kalman}^{k-1}(k) + (1 - A(k)) \times \Delta X_{LiDAR}^{k-1}(k) \right] \quad (69)$$

$$X_{Dönüşüm}(k) = X_{Dönüşüm}(k-1) + R_{Kalman}^{k-1} \Delta X_{Dönüşüm}(k) \quad (70)$$

7.2.4. Düzeltilmiş Ortalama

Bu metod Global ortalama ve Dönüşüm ortalama metodlarının bir birleşimi olarak düşünülebilir. Buradaki amaç, hem Global ortalama metodu gibi mutlak hatası düşük, hem de Dönüşüm ortalama metodu gibi görece hatası düşük bir pozisyon kestirimi yapabilmektir.

Kısım (7.3.)'de veriler üzerinden daha detaylı anlatılacağı üzere, Dönüşüm ortalama metodunun görece hatasının en yüksek olduğu zaman, Global ortalama metodunun mutlak hatasının en düşük olduğu zamana dek gelmektedir ve bu da Kalman Filtresi'nin GPS

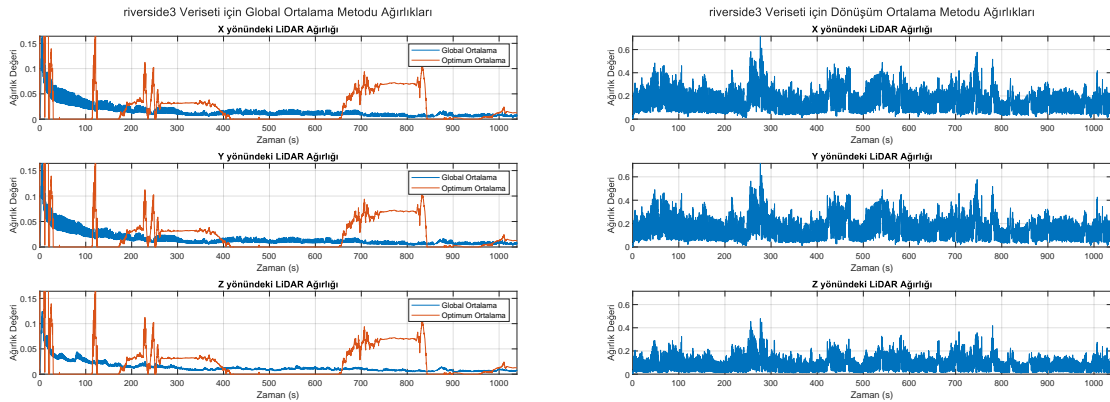
verisi ile doğrulama yapmasından sonraki ilk füzyon zamanıdır(n). Bu durumdan istifade ederek tam olarak bu zamandaki Global Ortalama pozisyon kestirimi $T_G(n)$ ile bu zamandan beri Dönüşüm Ortalama metodunun yaptığı toplam dönüşüm kestirimini ($T_D^{-1}(n)T_D(k)$) Denklem (71)'deki gibi Düzeltilmiş Ortalama kestirimi $T_C(k)$ elde etmek için kullanabiliriz.

$$T_C(k) = T_G(n)T_D^{-1}(n)T_D(k), \quad n < k \quad (71)$$

Bu denklem, LOAM algoritmasının haritalama adımı sonucunu, LiDAR Odometrisi sonucunu düzeltmek için kullanan denklemin (Denklem (20)) aynısıdır. Zaten bu fikrin ilham kaynağı da bu denklem olmuştur.

7.3. Sonuçlar

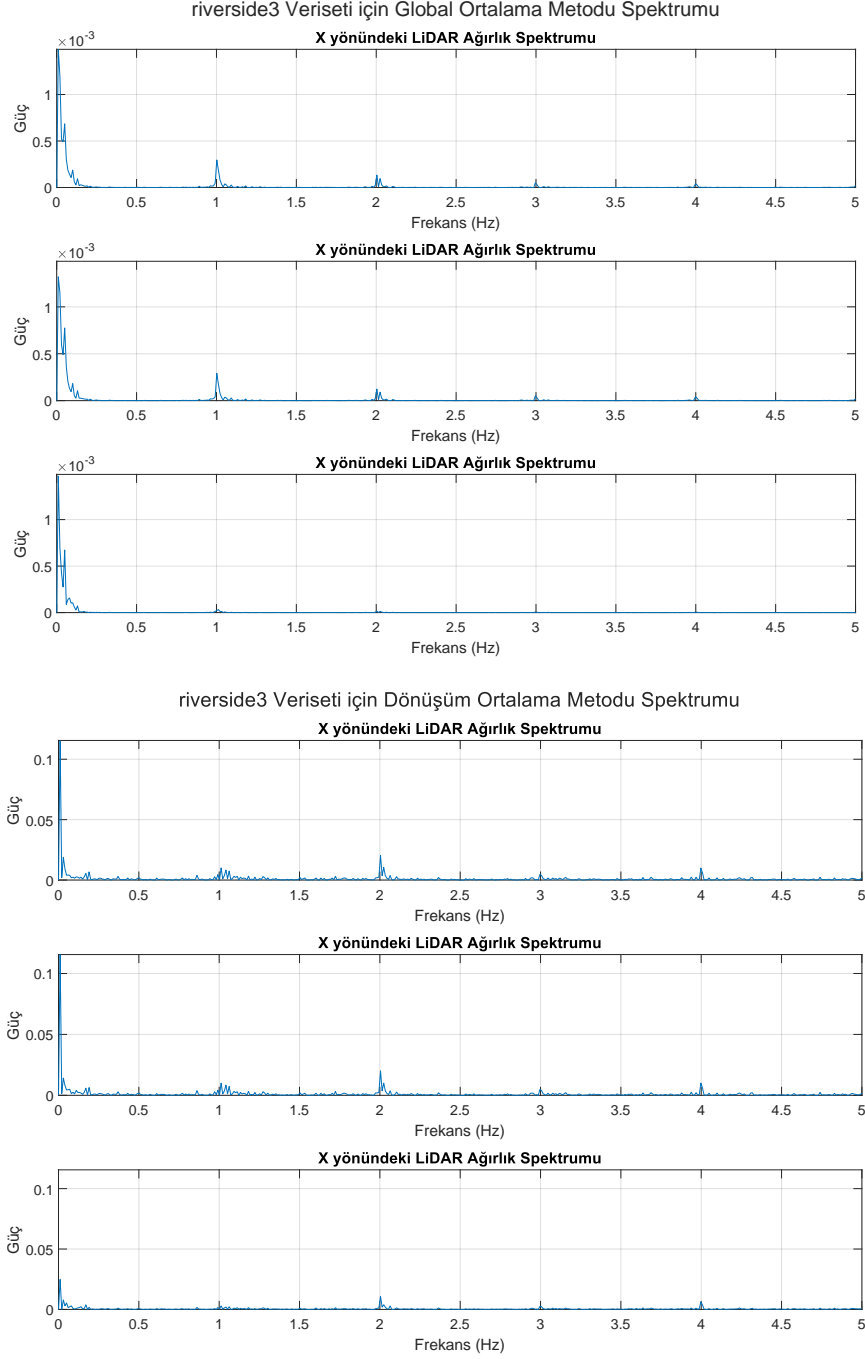
Dönüşüm Ortalama ve Global Ortalama metotlarının ağırlıklarını incelediğimizde, buradaki değerlerin gürültü bir sinyale benzer bir karakteristiğe sahip olduğunu gözlemledik.



Şekil 7.4 Riverside 3 Veriseti için Global ve Dönüşüm Ortalama Metotlarında kullanılan LiDAR Sensörü Ağırlıkları

Sensör ağırlıklarının gürültülü görünümünün nedenini anlamak için bu sinyallere Lomb-Scargle periodogramı[55] ile frekans analizi yaptık. Lomb-Scargle periodogramı, düzensiz zaman örnekleme verilerinde sinyallerin frekans spektrumunu analiz etmek için kullanılan, Fourier dönüşümüne dayalı bir yöntemdir. Analiz sonucunda, Şekil (7.5), (??)'de görüleceği üzere bu karakteristiğin gücünün 1 Hz ve harmoniklerindeki frekanslarda

toplanmıştır. Bu değer GPS sensörünün frekansına denk gelmektedir. Daha detaylı inceleme için GPS sensör verilerinin farklı oranlarda alt örneklerini aldığımızda ise bu frekans değerlerinin aynı oranlarda değiştiğini gözlemledik.



Şekil 7.5 Riverside 3 Veriseti için Global ve Dönüşüm Ortalama Metotlarında kullanılan LiDAR Sensörü Ağırlıklarının Spektrumu

Bu durumun nedeni, LiDAR sensörü ağırlığının Kalman Filtresi'nin pozisyon varyansı ile pozitif korelasyona sahip olmasıdır ve Kalman Filtresi'nin pozisyon varyansı doğrudan pozisyon ölçümü yapan tek sensör olan GPS ölçümü sonrası azalmaktadır. Böylece sistem, GPS sensörü ölçümü ardından Kalman Filtresi'nin ağırlığını arttırmaktadır. Bu andan sonra yapılan ilk tahminde pozisyon varyansı artmakta ve Kalman Filtresi'nin ağırlığı azalmaktadır ve bu döngü GPS sensörünün frekansında tekrarlanmaktadır.

Global ortalama metodu ağırlığı ile GPS frekansının ilişkisi doğrudan anlaşılabilir. Dönüşüm ortalama metodunda ise, varyans kestiriminin üst limiti hız, ivme varyansları ve proses gürültüsüne bağlıdır ve bu durumdan etkilenmez. Öte yandan alt limitin değeri ölçüm sonrası 0'a düşebilmektedir. Bu da GPS düzeltmesi sonrası Kalman Filtresi'nin pozisyon kestiriminde yapılan güncellemeyi daha büyük bir ağırlıkla ortalamaya yansıtılmaktadır. Bu dönüşüm sonucu pozisyon düzeltmesini de içerdiği için, gerçekte yapılan hareketle farklılıklar göstermekte ve o andaki görece pozisyon hatasını arttırmaktadır. Öte yandan bu düzeltme, kestirimin mutlak hatasını düşürmektedir. Sonuç olarak elimizde bulunan iki metottan Dönüşüm Ortalamasının en başarısız olduğu an, Global Ortalamasının en başarılı olduğu ana denk gelmektedir. Bu durumdan istifade etmek için Düzeltilmiş ortalama metodunu önerdik.

Tablo (7.1) ve (7.2)'e baktığımızda altı farklı sonuç görmekteyiz. Bunlardan Kalman Filtresi ve LiDAR sensörlerden elde edilen kestirim sonuçlarını, Optimal Ortalama ise bu iki sensör sonucunun doğrusal kombinasyonu ile yapılabilecek en iyi kestirim sonucunu vermekte ve referans görevi görmektedir. Global Ortalama özellikle mutlak hatada, Dönüşüm Ortalama özellikle görece hatada, Düzeltilmiş Ortalama ise iki hata metriğinde de başarılı sonuç vermesini beklediğimiz sensör füzyonu sonuçlarıdır.

Hataların ölçümünü yapmak için, SLAM algoritmalarını değerlendirmek için oluşturulmuş evo[56] python paketini kullandık. Değerlendirme sırasında paket içerisinde bulunan Umayema'nın hizalama algoritmasını[57], ilk 90 derecelik dönüşün ardından gelen düz yolun bitimine kadar olan bölümü hizalayacak şekilde kullandık. Bu hizalamanın nedeni, herhangi bir pozisyon ölçümü almayan LiDAR odometrisinin, başlangıçtaki küçük bir açı

	RMSE	Ort.	Medyan	Std.	Min	Max
Optimal Ortalama	5.96	5.31	5.19	2.71	0.0	26.65
Global Ortalama	6.41	5.79	5.55	2.77	0.13	27.68
Kalman Filtresi	6.73	5.88	5.59	3.28	0.15	35.52
Düzeltilmiş Ortalama	6.07	5.46	5.13	2.64	0.11	27.35
LiDAR	357.36	284.94	232.19	215.68	1.05	982.52
Dönüşüm Ortalama	129.29	116.06	124.0	56.97	1.18	223.24

Tablo 7.1 Riverside 3 Veriseti için Mutlak Hata Tablosu. Hatalar metre cinsinden verilmiştir.

	RMSE	Ort.	Medyan	Std.	Min	Max
Optimal Ortalama	0.65	0.28	0.12	0.59	0.0	13.12
Global Ortalama	1.16	0.59	0.23	0.99	0.0	16.89
Kalman Filtresi	0.99	0.39	0.15	0.91	0.0	16.03
Düzeltilmiş Ortalama	1.11	0.54	0.24	0.96	0.0	13.82
LiDAR	0.68	0.14	0.03	0.66	0.0	11.54
Dönüşüm Ortalama	0.99	0.45	0.22	0.88	0.0	15.27

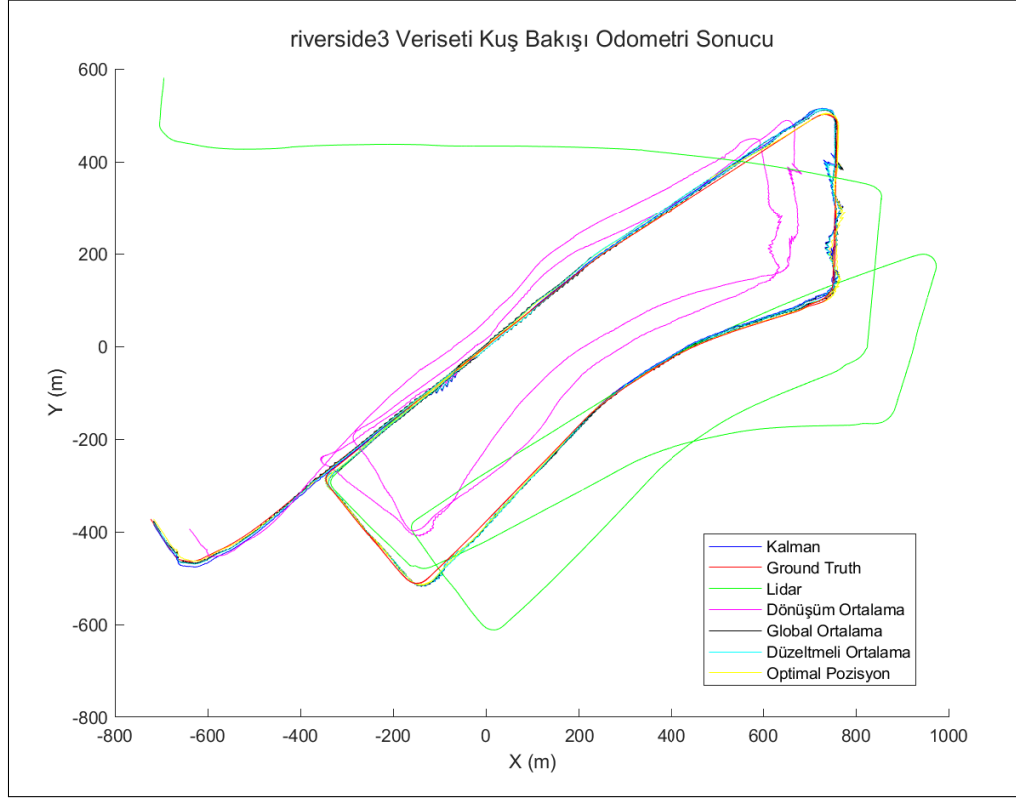
Tablo 7.2 Riverside 3 Veriseti için Görece Hata Tablosu. Hatalar 15 metre yer değiştirme için hesaplanan yüzdesel yer değiştirme hatasına denk gelmektedir.

farkından bile çok fazla etkileniyor olmasıdır. Ancak bu hizalamanın kestirim başarısına olan etkisini sınırlamak için, hizalamayı 3 boyutta düzgün hizalama yapabilmek için gerekli minimum veri ile gerçekleştirdik. Bu veri de başlangıçtan itibaren yapılan ilk "L" şeklindeki harekete denk gelmektedir. Son olarak bu hizalama, Optimal Ortalama kestiriminin başarısını düşürdüğü için, bu kestirimin hata ölçümünü hizalama yapmadan aldık.

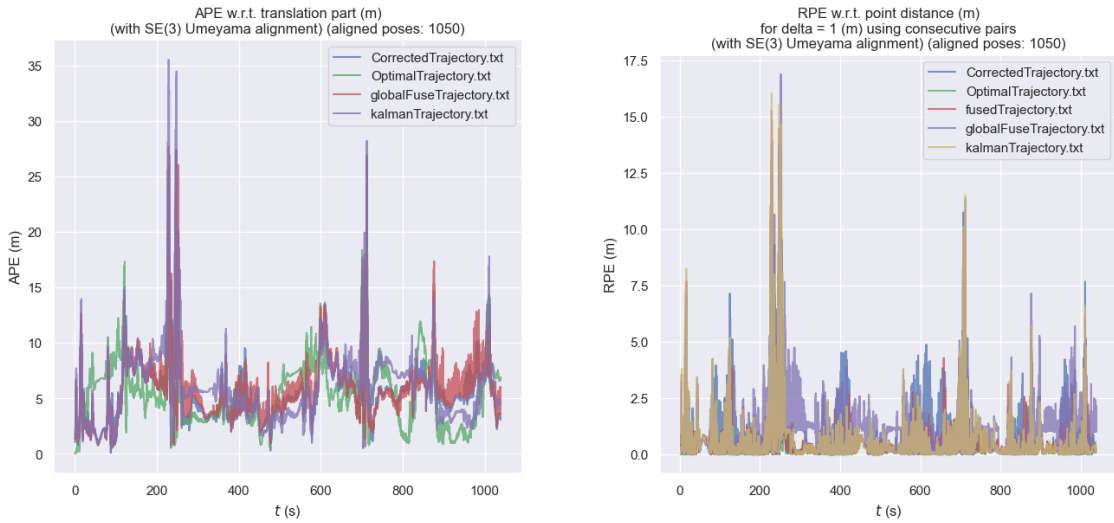
Mutlak hata tablosuna (7.1) göre, Optimal Ortalama yöntemi, beklendiği üzere tüm metriklerde en düşük hatayı göstermiştir. Özellikle Düzeltilmiş Ortalama yöntemi, Optimal Ortalama'ya en yakın sonuçları elde etmiş, hem RMSE hem de standart sapma değerleri açısından diğer yöntemlere göre daha başarılı olmuştur. Kalman Filtresi, Global Ortalama ve Dönüşüm Ortalama yöntemleriyle kıyaslandığında, nispeten daha yüksek hata oranlarına sahiptir. LiDAR sonuçları ise, muzdarip olduğu hata birikimi nedeni ile açık ara en yüksek hata değerlerini göstermiştir.

Görece hata tablosunda (7.2), Optimal Ortalama yöntemi yine tüm metriklerde en iyi sonuçları sağlamıştır. Dikkat çekici bir şekilde, LiDAR görece hata açısından daha düşük

bir ortalama değere sahiptir. Kalman Filtresi ve Dönüşüm Ortalama yöntemleri, görece hata açısından benzer sonuçlar sunarken, Düzeltmiş Ortalama bunların gerisinde kalmıştır.



Şekil 7.6 Riverside 3 Veriseti Sonucunun Kuşbakışı Görüntüsü



Şekil 7.7 Riverside 3 Veriseti için Mutlak (sol) ve Görece (sağ) Pozisyon Hatası Sonuçları

Kestim sonuçları ve zamana bağlı kestirim hataları Şekil (7.6) ve (7.7)'de görülebilir.

	RMSE	Ort.	Medyan	Std.	Min	Max
Optimal Ortalama	21.49	16.62	11.37	13.62	0.0	60.19
Global Ortalama	23.21	18.94	14.53	13.42	1.07	61.82
Kalman Filtresi	23.11	17.61	10.68	14.96	0.89	64.12
Düzeltilmiş Ortalama	22.9	18.65	14.29	13.3	0.87	61.61
LiDAR	232.64	190.95	147.88	132.89	11.27	482.57
Dönüşüm Ortalama	319.64	294.61	291.13	124.01	5.4	503.77

Tablo 7.3 DCC2 Veriseti için Mutlak Hata Tablosu

	RMSE	Ort.	Medyan	Std.	Min	Max
Optimal Ortalama	1.92	1.11	0.63	1.57	0.0	17.7
Global Ortalama	2.85	1.83	1.14	2.19	0.0	17.61
Kalman Filtresi	2.18	1.23	0.66	1.79	0.0	17.7
Düzeltilmiş Ortalama	2.14	1.37	0.82	1.64	0.0	15.8
LiDAR	1.36	0.6	0.07	1.22	0.0	25.08
Dönüşüm Ortalama	1.96	1.17	0.72	1.57	0.0	25.28

Tablo 7.4 DCC2 Veriseti için Görece Hata Tablosu

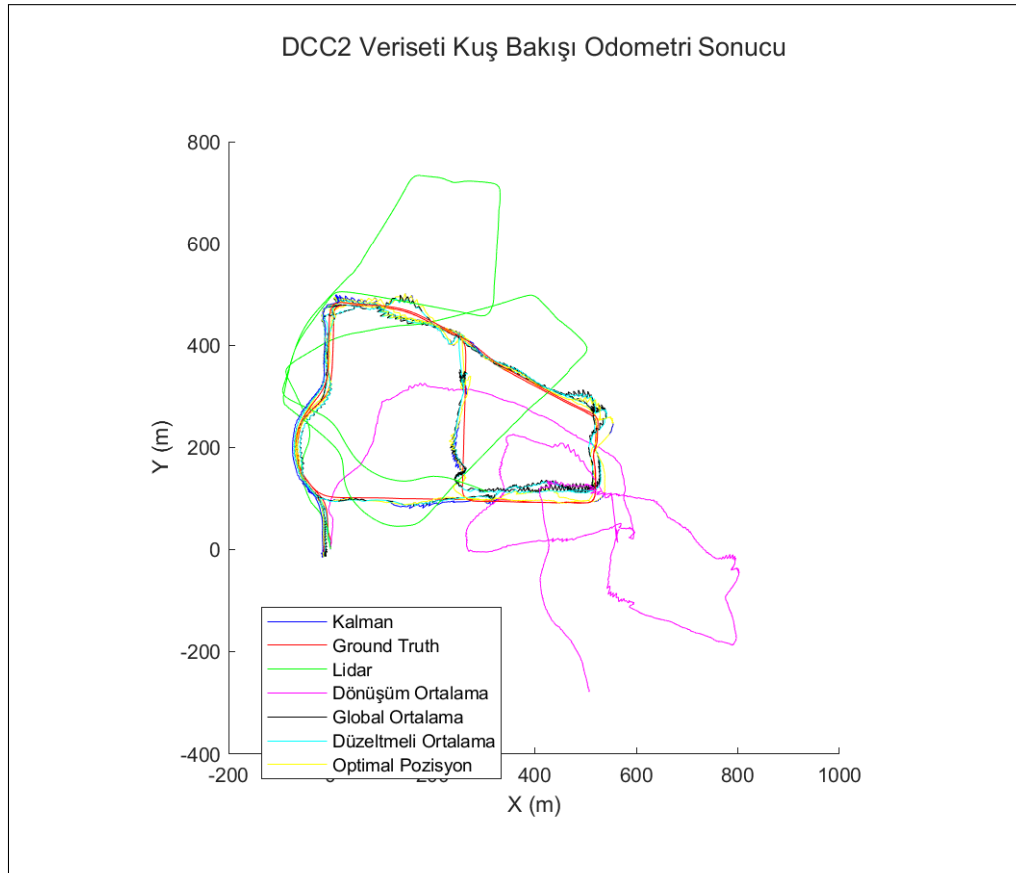
DCC2 veriseti için mutlak hata tablosuna (Tablo (7.3)) bakıldığında, Optimal Ortalama yöntemi beklendiği üzere tüm metriklerde (RMSE, ortalama, medyan, vb.) en düşük hatalara sahiptir ve bu açıdan referans görevi görmektedir. Onu takiben, Düzeltilmiş Ortalama ve Kalman Filtresi görece yakın sonuçlar sunmaktadır. RMSE değerinde Düzeltilmiş Ortalama (22.9 m), hem Global Ortalama (23.21 m) hem de Kalman Filtresi (23.11 m) yöntemlerine kıyasla daha iyi performans göstermiştir. Ancak ortalama değere bakıldığında Kalman Filtresi (17.61 m), Düzeltilmiş Ortalama (18.65 m) yöntemine göre daha düşük bir hataya sahiptir. Bu da her bir metrikte yöntemlerin farklı avantajlar sunduğunu göstermektedir. LiDAR ve Dönüşüm Ortalama yöntemleri ise ciddi miktarda hata birikimi sergilemiş, en yüksek RMSE değerlerine (sırasıyla 232.64 m ve 319.64 m) sahip olmuşlardır. Özellikle Dönüşüm Ortalama yöntemi, ortalama ve medyan hatalarda da oldukça yüksek değerlere sahip olup, tek başına uygulanmasının sağlıklı bir kestirim sağlamayacağı açıkça görülmektedir.

Görece hata tablosunda (Tablo (7.4)) ise Optimal Ortalama, birçok metrikte yine güçlü bir performans sergilemekle birlikte, LiDAR dikkat çekici şekilde bazı metriklerde (RMSE ve

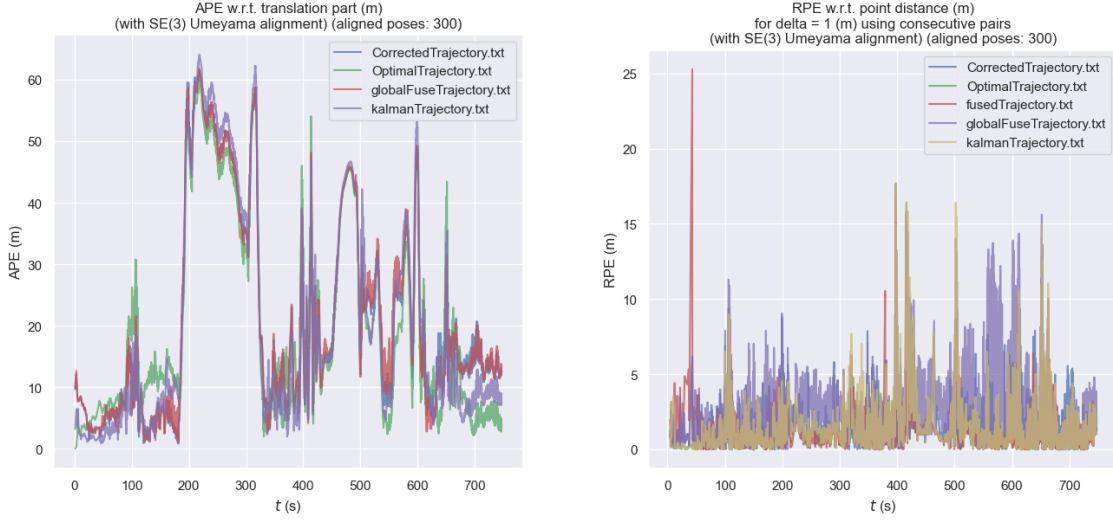
ortalama gibi) daha düşük hatalara sahiptir. Örneğin LiDAR'ın RMSE değeri (1.36) Optimal Ortalamadan (1.92) daha düşüktür ve ortalama hatası da (0.60) yine en düşük seviyededir. Ancak maksimum hata değerine bakıldığında (25.08), LiDAR yöntemi zaman zaman yüksek sapmalara neden olabilmektedir.

Kalman Filtresi (2.18 RMSE) ve Dönüşüm Ortalama (1.96 RMSE) yöntemleri görece benzer sonuçlar sunarken, Düzeltmiş Ortalama (2.14 RMSE) bu ikilinin gerisinde kalmaktadır. Global Ortalama yöntemi (2.85 RMSE) ise bu tabloda en yüksek görece hataya sahiptir. Sonuç olarak, görece hata açısından LiDAR beklenenden daha iyi ortalama ve RMSE değerleri sunsa da zaman zaman çok yüksek maksimum hatalar ürettiği, Optimal Ortalama yöntemi ise tutarlı bir şekilde en iyi genel performansı sağladığı görülmektedir.

Kestirim sonuçları ve zamana bağlı kestirim hataları Şekil (7.8) ve (7.9)'de görülebilir.



Şekil 7.8 DCC 2 Veriseti 3B Odometri Sonucu ve Sonucun Kuşbakışı Görüntüsü



Şekil 7.9 DCC 2 Veriseti için Mutlak ve Görece Pozisyon Hatası Sonuçları

Düzeltilmiş ortalama metodu ise riverside 3 ve DCC 2 verisetlerinde kaynak aldığı sensör verilerinden daha başarılı sonuç verebilmiştir. Bu başarımın tüm verisetlerinde tekrarlanamamış olması kesin bir yargıya varmanın önüne geçmekle beraber, Hausdorff Mesafesi metriğinin bir potansiyeli olduğunu göstermektedir.

8. BAĞLAŞIK METOTLAR ve İLERİ ÇALIŞMA

*All that is gold does not glitter,
Not all those who wander are lost.*

—BILBO BAGGINS, *Riddle of Strider*

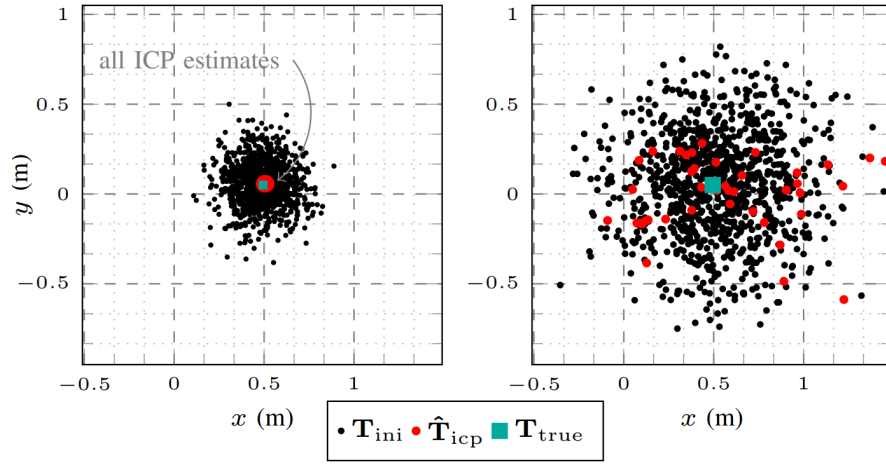
Çalışma sürecinde, LOAM algoritması ile EKF'yi bağlaşıklık (coupled) olarak kullandığımız sensör füzyonu metotları da denedik. Bağlaşıklık metotların diğer metotlardan farkı, sensör füzyonunu ham veri ile yapmasıdır. Biz buradaki metotlarda haritalama adımı EKF sonucunu da optimizasyon sürecine dahil ederek doğruluğu artırmayı amaçladık.

LiDAR odometrisini EKF'ye ölçüm olarak verirken yine Hausdorff Mesafesi ile elde ettiğimiz varyans değerini kullandık. Ayrıca EKF algoritması olarak robot_localization ROS paketini[58], LiDAR ölçümü güncellemesi sonrası sonuç verecek şekilde modifiye ederek kullandık.

Veriseti olarak ise, verileri .bag dosyası olarak kaydettiği için ROS ile kullanımı daha rahat olan M2DGR[34] verisetinin street4 sekansını kullandık. Bu verisetinin MulRan verisetine göre bir diğer avantajı, ROS REP-105'i[59] (Mobil Platformlar için Koordinat Sistemleri - Coordinate Frames for Mobile Platforms) IMU varyans bilgisi haricinde uygulamasıdır. Bu eksik bilgiyi ise IMU'nun veri kağıdından[60] elde ettik.

8.1. Kalman Filtresi ile İklendirilmiş Haritalama

İlk denediğimiz bağlaşıklık metot LOAM'ın haritalama adımı optimizasyon döngüsünü Kalman Filtresi'nin çıktısında elde edilen pozisyonda iklendirerek (initialization) başlatmaktı. Ancak bu metotta elde ettiğimiz pozisyon değerleri referans pozisyona yakın olmakla beraber, harita orijinal algoritma ile karşılaştırıldığında çok fazla gürültü içermiştir. Haritadaki bu gürültü, zaman geçtikçe birikerek, algoritmanın yer noktası bulutunu yanlış



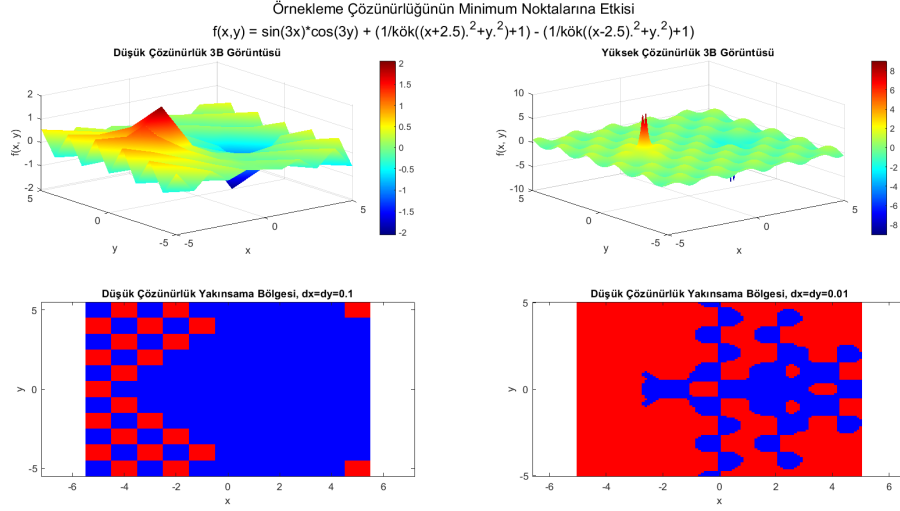
Şekil 8.1 Doğru (sol) ve Dağınık (sağ) pozisyanlarda iklendirilmiş ICP kestirimleri sonuçları. Siyah noktalar iklendirme pozisyonlarını, Kırmızı noktalar ICP kestirimlerini, M0 avi noktalar ise gerçek kestirim sonucunu göstermektedir[9].

eşlemesine ve sonuç olarak özellikle oryantasyon değerlerinde gerçek değerden sapmasına neden olmuştur.

ICP algoritmasının en kronik sorunlarından biri, doğru pozisyonda iklendirilmemesi durumunda yerel minimumlara takılma eğilimi göstermesidir, hatta Brossard ve ark.[9] bunun genellikle en dominant hata nedeni olduğunu belirtmekte ve bunu Şekil (8.1)'deki gibi göstermektedir.

LOAM kestirimlerini ICP kullanmadan yapmakla beraber aynı problemi benzer şekilde çözmesi nedeniyle benzer sorunlardan muzdarip olması beklenebilir. Öte yandan LiDAR Odometrisi adımıında sadece önceki nokta bulutunun referans alınması, bu bulutta daha az nokta bulunması nedeniyle problemde bulunan yerel minimum sayısını azaltarak global minimuma ulaşmayı mümkün kılıyor olabilir. Örneklem sayısının azaltılmasının yerel minimumlara etkisi Şekil (8.2)'de gösterilmiştir.

Haritalama algoritmasının iklendirmesini Kalman filtresinin çıktısı ile yapmak, algoritmayı bir yerel minimuma sokuyorken, bu iklendirme pozisyonunu, minimumlara takılma ihtimali düşük dönüşümlerin sonucu olarak almak algoritmayı global minimuma yakınsayabileceği bir noktadan başlatıyor olabilir. Bu durumu, bir bilyeyi engebeli bir yüzeyde ilerletmek yerine, bir bilye makinasında ard arda gelen huniler ile hedef pozisyona yönlendirmek



Şekil 8.2 Örnekleme sayısının azaltılması, yerel minimum sayısını azaltmakta ve uzaydaki bir noktadan global minimuma ulaşma ihtimalini arttırmaktadır. Şekilde $f(x,y) = \sin(3x)\cos(3y) + \frac{1}{\sqrt{(x+2.5)^2+y^2+1}} - \frac{1}{\sqrt{(x-2.5)^2+y^2+1}}$ fonksiyonunun 0.1 (sol) ve 0.01 (sağ) birim aralıklarda örneklenmiş hali görülmektedir. Bu örnekleme noktalarından gradyan vektörünün tersi yöndeki noktaya ilerleyerek global minimuma ulaşabilen noktalar mavi, ulaşamayanlar kırmızı renkte gösterilmiştir. Şekilden de anlaşılacağı üzere, daha küçük aralıklarla örnekleme yapıldığında, yerel minimuma takılma ihtimali artmaktadır.

gibi düşünebiliriz. Burada optimizasyon probleminin global minimum değerinin, gerçek gerçek pozisyona yakın olmakla beraber, tam olarak aynı sonucu vermiyor olabileceğine belirtmemiz gerekir.

Ancak bu konuda yapılan açıklama bir hipotez durumundadır ve bu konuda yapılabilinecek daha detaylı çalışmalar, tezin kapsamının dışına çıkmaktadır.

8.2. Kalman Filtresi Sonucu Destekli Maliyet Fonksiyonu

Guadagnino ve ark. Kinematic-ICP[6] makalesinde, LiDAR odometrisini teker odometrisinin sonucunda elde ettikleri pozisyonda iklendirmekte ve LiDAR odometrisinin bu pozisyona göre hesapladığı yer değiştirme miktarını da maliyet fonksiyonuna eklemektedirler. Buradaki amaçları, 2 Boyutta çalışıyor olmaları nedeniyle yüksek yer değiştirme ve düşük oryantasyon değişimi doğruluğuna sahip tekerlek odometrisi bilgisinden istifade etmektir.

Bu metotta, bu çalışmadan esinlenerek, haritalama adımımda çözülen optimizasyon problemine, Kalman Filtresi'nin sonucundan yapılan sapmayı ekleyerek, LOAM algoritmasında zamanla oluşacak hata birikimini azaltmayı amaçladık. Bu işlem sonucunda elde ettiğimiz yeni maliyet fonksiyonu Denklem (72-75)'de verilmiştir.

$$\hat{E}_k = \hat{P}_{LiDAR,k}^0 - \hat{P}_{Kalman,k}^0 \quad (72)$$

$$W_k = \frac{diag(cov(\hat{P}_{Kalman,k}^0))}{\sqrt[3]{det(cov(\hat{P}_{Kalman,k}^0))}} \quad (73)$$

$$d_G = |W_k^{-1} \times \hat{E}_k| \quad (74)$$

$$\hat{T}_k = \min_{\mathbf{T}_k} \sum_{i=1}^N (d_{\mathcal{E}}^2 + d_{\mathcal{H}}^2) + \sqrt{N} \times d_G^2 \quad (75)$$

Burada öncelikle LiDAR kestiriminin Kalman Filtresi kestiriminden ne kadar saptığını bulduk (Denklem (72)). Daha sonra ise Kalman filtresinin kovaryans matrisini normalize ederek, ağırlık matrisini (W_k) elde ettik (Denklem (73)). Bu ağırlık matrisi, ilk denklemde hesapladığımız hatanın hangi eksenine ne kadar güvenebileceğimizi göstermektedir. Denklem (74)'de hesapladığımız ağırlık matrisinin tersi ile çarpıp sonucun mutlak değerini aldık. Bu işlem aslında bir baz değişimi işlemi olarak görülebilir. Bir eksenindeki varyans değeri ne kadar büyükse, o eksenin efektif uzunluğunu ve dolayısıyla maliyete etkisini o oranda azalttık. Son olarak elde ettiğimiz bu yeni maliyeti d_G^2 , maliyet fonksiyonumuza ekledik. Ancak burada maliyet fonksiyonunda bulunan karşılıklılık sayısının kökü \sqrt{N} ile çarparak ağırlığını arttırdık. Burada N yerine \sqrt{N} kullanmamızın 2 nedeni vardır. Bunlardan birincisi, d_G^2 'nin $d_{\mathcal{H}}^2$ ve $d_{\mathcal{E}}^2$ 'te göre daha büyük olması nedeniyle, optimizasyon problemine eşit ağırlık ile eklendiği durumda problemi domine etmesidir. İkincisi ise N değerinin büyük olması, bulunan öznitelik sayısının fazla olduğu anlamına gelmektedir ve bu da LOAM'ın performansını arttırmaktadır.

Sonuçlar bir önceki kısımdaki gibi evo python paketi ile alınmıştır. Buradaki tek fark hizalamanın tüm veriyi hizalayacak şekilde yapılmasıdır. Önerilen yöntem, yalın LiDAR

	RMSE	Ort.	Medyan	Std.	Min	Max
Kalman Destekli	14.7150	13.1506	13.5750	6.6025	0.7352	31.4388
LeGO LOAM	14.6938	13.1271	13.5175	6.6019	0.6680	31.2379
LeGO LOAM_w_IMU	0.9959	0.7999	0.5255	0.5934	0.0111	2.3625

Tablo 8.1 Mutlak Hata Değerleri

	RMSE	Ort.	Medyan	Std.	Min	Max
Kalman Destekli	0.1831	0.1286	0.0951	0.1303	0.0000	1.1009
LeGO LOAM	0.1882	0.1306	0.0905	0.1355	0.0002	1.1595
LeGO LOAM_w_IMU	0.0588	0.0450	0.0350	0.0378	0.0000	0.2372

Tablo 8.2 Görece Hata Değerleri

odometrisinde bir miktar gelişme sağlamıştır, ancak bu metodun başarımlarını arttırımı, IMU verisinin kayma giderme sürecindeki başarımlarının altında kalmaktadır. Bu durum LiDAR odometrisi için kayma giderme sürecinin ne kadar kritik olduğunu bir kez daha göz önüne sermiştir.

Buradaki verisetindeki LiDAR odometri başarısının önceki kısımdan daha iyi olmasının bir nedeni ise senaryolar arasındaki farktır. MulRan verisetinde, LiDAR'ın arkasına konumlanan Radar, LiDAR'ın görüş açısının yaklaşık 70 derecelik bir bölümünü kapatmaktadır. Özellikle riverside verisetlerinde LiDAR'ın kalan görüş açısını yaklaşık yarısı çok fazla öznelik içermeyen nehir bölgesine bakmaktadır. Ayrıca bu veri setinde bir otomobil üzerinde daha yüksek hız ve ivmelerde veri alınması da LiDAR odometrisini olumsuz etkilemektedir.

8.3. İleri Çalışma

Bu çalışmada, Hausdorff Mesafesinin varyans kestirimi için kullanılabilir sonuçlar sunduğu ortaya konulmuştur. Öte yandan Optimal Ortalama Metodunun sonucu ile arasında hala gelişmeye açık bir alan mevcuttur. İlerleyen çalışmalarda farklı hata metrikleri, bu çalışmada kullanılan metotlar kullanılarak denenebilir. Örneğin, P³ LOAM makalesinde SVD ile hatanın yayılımı hesaplanarak varyans değeri kestirimi yapılmıştır.

Kullanılacak bir diğer yöntem, öznelik noktalarının ait oldukları doğru yada düzleme olan normal vektörlerinin kovaryansı olabilir. Buna benzer bir diğer alternatif ise, KISS-ICP

ve Kinematic-ICP makalelerinin maliyet fonksiyonunda kullandığı en yakın noktaya olan ortalama uzaklık yada uzaklık vektörlerinin kovaryansı olabilir.

Bu konuda potansiyeli olabilecek bir diğer metrik ise, özellikle son yıllarda makina öğrenmesi alanında sıkça kullanılan MMD (Maximum Mean Discrepancy - Maksimum Ortalama Farklılık)[61] metriğidir. Ancak bu yöntem makina öğrenmesi dışındaki alanlarda da uygulanmış ve başarılı sonuçlar elde edilmiştir[62]. Kısaca, iki verinin dağılımlarının ortalaması arasındaki farkı ölçen bu yöntem, eğer iki nokta bulutu düzgün bir şekilde eşlenirse düşük bir değer vermelidir. Özellikle ortalama üzerinden ölçüm yapması nedeniyle ayrık verilere karşı düşük hassasiyete sahip olması bu metriği Hausdorff Mesafesi metriğinin önüne çıkarabilir. Ancak burada dikkat edilmesi gereken konu, özellikle harita bulutunda haritanın keşfedilmiş tarafındaki nokta yoğunluğunun, keşfedilmemiş taraflarından fazla olması nedeniyle dönme eksenine etrafında normalize edilmesi gerekliliğidir. Bu konuda ise Giseop Kim ve ark.'nın Scan Context[21] yönteminde kullandıkları bakış açısı faydalı olabilir.

9. SONUÇ

Bu tez çalışmasında, LOAM algoritmasının eksikliklerini gidererek Hausdorff Mesafesi tabanlı bir varyans kestirim yöntemi geliştirilmiş ve bu yöntemin sensör füzyonu üzerindeki etkisi incelenmiştir. Çalışmanın temel amacı, LiDAR, IMU ve GPS gibi farklı sensörlerden elde edilen verilerin stokastik bir düzlemde birleştirilmesiyle, güvenilir ve hassas bir pozisyon kestirimi gerçekleştirmektir. LOAM algoritmasının temel problemi olan varyans bilgisi eksikliği, Hausdorff Mesafesi ile başarılı bir şekilde giderilmiştir. Bu yöntem, haritalama adımının varyansa etkisini göz önünde bulundurarak, LOAM algoritmasının çıktılarındaki güvenilirliği artırmış ve sensör füzyonunda daha sağlıklı bir temel oluşturmuştur. Ayrıca, Kalman filtresi ile entegre edilen bu yaklaşım, kısa ve uzun vadeli hata birikimlerini azaltarak LOAM algoritmasının performansını önemli ölçüde iyileştirmiştir.

Ortalama maliyet metriği ilk denediğimiz metriktir ve bu metrik ile yaptığımız sensör füzyonunun sonuçları çok stabil sonuçlar vermemiştir. Ayrıca ortalama, hem mutlak hem de görece hata açısından, sabit bir ağırlık ile yapılabilecek sensör füzyonunun altında kalmıştır. Bu metriğin bir diğer dezavantajı ise LOAM algoritmasının doğruluğunu büyük oranda etkileyen haritalama adımının varyansa etkisini göz ardı etmesidir. Tüm bunları göz önünde bulundurduğumuzda, bu metrik üzerinden yapılan varyans kestirimi sonuçlarında kaydadeğer bir başarı elde edilememiştir.

Hausdorff Mesafesi tabanlı varyans kestirimi, farklı verisetleri üzerinde test edilmiş ve önerilen yöntemin doğruluğu çeşitli deney ortamlarında değerlendirilmiştir. Çalışmada önerilen yöntemin sensör verilerini hatayı azaltacak şekilde birleştirebildiği ve daha stabil pozisyon kestirimi sunduğu gözlemlenmiştir. Test ettiğimiz dört verisetinin ikisinde (riverside 3 ve DCC2) pozisyonların, ikisinde (riverside 1 ve DCC 2) ise dönüşümlerin ters varyans ortalaması sonucunda Kalman Filtresinin sonucundan daha başarılı sonuç vermiştir. Ayrıca önerdiğimiz Düzeltilmiş Ortalama metodu tüm verisetlerinde, P³-LOAM[10] makalesinde referans olarak önerilen Optimal Ortalama sonucunda elde edilen mutlak

hata deęerinden %6.6'dan daha dk sapma gstermitir. Bunun sonucunda Hausdorf Mesafesi ile varyans kestirimini kullanabilecek yeni algoritmalar nerilmi ve bu konudaki alımalardan ilk sonular alınmıtır. Ayrıca, bu baęlamda, Kalman filtresinin son iki tahmini arasındaki dnmn varyansını hesaplamak iin zgn bir yntem nerilmi ve bu yntem sensr fzyonunda kullanılmıtır.

Donanım tarafında ise LiDAR, IMU, GPS ve stereo kamera gibi eitli sensrlerden veri toplayabilen bir sistem gelitirilmitir. Bu sistem, hem kapalı hem de aık alanlarda veri toplamaya uygun olacak ekilde tasarlanmı ve geniletilebilir bir altyapı sunmutur. Bu sistem ile kapalı alanda LiDAR ve stereo kamera odometri verilerini ieren, mutlak referansa sahip bir veriseti oluturulmutur.

Sonu olarak, bu tez alıması, LOAM algoritmasının eksikliklerini gidererek, sensr fzyonu ve varyans kestirimi alanında katkı saęlamaktadır. nerilen Hausdorff Mesafesi tabanlı varyans kestirim yntemi, SLAM algoritmalarında baarılı sensr fzyonu iin yeterli bir ara olarak deęerlendirilmekle birlikte, gelecekte bu yntemin geni veri setleri zerinde daha kapsamlı bir ekilde test edilmesi ve alternatif metriklerle karılatırılması nerilmitir.

KAYNAKÇA

- [1] Arthur Ed LeBouthillier. W. grey walter and his turtle robots, **2010**. Accessed: 2024-12-05.
- [2] Ji Zhang and Sanjiv Singh. Visual-lidar odometry and mapping: low-drift, robust, and fast. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. **2015**. doi:10.1109/ICRA.2015.7139486.
- [3] E Recherche, Et Automatique, Sophia Antipolis, and Zhengyou Zhang. Iterative point matching for registration of free-form curves. *Int. J. Comput. Vision*, 13, **1992**.
- [4] Guolai Jiang, Lei Yin, Guodong Liu, Weina Xi, and Yongsheng Ou. Fft-based scan-matching for slam applications with low-cost laser range finders. In *Applied Sciences*, volume 9, page 41. **2019**. doi:10.3390/app9010041.
- [5] Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters (RA-L)*, 8(2):1029–1036, **2023**. doi:10.1109/LRA.2023.3236571.
- [6] Tiziano Guadagnino, Benedikt Mersch, Ignacio Vizzo, Saurabh Gupta, Meher V.R. Malladi, Luca Lobefaro, Guillaume Doisy, and Cyrill Stachniss. Kinematic-ICP: Enhancing LiDAR Odometry with Kinematic Constraints for Wheeled Mobile Robots Moving on Planar Surfaces. volume arXiv:2410.10277. **2024**.
- [7] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and François Goulette. Ct-icp: Real-time elastic lidar odometry with loop closure. volume arXiv.2109.12979. **2021**. doi:10.48550/arXiv.2109.12979.

- [8] Andrea Censi. An accurate closed-form estimate of icp's covariance. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3167–3172. **2007**. doi:10.1109/ROBOT.2007.363961.
- [9] Martin Brossard, Silvere Bonnabel, and Axel Barrau. A New Approach to 3D ICP Covariance Estimation. *IEEE Robotics and Automation Letters*, **2020**. doi:10.1109/LRA.2020.2965391.
- [10] Tao Li, Ling Pei, Yan Xiang, Qi Wu, Songpengcheng Xia, Lihao Tao, Xujun Guan, and Wenxian Yu. P3-loam: Ppp/lidar loosely coupled slam with accurate covariance estimation and robust raim in urban canyon environment. *IEEE Sensors Journal*, 21:6660–6671, **2020**.
- [11] D. H. Titterton and J. L. Weston. *Strapdown inertial navigation technology*. IET, **2004**.
- [12] GPS.gov. Gps system overview, **2023**.
- [13] Ferhat Gölbol, Akif Hacinecipoglu, and Mustafa Mert Ankaralı. Developing a motion controller for autonomous agricultural robot otonom tarim robotu icin hareket planlayicinin gelistirilmesi. **2020**.
- [14] Trimble. Rtk gps overview, **2023**.
- [15] TKGM. Tusaga-aktif system, **2023**.
- [16] O. Altan and G. N. Kemper. The role of tusaga-aktif in precision agriculture. *Journal of Turkish Agricultural Engineering*, **2011**.
- [17] Rtcn standard 10403.3 differential gnss (global navigation satellite systems) services – version 3, **2016**.
- [18] N. El-Sheimy et al. Emerging trends in mobile mapping. *ISPRS Journal of Photogrammetry and Remote Sensing*, **2006**.

- [19] E. V. Ashburn, F. Conner, W. W. Hildreth Jr., and H. A. Thorpe. Study of high resolution wind measuring systems. phase ii- analysis. NASA Technical Report, Report Number: NASA-CR-67138, **1965**.
- [20] Donghai Yang, Yifan Liu, Qingjiu Chen, Meng Chen, Shaodong Zhan, Nim-kwan Cheung, Ho-Yin Chan, Zhidong Wang, and Wen Jung Li. Development of the high angular resolution 360° lidar based on scanning mems mirror. In *Sci Rep 13, 1540 (2023)*. Nature Publishing Group, **2023**.
- [21] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809. **2018**. doi:10.1109/IROS.2018.8593953.
- [22] Nikhil Jonnavithula, Yecheng Lyu, and Ziming Zhang. Lidar odometry methodologies for autonomous driving: A survey. volume arXiv:2109.06120. **2021**.
- [23] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 3, pages 2743–2748 vol.3. **2003**. doi:10.1109/IROS.2003.1249285.
- [24] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, **1981**. ISSN 0001-0782. doi:10.1145/358669.358692.
- [25] Kenneth Levenberg. A method for the solution of certain non – linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, **1944**.
- [26] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE/RSJ International*

- Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, **2018**.
- [27] Bengisu Özbay, Elvan Kuzucu, Mustafa Gül, Dilan Öztürk, Muhittin Taşcı, A. Mansur Arısoy, Halil Onur Şirin, and İsmail Uyanık. A high frequency 3d lidar with enhanced measurement density via papoulis-gerchberg. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 543–548. **2015**. doi:10.1109/ICAR.2015.7251509.
- [28] Elvan Kuzucu, Dilan Öztürk, Mustafa Gül, Bengisu Özbay, A Mansur Arısoy, H Onur Sirin, and Ismail Uyanik. Enhancing 3d range image measurement density via dynamic papoulis–gerchberg algorithm. *Transactions of the Institute of Measurement and Control*, 40(16):4407–4420, **2018**. doi:10.1177/0142331218759899.
- [29] Frank Dellaert and GTSAM Contributors. borglab/gtsam, **2022**. doi:10.5281/zenodo.5794541.
- [30] Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*. Foundations and Trends in Robotics, Vol. 6, **2017**.
- [31] Yibin Wu, Tiziano Guadagnino, Louis Wiesmann, Lasse Klingbeil, Cyrill Stachniss, and Heiner Kuhlmann. LIO-EKF: High frequency LiDAR-inertial odometry using extended Kalman filters. In *IEEE International Conference on Robotics and Automation (ICRA)*. **2024**.
- [32] Wenhong Wu, Xunyu Zhong, Dongjie Wu, Bushi Chen, Xungao Zhong, and Qiang Liu. Lio-fusion: Reinforced lidar inertial odometry by effective fusion with gnss/relocalization and wheel odometry. *IEEE Robotics and Automation Letters*, 8(3):1571–1578, **2023**. doi:10.1109/LRA.2023.3240372.
- [33] Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. Mulran: Multimodal range dataset for urban place recognition. In *Proceedings*

of the *IEEE International Conference on Robotics and Automation (ICRA)*. Paris, **2020**.

- [34] Jie Yin, Ang Li, Tao Li, Wenxian Yu, and Danping Zou. M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots. *IEEE Robotics and Automation Letters*, 7(2):2266–2273, **2021**.
- [35] NVIDIA Corporation. *NVIDIA Jetson Nano Developer Kit*, **2019**. Accessed: 2024-12-18.
- [36] NVIDIA Corporation. *NVIDIA Jetson Xavier NX Developer Kit*, **2020**. Accessed: 2024-12-18.
- [37] NVIDIA Corporation. *NVIDIA Jetson Orin NX Developer Kit*, **2022**. Accessed: 2024-12-18.
- [38] Jinyong Jeong, Younggun Cho, Young-Sik Shin, Hyunchul Roh, and Ayoung Kim. Complex urban dataset with multi-level sensors from highly diverse urban environments. *International Journal of Robotics Research*, 38(6):642–657, **2019**.
- [39] Maciej Trzeciak, Kacper Pluta, Yasmin Fathy, Lucio Alcalde, Stanley Chee, Antony Bromley, Ioannis Brilakis, and Pierre Alliez. Conslam: Periodically collected real-world construction dataset for slam and progress monitoring. In *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 317–331. Springer, **2023**.
- [40] Jie Yin, Hao Yin, Conghui Liang, Haitao Jiang, and Zhengyou Zhang. Ground-challenge: A multi-sensor slam dataset focusing on corner cases for ground robots. In *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–5. IEEE, **2023**.
- [41] Joshua Knights, Kavisha Vidanapathirana, Milad Ramezani, Sridha Sridharan, Clinton Fookes, and Peyman Moghadam. Wild-places: A large-scale dataset for lidar place recognition in unstructured natural environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. **2023**.

- [42] Velodyne Lidar Inc. *VLP-16 User Manual*. Velodyne Lidar Inc., **2016**. Accessed: 2024-12-18.
- [43] Stereolabs. *ZED Stereo Camera*, **2015**. Accessed: 2024-12-18.
- [44] Stereolabs. *ZED2 Stereo Camera*, **2019**. Accessed: 2024-12-18.
- [45] Xsens Technologies B.V. *MTi-7 Datasheet*, **2023**. Document Revision May 2023.
- [46] Ardic Research and Development Corporation. *ArNav SIG User Manual*, **2024**. Document Revision 1.0.13.
- [47] Stereolabs. *Zed ros wrapper*, **2025**. Accessed: 2025-01-12.
- [48] ROS Drivers Maintainers. *Velodyne ros drivers*, **2025**. Accessed: 2025-01-12.
- [49] Robust Field Autonomy Lab. *Lego-loam: Lightweight and ground-optimized lidar odometry and mapping*, **2025**. Accessed: 2025-01-12.
- [50] HU NeuroLab Robot Group. *Aruco ros package*, **2025**. Accessed: 2025-01-12.
- [51] International Olympic Committee. *10m air pistol mixed team qualification results - paris 2024 olympics*. <https://olympics.com/en/paris-2024/results/shooting/10m-air-pistol-mixed-team/qual----->. Accessed: 2024-12-21.
- [52] MathWorks. *insfilterAsync Documentation*, **2024**. [Accessed: December 24, 2024].
- [53] O. YILMAZ. *Loam_variance_estimation*. https://github.com/Urtii/LOAM_variance_estimation, **2024**. Accessed on 2024-12-25.
- [54] Darryl Morrell. *Lecture notes on the extended kalman filter*. https://www.cs.cmu.edu/~./motionplanning/papers/sbp_papers/kalman/ekf_lecture_notes.pdf, **1997**. Spring 1997, accessed on 2024-12-25.

- [55] Norman R. Lomb. Least-squares frequency analysis of unequally spaced data. *Astrophysics and Space Science*, 39(2):447–462, **1976**.
- [56] Michael Grupp. evo: Python package for the evaluation of odometry and slam. <https://github.com/MichaelGrupp/evo>, **2017**.
- [57] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, **1991**.
- [58] T. Moore and D. Stouch. A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, **2014**.
- [59] Open Source Robotics Foundation. Rep-105: Coordinate frames for mobile platforms, **2025**. Accessed: 2025-01-12.
- [60] Taobotics. Hfi imu documentation: Comprehensive introduction, **2025**. Accessed: 2025-01-12.
- [61] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, **2006**.
- [62] Osman Karagoz, Aysegul Kilic, Emin Aydın, Mustafa Ankarali, and İsmail Uyanık. Predictive uncertainty in state-estimation drives active sensing. *Bioinspiration & biomimetics*, 20, **2024**. doi:10.1088/1748-3190/ad9534.

EKLER

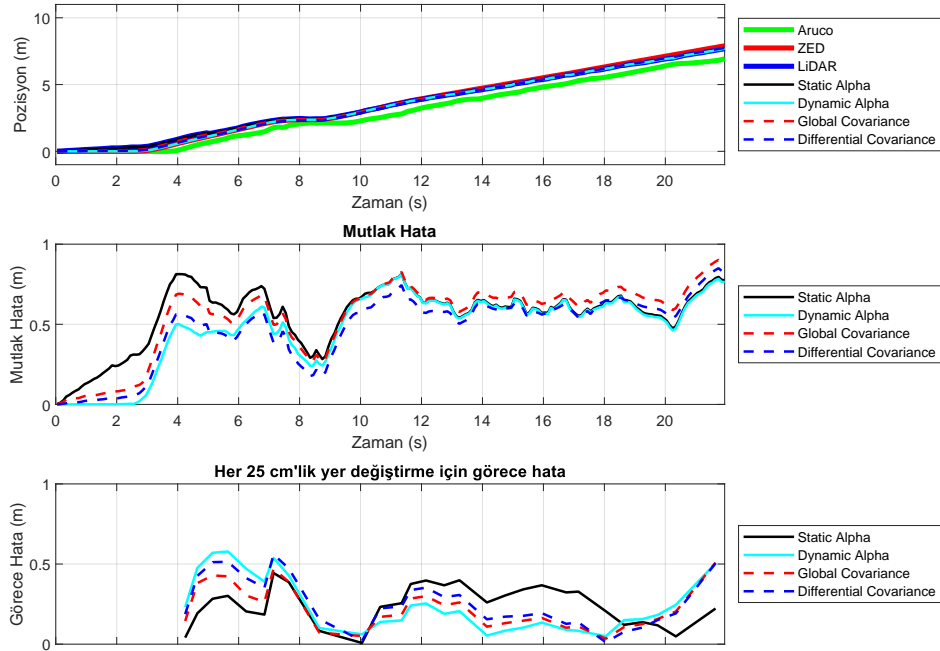
A. 1B Sensör Füzyonu Sonuçları

	Ort.	Medyan	Std.	RMSE
Sabit	0.5588	0.6080	0.0344	0.5886
Dinamik	0.4805	0.5608	0.0520	0.5316
Global	0.5518	0.6445	0.0515	0.5965
Dönüşüm	0.4761	0.5624	0.0479	0.5238

Kapalı Alan Veriseti Deney 07_02 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.2475	0.2577	0.0148	0.2749
Dinamik	0.2450	0.1860	0.0301	0.2985
Global	0.2238	0.1771	0.0178	0.2594
Dönüşüm	0.2630	0.2084	0.0250	0.3054

Kapalı Alan Veriseti Deney 07_02 için Görece Hata Tablosu



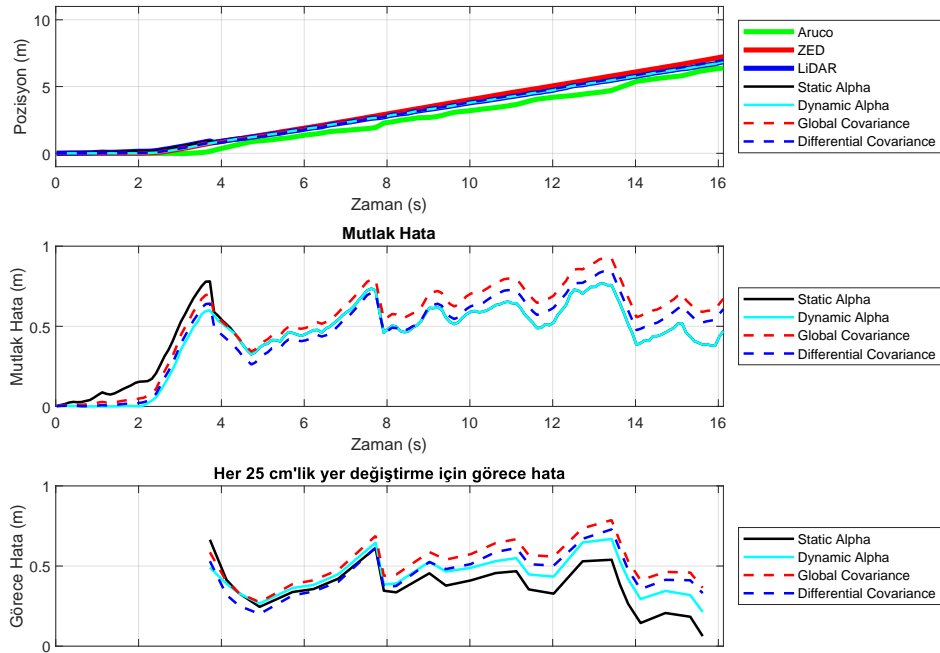
Kapalı Alan Veriseti Deney 07_02 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.4662	0.4905	0.0390	0.5061
Dinamik	0.4393	0.4844	0.0478	0.4904
Global	0.5340	0.5975	0.0649	0.5913
Dönüşüm	0.4669	0.5298	0.0553	0.5224

Kapalı Alan Veriseti Deney 07_03 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.3691	0.3562	0.0196	0.3937
Dinamik	0.4387	0.4337	0.0140	0.4538
Global	0.5195	0.5307	0.0172	0.5352
Dönüşüm	0.4576	0.4760	0.0180	0.4761

Kapalı Alan Veriseti Deney 07_03 için Görece Hata Tablosu



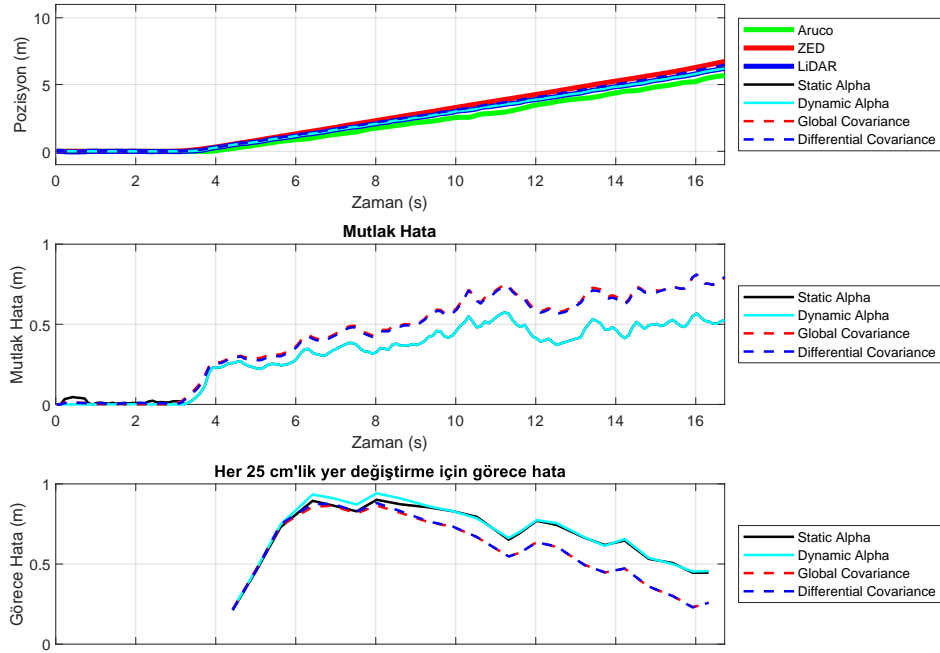
Kapalı Alan Veriseti Deney 07_03 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.3185	0.3705	0.0341	0.3680
Dinamik	0.3156	0.3705	0.0360	0.3679
Global	0.4346	0.4900	0.0709	0.5093
Dönüşüm	0.4289	0.4815	0.0695	0.5031

Kapalı Alan Veriseti Deney 07_04 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.6804	0.7136	0.0334	0.7035
Dinamik	0.6937	0.7256	0.0368	0.7186
Global	0.5788	0.5899	0.0468	0.6162
Dönüşüm	0.5822	0.5915	0.0488	0.6209

Kapalı Alan Veriseti Deney 07_04 için Görece Hata Tablosu



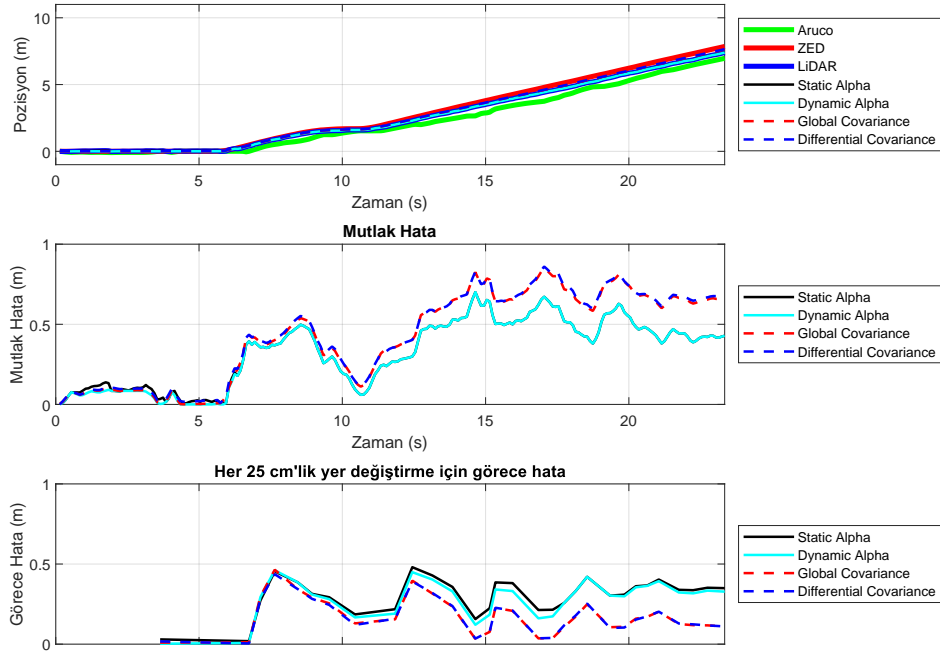
Kapalı Alan Veriseti Deney 07_04 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.3329	0.3936	0.0402	0.3883
Dinamik	0.3279	0.3936	0.0428	0.3875
Global	0.4267	0.4750	0.0755	0.5072
Dönüşüm	0.4350	0.4839	0.0750	0.5137

Kapalı Alan Veriseti Deney 07_05 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.3047	0.3339	0.0122	0.3235
Dinamik	0.2872	0.3198	0.0133	0.3087
Global	0.1756	0.1569	0.0131	0.2086
Dönüşüm	0.1743	0.1558	0.0123	0.2057

Kapalı Alan Veriseti Deney 07_05 için Görece Hata Tablosu



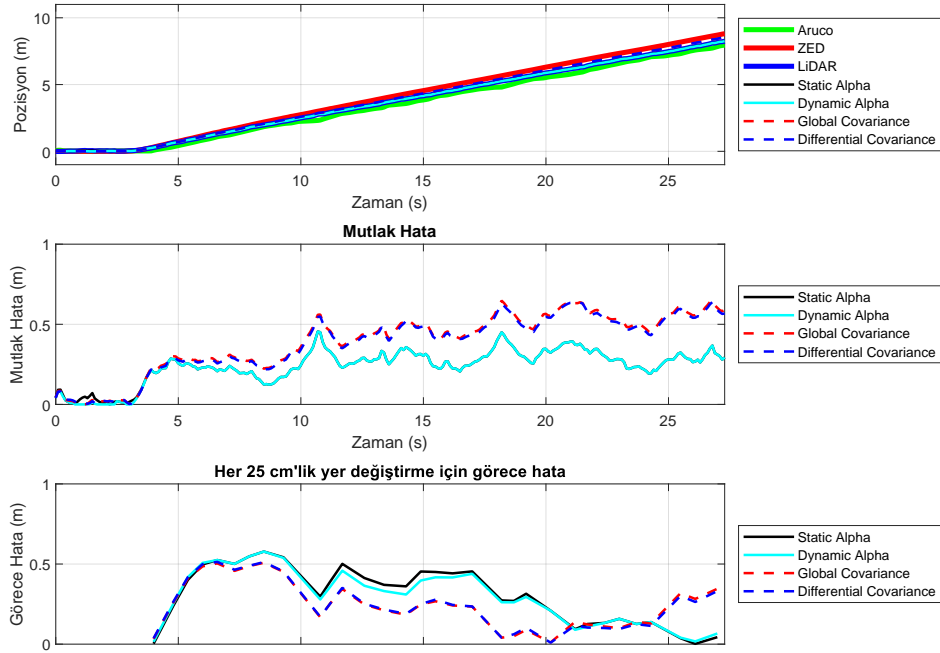
Kapalı Alan Veriseti Deney 07_05 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.2421	0.2589	0.0105	0.2629
Dinamik	0.2402	0.2589	0.0114	0.2627
Global	0.3908	0.4398	0.0340	0.4320
Dönüşüm	0.3806	0.4299	0.0325	0.4210

Kapalı Alan Veriseti Deney 07_06 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.3089	0.3377	0.0322	0.3558
Dinamik	0.2986	0.3024	0.0295	0.3431
Global	0.2459	0.2444	0.0226	0.2870
Dönüşüm	0.2455	0.2475	0.0235	0.2882

Kapalı Alan Veriseti Deney 07_06 için Görece Hata Tablosu



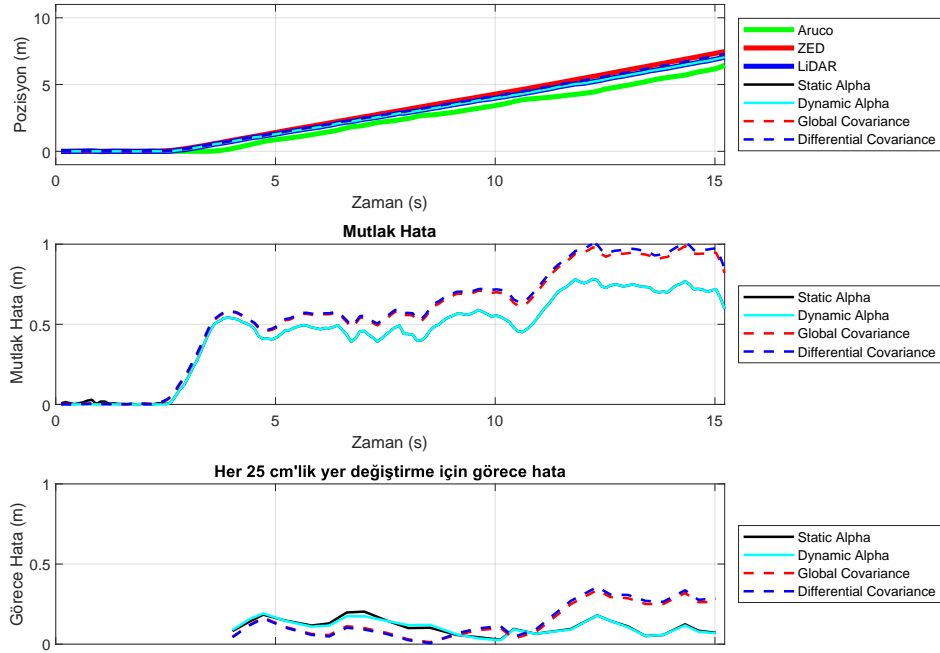
Kapalı Alan Veriseti Deney 07_06 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.4545	0.4866	0.0607	0.5166
Dinamik	0.4533	0.4866	0.0617	0.5165
Global	0.5592	0.5667	0.0991	0.6412
Dönüşüm	0.5714	0.5731	0.1038	0.6555

Kapalı Alan Veriseti Deney 07_07 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.1103	0.1023	0.0024	0.1204
Dinamik	0.1084	0.1096	0.0022	0.1179
Global	0.1503	0.1079	0.0111	0.1822
Dönüşüm	0.1565	0.1035	0.0129	0.1921

Kapalı Alan Veriseti Deney 07_07 için Görece Hata Tablosu



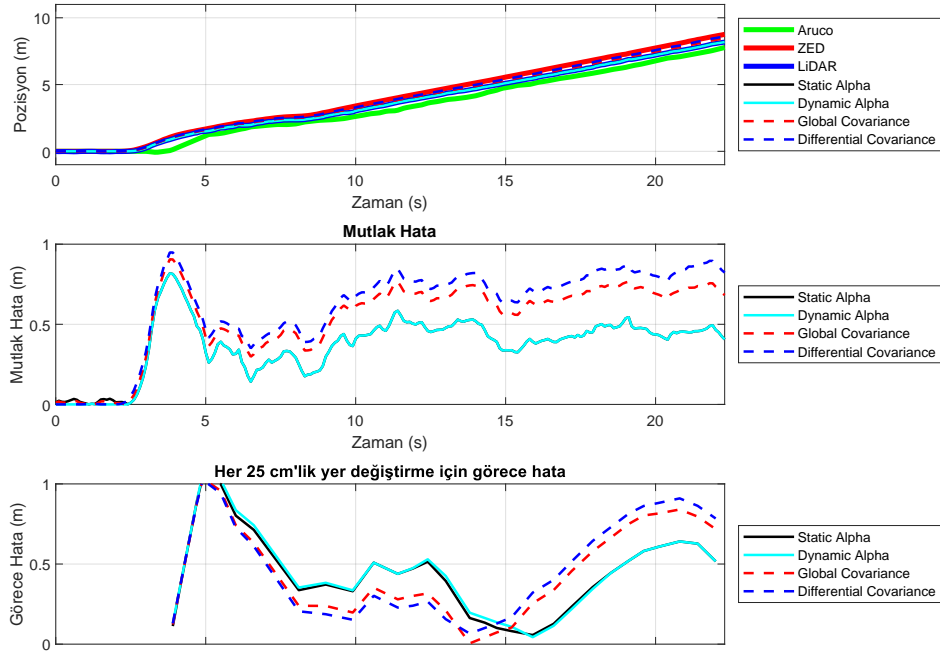
Kapalı Alan Veriseti Deney 07_07 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.3744	0.4203	0.0317	0.4143
Dinamik	0.3725	0.4203	0.0331	0.4143
Global	0.5406	0.6397	0.0573	0.5909
Dönüşüm	0.6065	0.7177	0.0732	0.6638

Kapalı Alan Veriseti Deney 07_08 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.4528	0.4439	0.0706	0.5228
Dinamik	0.4622	0.4431	0.0714	0.5318
Global	0.4730	0.3523	0.0917	0.5590
Dönüşüm	0.4891	0.4016	0.0994	0.5792

Kapalı Alan Veriseti Deney 07_08 için Görece Hata Tablosu



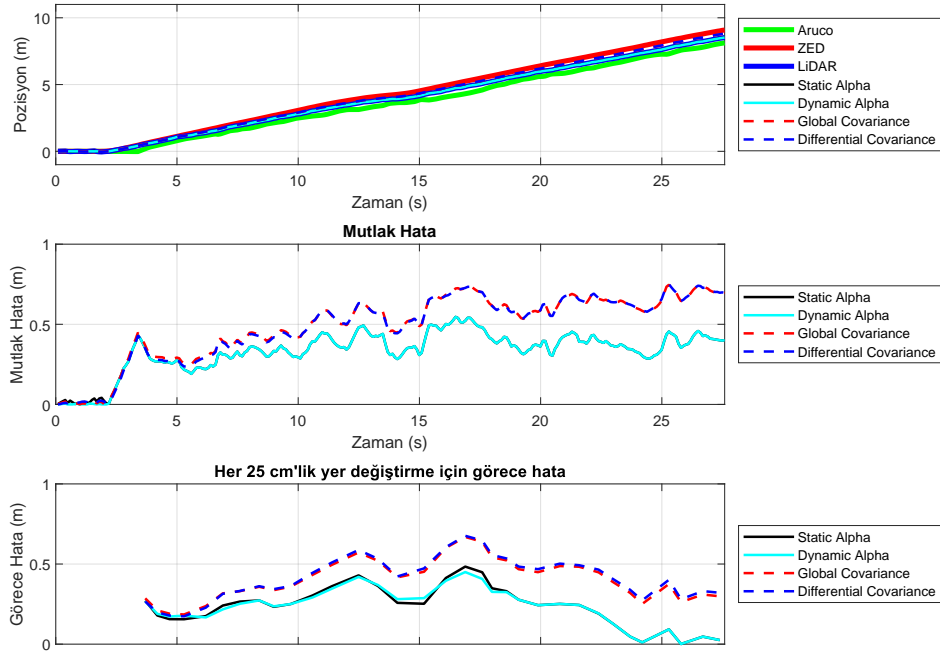
Kapalı Alan Veriseti Deney 07_08 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.3350	0.3558	0.0153	0.3570
Dinamik	0.3339	0.3558	0.0160	0.3569
Global	0.4901	0.5537	0.0409	0.5301
Dönüşüm	0.4867	0.5552	0.0421	0.5280

Kapalı Alan Veriseti Deney 07_09 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.2348	0.2503	0.0161	0.2660
Dinamik	0.2321	0.2503	0.0145	0.2606
Global	0.4004	0.3870	0.0168	0.4202
Dönüşüm	0.4094	0.4059	0.0181	0.4303

Kapalı Alan Veriseti Deney 07_09 için Görece Hata Tablosu



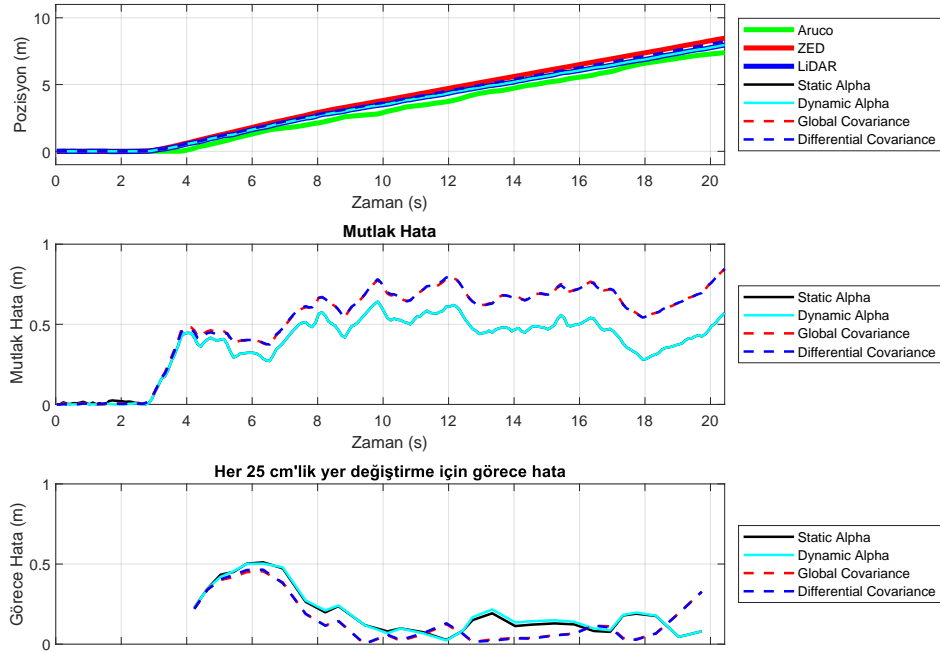
Kapalı Alan Veriseti Deney 07_09 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.3826	0.4455	0.0329	0.4232
Dinamik	0.3814	0.4455	0.0338	0.4232
Global	0.5187	0.6206	0.0622	0.5753
Dönüşüm	0.5184	0.6206	0.0626	0.5754

Kapalı Alan Veriseti Deney 07_10 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.1995	0.1502	0.0212	0.2455
Dinamik	0.2046	0.1682	0.0206	0.2485
Global	0.1577	0.1115	0.0222	0.2153
Dönüşüm	0.1585	0.1035	0.0230	0.2176

Kapalı Alan Veriseti Deney 07_10 için Görece Hata Tablosu



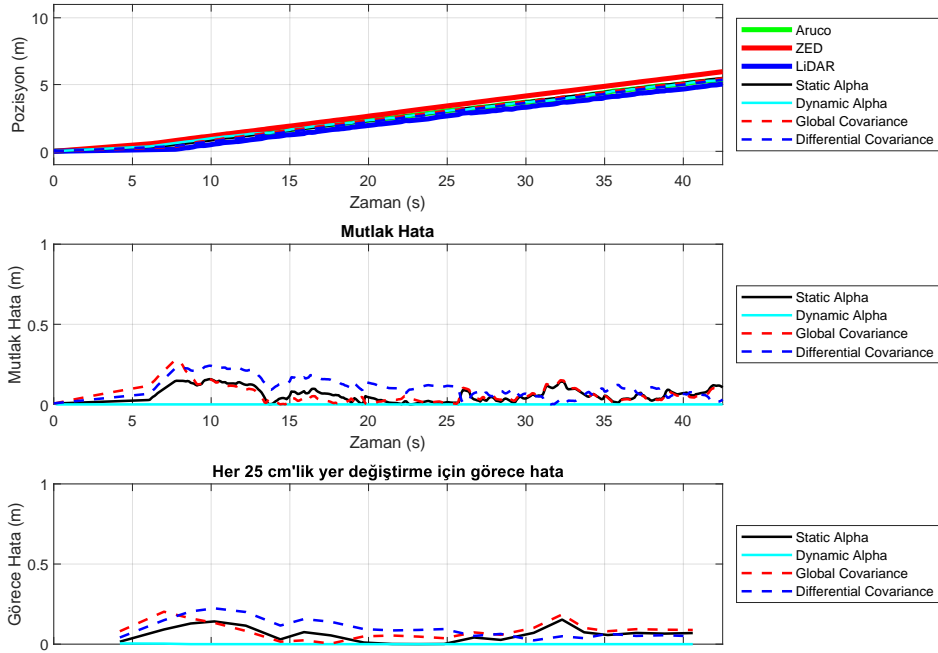
Kapalı Alan Veriseti Deney 07_10 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.0564	0.0433	0.0019	0.0712
Dinamik	0.0000	0.0000	0.0000	0.0003
Global	0.0648	0.0496	0.0030	0.0849
Dönüşüm	0.1003	0.0944	0.0039	0.1183

Kapalı Alan Veriseti Deney 11.01 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.0616	0.0658	0.0021	0.0762
Dinamik	0.0004	0.0000	0.0000	0.0012
Global	0.0839	0.0808	0.0027	0.0982
Dönüşüm	0.0972	0.0856	0.0036	0.1134

Kapalı Alan Veriseti Deney 11.01 için Görece Hata Tablosu



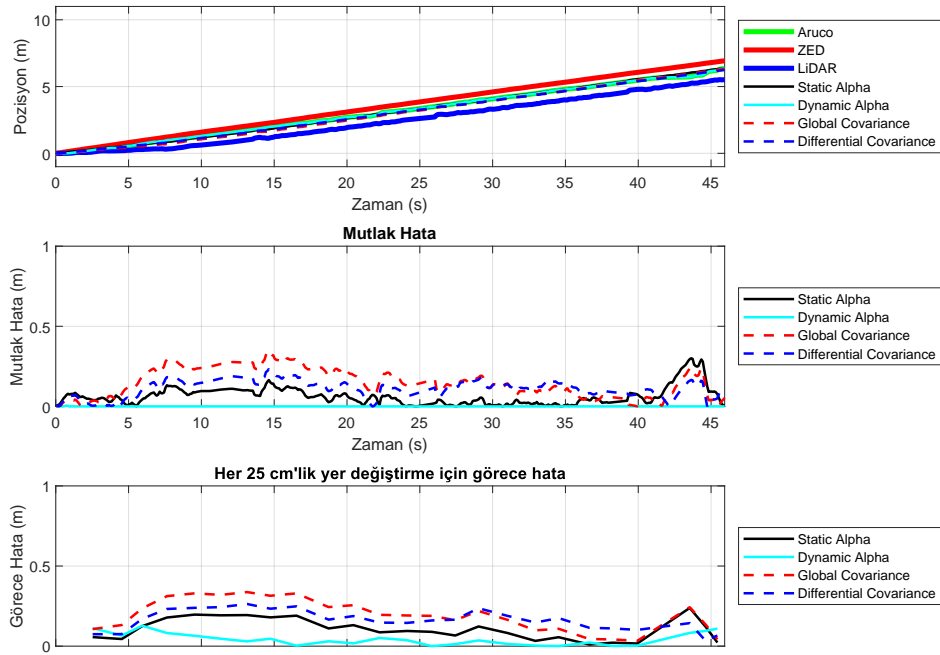
Kapalı Alan Veriseti Deney 11.01 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.0631	0.0513	0.0032	0.0849
Dinamik	0.0000	0.0000	0.0000	0.0003
Global	0.1479	0.1383	0.0080	0.1726
Dönüşüm	0.1105	0.1156	0.0027	0.1220

Kapalı Alan Veriseti Deney 11_02 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.1063	0.0947	0.0045	0.1251
Dinamik	0.0438	0.0358	0.0015	0.0582
Global	0.1913	0.1914	0.0100	0.2150
Dönüşüm	0.1627	0.1652	0.0041	0.1744

Kapalı Alan Veriseti Deney 11_02 için Görece Hata Tablosu



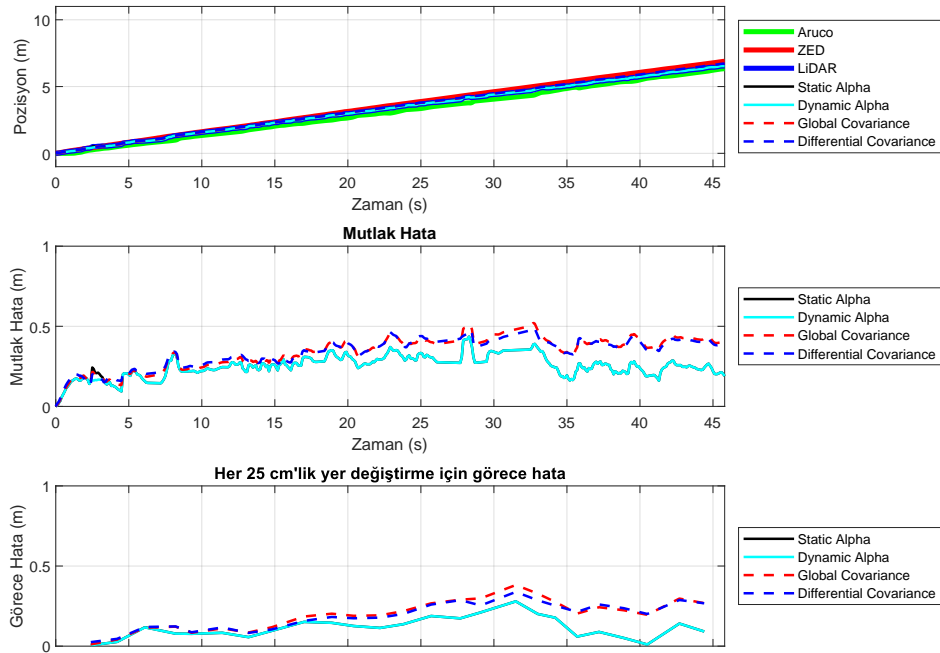
Kapalı Alan Veriseti Deney 11_02 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.2524	0.2570	0.0045	0.2610
Dinamik	0.2516	0.2570	0.0045	0.2605
Global	0.3412	0.3718	0.0100	0.3554
Dönüşüm	0.3383	0.3688	0.0083	0.3502

Kapalı Alan Veriseti Deney 11_03 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.1155	0.1137	0.0044	0.1327
Dinamik	0.1155	0.1137	0.0044	0.1326
Global	0.1978	0.2026	0.0088	0.2180
Dönüşüm	0.1897	0.2008	0.0071	0.2069

Kapalı Alan Veriseti Deney 11_03 için Görece Hata Tablosu



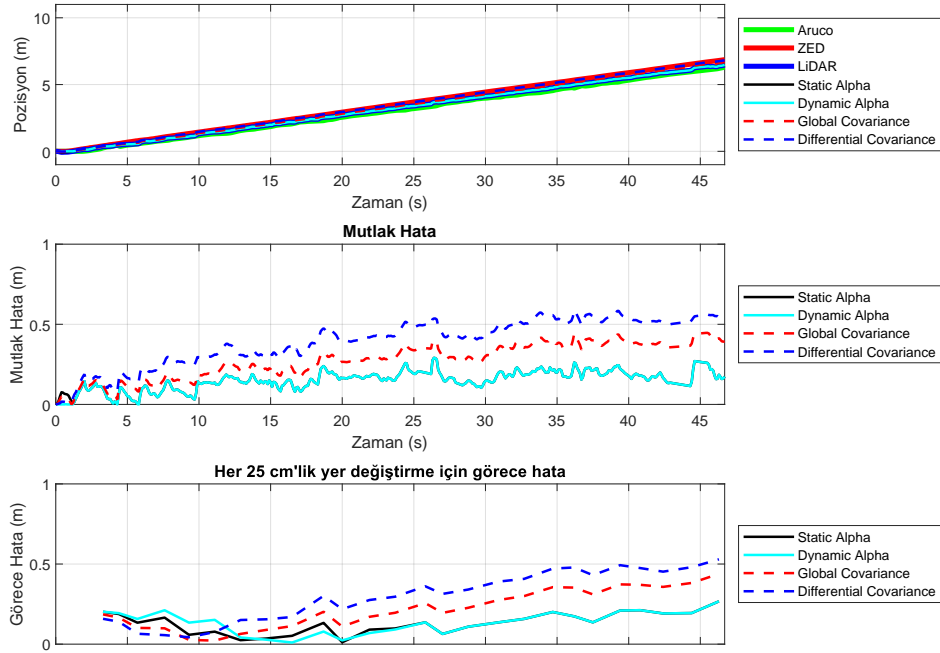
Kapalı Alan Veriseti Deney 11_03 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.1515	0.1592	0.0034	0.1623
Dinamik	0.1503	0.1592	0.0037	0.1621
Global	0.2748	0.2885	0.0120	0.2958
Dönüşüm	0.3937	0.4263	0.0205	0.4189

Kapalı Alan Veriseti Deney 11_04 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.1330	0.1360	0.0045	0.1484
Dinamik	0.1377	0.1439	0.0048	0.1535
Global	0.2207	0.1990	0.0149	0.2511
Dönüşüm	0.2973	0.3076	0.0250	0.3352

Kapalı Alan Veriseti Deney 11_04 için Görece Hata Tablosu



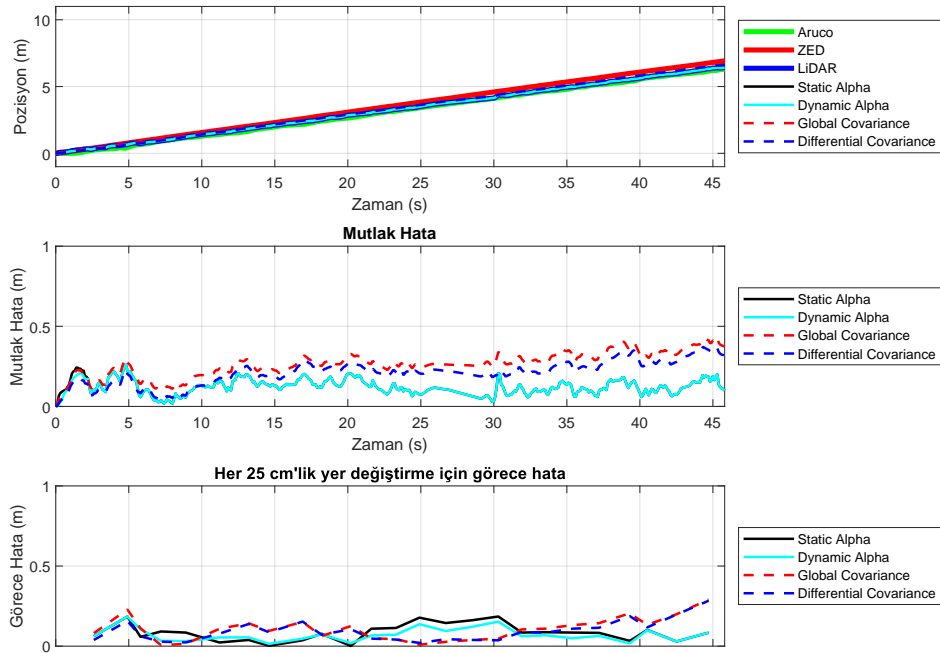
Kapalı Alan Veriseti Deney 11_04 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.1218	0.1174	0.0022	0.1305
Dinamik	0.1204	0.1174	0.0021	0.1289
Global	0.2632	0.2644	0.0057	0.2738
Dönüşüm	0.2095	0.2152	0.0055	0.2222

Kapalı Alan Veriseti Deney 11_05 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.0855	0.0847	0.0029	0.1004
Dinamik	0.0718	0.0647	0.0019	0.0833
Global	0.1063	0.1084	0.0053	0.1278
Dönüşüm	0.0940	0.0840	0.0042	0.1133

Kapalı Alan Veriseti Deney 11_05 için Görece Hata Tablosu



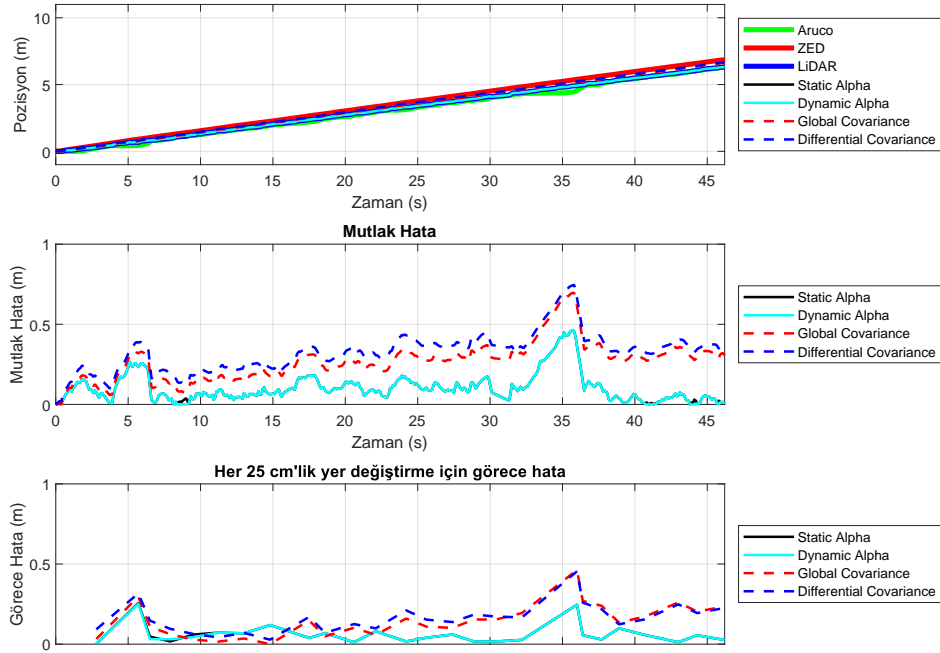
Kapalı Alan Veriseti Deney 11_05 1B Odometri Sonucu

	Ort.	Medyan	Std.	RMSE
Sabit	0.1031	0.0825	0.0078	0.1358
Dinamik	0.1023	0.0825	0.0080	0.1357
Global	0.2678	0.2776	0.0144	0.2934
Dönüşüm	0.3240	0.3344	0.0148	0.3460

Kapalı Alan Veriseti Deney 11_06 için Mutlak Hata Tablosu

	Ort.	Medyan	Std.	RMSE
Sabit	0.0614	0.0513	0.0039	0.0869
Dinamik	0.0606	0.0420	0.0039	0.0860
Global	0.1456	0.1411	0.0113	0.1791
Dönüşüm	0.1625	0.1540	0.0085	0.1860

Kapalı Alan Veriseti Deney 11_06 için Görece Hata Tablosu



Kapalı Alan Veriseti Deney 11_06 1B Odometri Sonucu

B. Kesin Pozitif Matris ile Yarı-Kesin Pozitif Matris Toplamının Daima Pozitif Matris Olduğunun Kanıtlanması

Problemin Kurulumu

Aşağıdaki şartları sağlayan A , B ve C matrislerini varsayalım:

- $C = A + B$,
- A simetrik yarı-kesin pozitif matris,
- B simetrik kesin pozitif matris,
- C simetrik yarı-kesin pozitif matris.

Amacımız bu şartları sağlayan A ve B matrislerini bulmaktır.

Özdeğer Ayrışımı

Simetrik A , B ve C matrislerini aşağıdaki gibi yazabiliriz:

$$A = Q_A \Lambda_A Q_A^T, \quad (76)$$

$$B = Q_B \Lambda_B Q_B^T, \quad (77)$$

$$C = Q_C \Lambda_C Q_C^T, \quad (78)$$

burada:

- Q_A , Q_B ve Q_C ortogonal matrislerdir ($Q_A^T Q_A = Q_B^T Q_B = Q_C^T Q_C = I$),
- $\Lambda_A = \text{diag}(\lambda_1^A, \lambda_2^A)$, $\Lambda_B = \text{diag}(\lambda_1^B, \lambda_2^B)$, $\Lambda_C = \text{diag}(\lambda_1^C, \lambda_2^C)$ özdeğer matrisleridir.

Bu özdeğerler şu şartları sağlar:

$$\lambda_1^A > 0, \quad \lambda_2^A = 0, \quad (\text{yarı-kesin pozitif}), \quad (79)$$

$$\lambda_1^B > 0, \quad \lambda_2^B > 0, \quad (\text{kesin pozitif}), \quad (80)$$

$$\lambda_1^C > 0, \quad \lambda_2^C = 0, \quad (\text{yarı-kesin pozitif}). \quad (81)$$

$C = A + B$ Eşitliğinin Hesaplanması

Daha genel bir çözüm yapabilmek için, A ve B 'nin aynı özvektör tabanını paylaşmadığını varsayalım. Hesaplamayı kolaylaştırmak için, Q_A ve Q_B şu şekilde olsun (θ 'nın 0 olması durumunda aynı özvektör tabanını elde ederiz.):

$$Q_A = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}, \quad Q_B = I. \quad (82)$$

Bu durumda:

$$A = Q_A \Lambda_A Q_A^T = \begin{bmatrix} x \cos^2 \theta & x \cos \theta \sin \theta \\ x \cos \theta \sin \theta & x \sin^2 \theta \end{bmatrix}, \quad \Lambda_A = \begin{bmatrix} x & 0 \\ 0 & 0 \end{bmatrix}, \quad x > 0, \quad (83)$$

ve

$$B = \Lambda_B = \begin{bmatrix} y & 0 \\ 0 & z \end{bmatrix}, \quad y > 0, \quad z > 0. \quad (84)$$

C matrisi şu şekilde olur:

$$C = A + B = \begin{bmatrix} x \cos^2 \theta + y & x \cos \theta \sin \theta \\ x \cos \theta \sin \theta & x \sin^2 \theta + z \end{bmatrix}. \quad (85)$$

C 'nin Özdeğerleri

C 'nin özdeğerlerini bulmak için aşağıdaki denklemi çözeriz:

$$\det(C - \lambda I) = 0. \quad (86)$$

$C - \lambda I$ 'yi yerine koyarsak:

$$C - \lambda I = \begin{bmatrix} x \cos^2 \theta + y - \lambda & x \cos \theta \sin \theta \\ x \cos \theta \sin \theta & x \sin^2 \theta + z - \lambda \end{bmatrix}. \quad (87)$$

Determinant şu şekilde olur:

$$\det(C - \lambda I) = (x \cos^2 \theta + y - \lambda) (x \sin^2 \theta + z - \lambda) - (x \cos \theta \sin \theta)^2. \quad (88)$$

0 Özdeğer Durumu

$\lambda = 0$ durumunda Denklem (88) şu hale gelir:

$$(x \cos^2 \theta + y) (x \sin^2 \theta + z) - (x \cos \theta \sin \theta)^2 = 0. \quad (89)$$

Terimleri genişletirsek:

$$(x \cos^2 \theta + y) (x \sin^2 \theta + z) = x^2 \cos^2 \theta \sin^2 \theta + xz \cos^2 \theta + yx \sin^2 \theta + yz, \quad (90)$$

$$(x \cos \theta \sin \theta)^2 = x^2 \cos^2 \theta \sin^2 \theta. \quad (91)$$

Bu ifadeleri çıkarırsak:

$$xz \cos^2 \theta + yx \sin^2 \theta + yz = 0. \quad (92)$$

Denklemin Analizi

Denklemin (92)'deki ilk iki terim $x, y, z > 0$ ve $\cos^2 \theta, \sin^2 \theta \geq 0$ şartları sağlandığında negatif olmayan bir değer alacaktır. Üçüncü terim ise aynı şartlar altında daima pozitiftir. Bu nedenle, toplamları daima pozitiftir:

$$xz \cos^2 \theta + yx \sin^2 \theta + yz > 0. \quad (93)$$

Bu durum, denklemin sağlanamayacağı anlamına gelir.

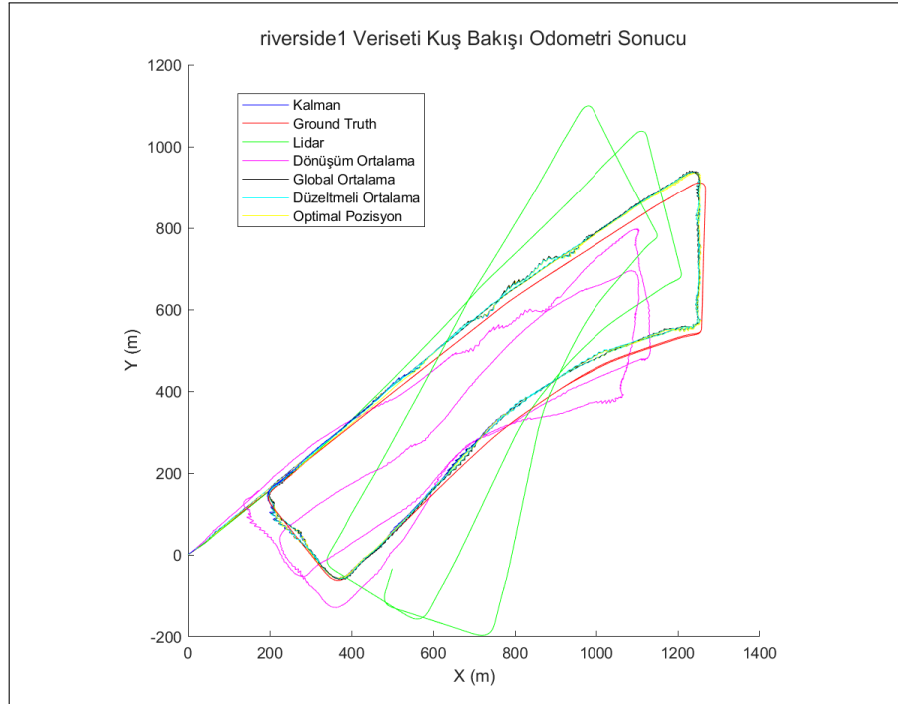
C. 3B Sensör Füzyonu Sonuçları

	RMSE	Ort.	Medyan	Std.	Min	Max
Düzeltilmiş Ortalama	6.24	5.15	4.3	3.52	0.25	30.0
LiDAR	330.58	246.28	187.94	220.53	1.8	738.87
Optimal Ortalama	19.07	16.69	19.35	9.22	0.0	38.66
Dönüşüm Ortalama	262.18	218.94	140.1	144.22	4.56	493.65
Global Ortalama	6.34	5.24	4.34	3.57	0.17	28.93
Kalman Filtresi	5.88	4.74	3.7	3.48	0.19	29.73

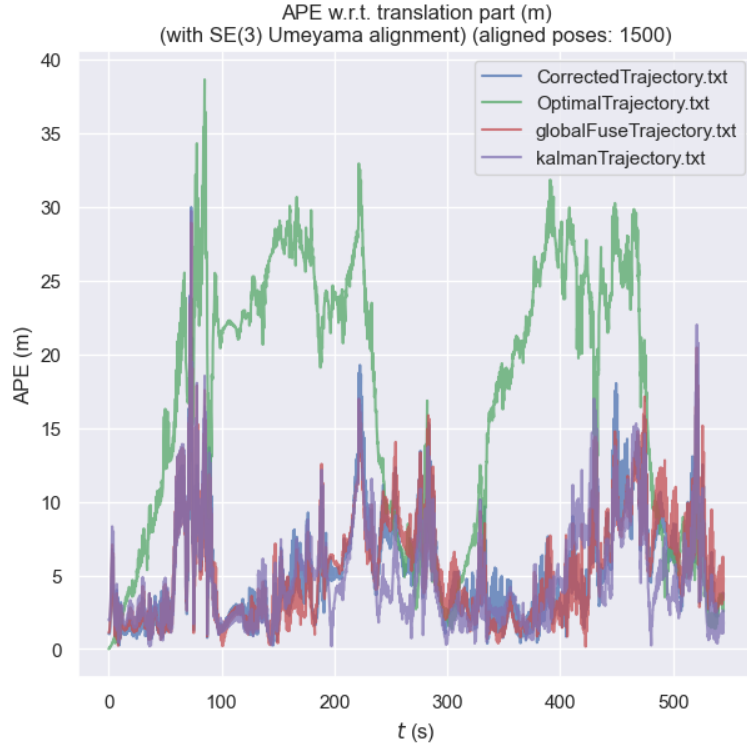
riverside1 Veriseti için Mutlak Hata Tablosu

	RMSE	Ort.	Medyan	Std.	Min	Max
Düzeltilmiş Ortalama	1.25	0.6	0.26	1.1	0.0	16.94
LiDAR	0.05	0.03	0.02	0.04	0.0	0.72
Optimal Ortalama	0.93	0.4	0.16	0.84	0.0	10.86
Dönüşüm Ortalama	0.98	0.47	0.25	0.85	0.0	15.13
Global Ortalama	1.09	0.5	0.19	0.97	0.0	13.18
Kalman Filtresi	1.0	0.44	0.18	0.9	0.0	16.11

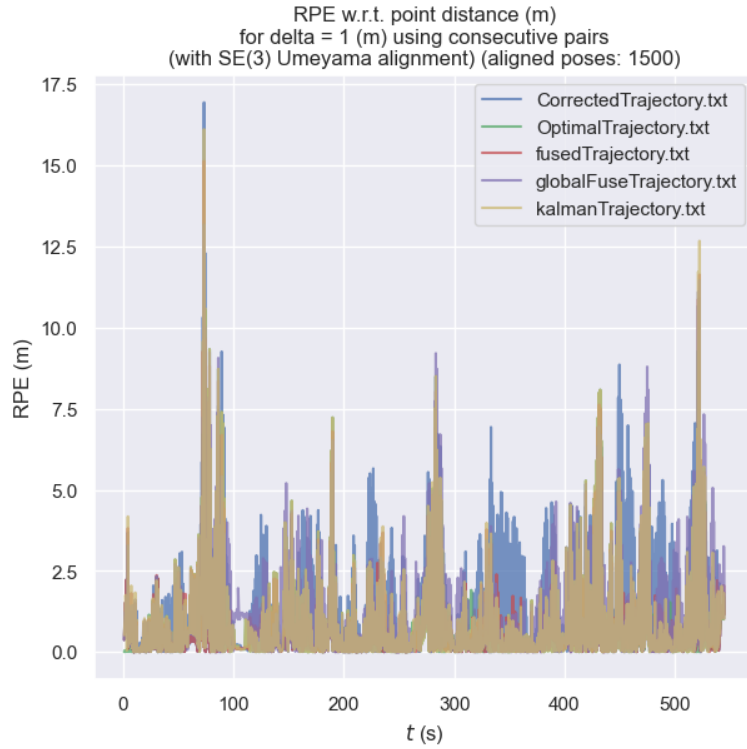
riverside1 Veriseti için Görece Hata Tablosu



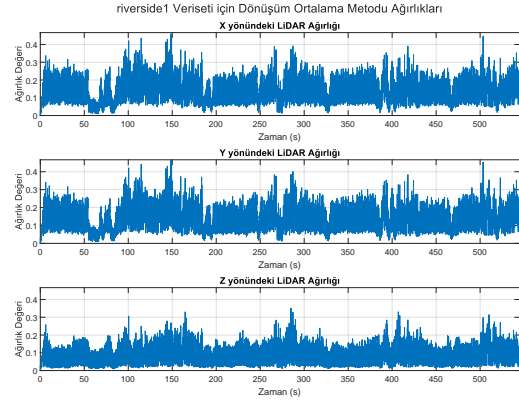
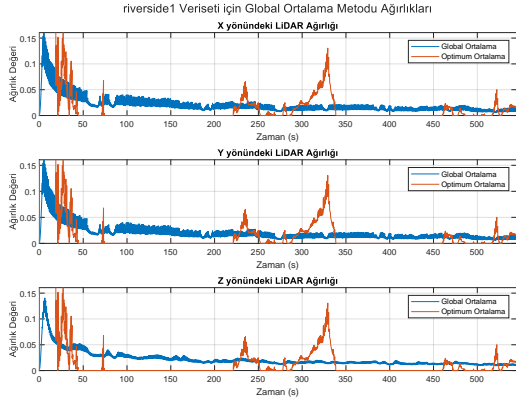
Riverside 1 Veriseti 3B Odometri Sonucu ve Sonucun Kuşbakışı Görüntüsü



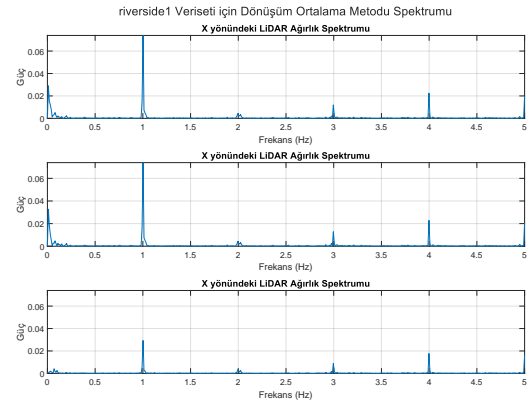
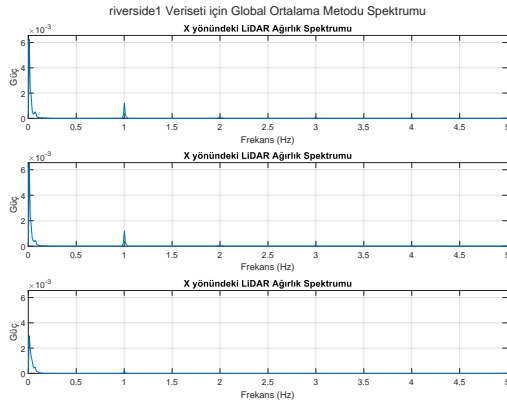
Riverside 1 Veriseti için Mutlak Pozisyon Hatası Sonuçları



Riverside 1 Veriseti için Görece Pozisyon Hatası Sonuçları



Riverside 1 Veriseti için Global ve Dönüşüm Ortalama Metotlarında kullanılan LiDAR Sensörü Ağırlıkları



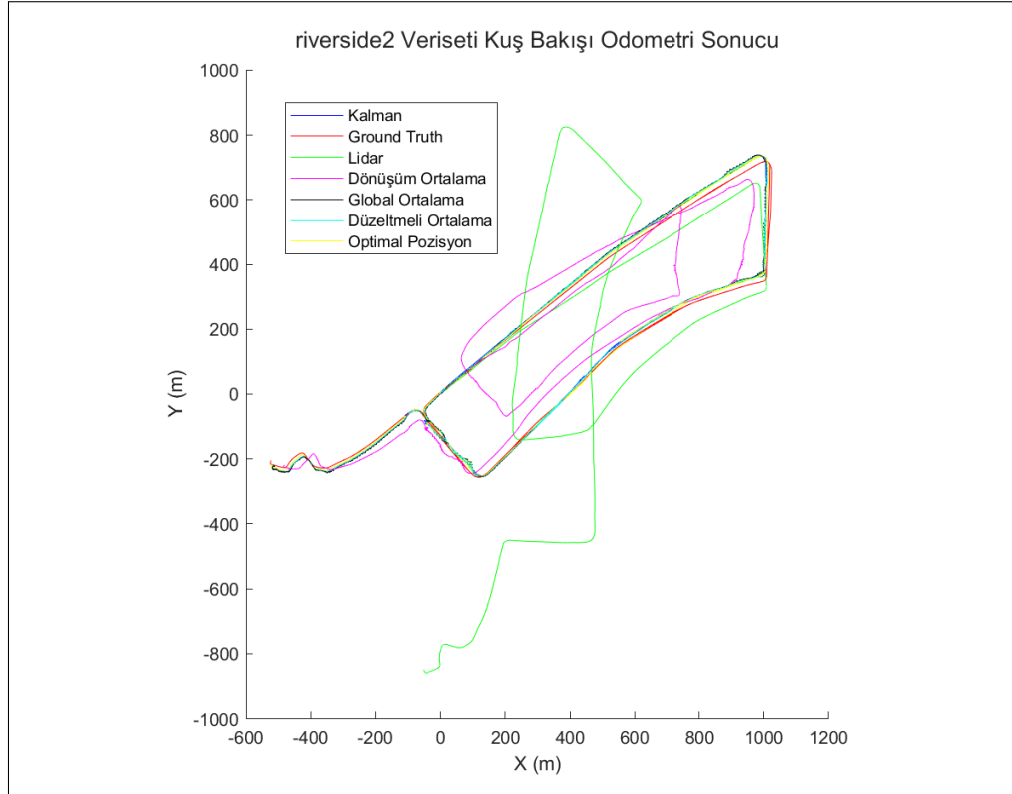
Riverside 1 Veriseti için Global ve Dönüşüm Ortalama Metotlarında kullanılan LiDAR Sensörü Ağırlıklarının Spektrumu

	RMSE	Ort.	Medyan	Std.	Min	Max
Düzeltilmiş Ortalama	5.7	4.93	5.03	2.87	0.13	32.87
LiDAR	380.55	274.96	190.7	263.09	8.61	846.19
Optimal Ortalama	11.57	9.71	7.89	6.28	0.0	34.83
Dönüşüm Ortalama	131.82	119.23	107.85	56.23	28.42	321.65
Global Ortalama	6.35	5.38	5.64	3.36	0.07	35.21
Kalman Filtresi	4.68	3.85	3.51	2.66	0.11	32.83

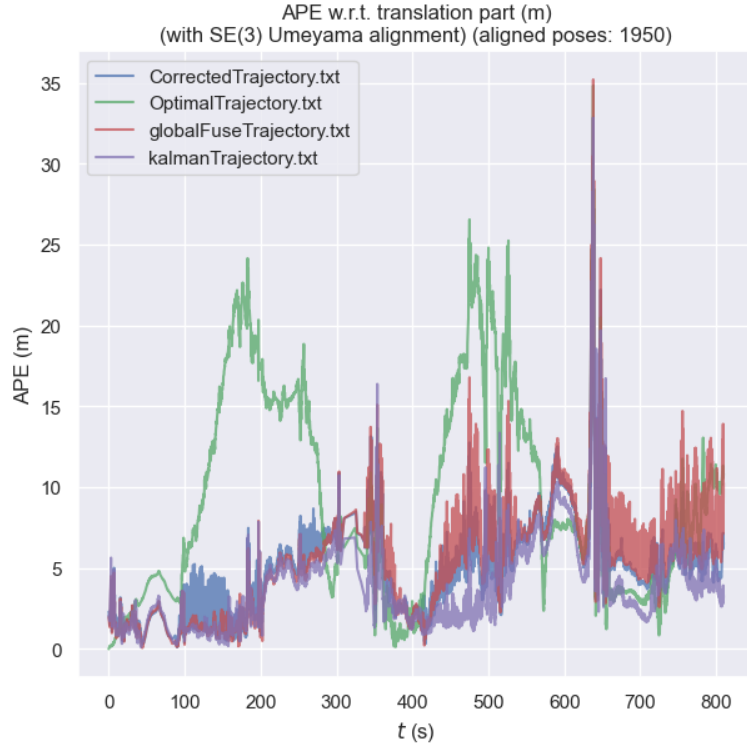
riverside2 Veriseti için Mutlak Hata Tablosu

	RMSE	Ort.	Medyan	Std.	Min	Max
Düzeltilmiş Ortalama	1.13	0.63	0.31	0.95	0.0	12.4
LiDAR	3.57	0.32	0.03	3.55	0.0	201.97
Optimal Ortalama	0.75	0.29	0.11	0.69	0.0	13.89
Dönüşüm Ortalama	3.53	0.57	0.28	3.49	0.0	195.9
Global Ortalama	1.2	0.55	0.19	1.06	0.0	14.4
Kalman Filtresi	0.85	0.34	0.12	0.78	0.0	13.89

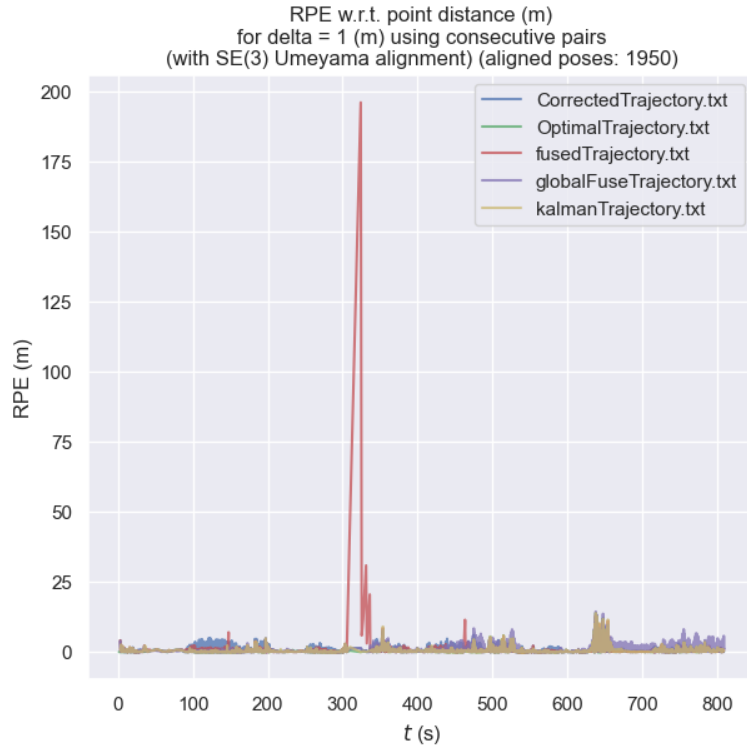
riverside2 Veriseti için Görece Hata Tablosu



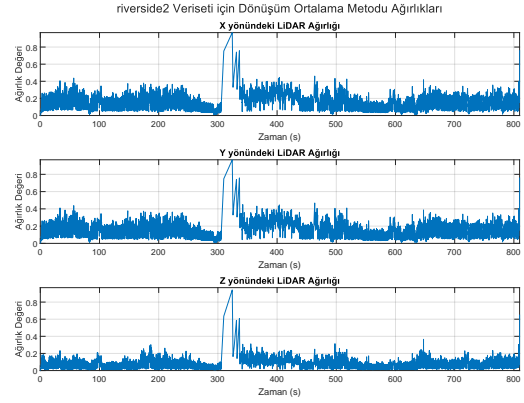
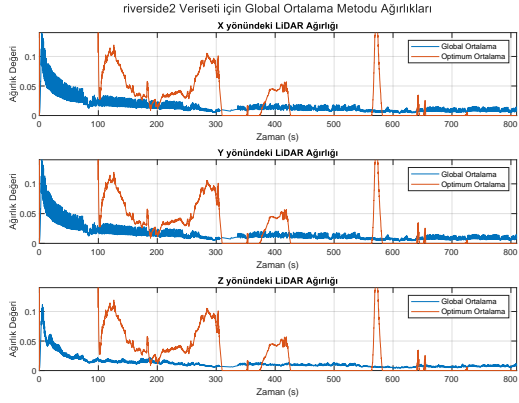
Riverside 2 Veriseti 3B Odometri Sonucu ve Sonucun Kuşbakışı Görüntüsü



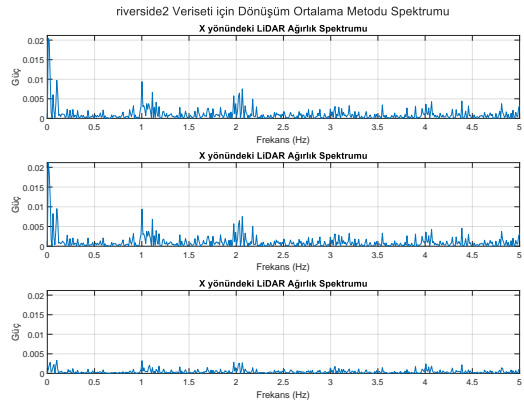
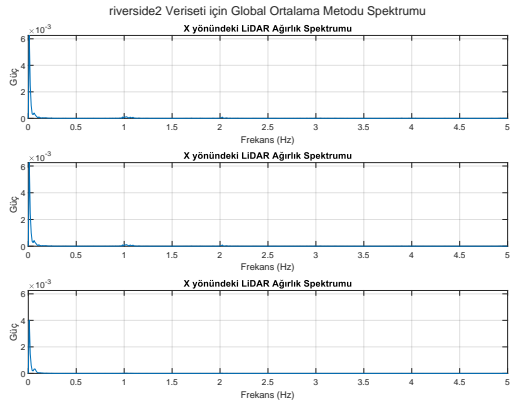
Riverside 2 Veriseti için Mutlak Pozisyon Hatası Sonuçları



Riverside 2 Veriseti için Görece Pozisyon Hatası Sonuçları



Riverside 2 Veriseti için Global ve Dönüşüm Ortalama Metotlarında kullanılan LiDAR Sensörü Ağırlıkları



Riverside 2 Veriseti için Global ve Dönüşüm Ortalama Metotlarında kullanılan LiDAR Sensörü Ağırlıklarının Spektrumu