



Tekstil Sektörü için ROS Destekli Otonom Mobil Robot Geliştirilmesi

Eylül Özer^{1*}, Burak İner²

¹ Birfen Elektrik Elektronik San. ve Tic. Ltd. Şti., Yalova, Türkiye (ORCID: 0000-0002-7468-5810), eylul.ozer@birfenelektrik.com.tr

² Kocaeli Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Kocaeli, Türkiye (ORCID: 0000-0003-0933-654X), binner@kocaeli.edu.tr

(3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications June 11-13, 2021)

(DOI: 10.31590/ejosat.961670)

ATIF/REFERENCE: Özer, E. & İner, B. (2021). Tekstil Sektörü için ROS Destekli Otonom Mobil Robot Geliştirilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, (26), 482-487.

Öz

Tekstil sektöründe endüstriyel fabrikalardaki ürün yerden yükseltilmiş kutu içinde taşınır. Maksimum 300 kg ağırlıkta olan bu kutuların, fabrika ortamında otonom bir şekilde taşınması için yükü üzerine alacak şekilde tasarlanmış bir robot mekaniğine ihtiyaç vardır. Tekstil sektörü için tasarlanan endüstriyel mobil robotlarda hedefe hassas yaklaşım yapılmadan önce hedefe minimum hata ile yaklaşma önemli bir konudur. Yaklaşma hataları sebepleri fiziksel ve/veya yazılımsal olabilir. Bu çalışma, tekstil sektöründe bitmiş ürün ve malzemeleri fabrika ortamında taşımak için kullanılacak doğal navigasyon ile yönlendirilen diferansiyel sürüş tipli AMR'nin (Otonom Mobil Araç) geliştirmeye ve robotun hedefe yaklaşma hatalarının tespitine yönelik metodu içerir. Yaklaşma hata miktarı sebeplerini, simülasyon ortamında ve gerçek ortamda karşılaştırılarak bulmak için çalışma metodu önermektedir. Bu sebepler arasında, testlerin yapıldığı zeminde gözle görülmeyen farklılıklar, mekanik imalatta meydana gelebilecek hatalar, global ve local planner parametreleri bulunmaktadır. Çalışma, Linux üzerinde ROS (Robot İşletim Sistemi) ile gerçekleştirilmiştir. İlk olarak, Gazebo sanal dünya, gerçek ortam ölçüleri dae formatında alınarak config ve sdf dosyaları ile oluşturulmuştur. ROS kullanılarak, Eş Zamanlı Konum Belirleme ve Haritalama (SLAM) gerçekleştirilmiştir. Safety Laser Scanner sensörü kullanılarak için her iki ortam için harita oluşturulmuştur. Haritalama metodu için , gmapping seçilmiş ve iki ortam için ayrı haritalar oluşturulmuştur. Lokalizasyon için amel (Adaptive Monte Carlo Localization) kullanılmıştır. Robotun aynı konumdaki hedeflere sonsuz bir döngü içerisinde gitmesi sağlanmıştır. Global ve local planner parametreleri değiştirilerek testler tekrarlanmıştır. Tüm uygulamalar başarıyla gerçekleşmiştir. Hedefe yaklaşma hataları /odom konusuna abone olarak gelmesi gereken nokta ve geldiği nokta arasındaki fark ölçülerek hesaplanmıştır. Her iki dünya için ayrı hesaplamalar yapılarak sonuçlar karşılaştırılmıştır. Sonuç olarak, simülasyon değerlerinin daha başarılı olduğu tespit edilmiştir. Bu sayede mekanik tasarım üzerinde tahmin edilen hatalar üzerinde durulmuştur. Odometri hesaplamasında kullanılan merkez ile robotun ağırlık merkezinin aynı ekseninde olmamasından kaynaklandığı tespit edilmiştir.

Anahtar Kelimeler: Otonom sistem, Mobil robot, Gazebo, ROS

Tekstil Sektörü için Otonom Mobil Robot Sistem Değerlendirilmesi

Abstract

In the textile sector, the product in industrial factories is transported in an elevated box from the ground. A robot mechanic designed to take the load is needed for these boxes, which have a maximum weight of 300 kg, to be transported autonomously in the factory environment. In industrial mobile robots designed for the textile industry, it is important to approach the target with minimum error before making a precise approach to the target. Causes of approach errors can be physical and/or software. This study includes the method for the development of the differential driving type AMR (Autonomous Mobile Robot) guided by natural navigation, which will be used to transport the finished products and materials in the factory environment in the textile industry, and the detection of the approach errors of the robot to the target. It proposes a working method to find the causes of the approximation error amount by comparing them in the simulation environment and the real environment. Among these reasons, there are invisible differences in the ground where the tests are made, errors that may occur in mechanical manufacturing, global and local planner parameters. The study was carried out with ROS (Robot Operating System) on Linux. First, Gazebo virtual world was created with real environment dimensions in dae format with config and sdf files. Simultaneous Localization and Mapping (SLAM) was performed using ROS. A map was created for both environments using the Safety Laser Scanner sensor. For the mapping method, gmapping was chosen and separate maps were created for the two environments. Amcl (Adaptive Monte Carlo Localization) was used for localization. It is ensured

* Sorumlu Yazar: eylul.ozer@birfenelektrik.com.tr

that the robot goes to the targets at the same location in an endless loop. The tests were repeated by changing the global and local planner parameters. All applications have been carried out successfully. Errors in approaching the target were calculated by measuring the difference between the point that should be reached by subscribing to the topic /odom and the point where it came from. Separate calculations were made for both worlds and the results were compared. As a result, it has been determined that the simulation values are more successful. In this way, the estimated errors on the mechanical design are emphasized. It has been determined that the center used in the odometry calculation and the center of gravity of the robot are not on the same axis.

Keywords: Autonomous system, mobile robot, Gazebo, Ros

1. Giriş

Otonom Mobil Robotlar dördüncü sanayi devriminin getirdiği sensör ve işlem gücünü yapay zekâ ile birleştirerek otomatik yönlendirmeli araçlardan çok daha esnek, çok daha verimli ve çok daha entegre bir çalışma yöntemi sunmaktadır. Çalışacakları alanın doğal navigasyon (natural navigation) olarak adlandırılan yöntemle önceden haritalama işlemini yaptıkları için bant, şerit, kablo ya da ray gibi hiçbir altyapı yatırımına gerek duymamaktadır. Ayrıca doğal navigasyon fabrika ortamında yapılacak değişikliklere de kolayca adapte olabilmektedir. Böylece faaliyet gösterdikleri tesislerde maliyetlerin düşürülmesinde ve verimlilik artışında önemli bir role sahiptir.

Literatürde robot tasarımı ve ROS desteğinin gerçekleştirilmesi ile ilgili çalışmalar bulunmaktadır. Hem donanım mimarisini hem de elektronik iletişim protokolleri dikkate alınarak ROS destekli diferansiyel sürüş sistemine sahip, tek kart bilgisayar odaklı iç mekanlar için otonom bir mobil robot geliştirilmiştir. 2 boyutlu iç mekanlar için robotun verilen görevleri yerine getirdiği ifade edilmiştir fakat hata miktarlarından bahsedilmemiştir[1].

Gazebo simülasyon ortamında diferansiyel sürüş tekniğine sahip otonom bir mobil robot gmapping SLAM algoritması kullanılarak haritalama işlemi gerçekleştirilmiş ve robotun verilen görev noktalarına ulaşabilmesi için geçen zamanı hesaplamışlardır. Bu çalışma gerçek bir robot üzerinde denemesi yapılmayan sadece simülasyon ortamında bir çalışma olarak yapılmıştır[2].

Diğer bir çalışmada[3] ise hem Gazebo simülasyon ortamında hem de hazır bir robotik platform olan Turtlebot 3 mobil robotunda yerel planlayıcıların performansları robotun konumu ve oryantasyonu ile seyahat süresi için doğruluk ve tekrarlanabilirlik metriklerine göre karşılaştırılmıştır. Test edilen tüm planlayıcıların gereksinimleri karşıladığı ifade edilmiştir. Bu çalışmada ise araştırma geliştirme için kullanılan Turtlebot 3 robotu kullanılmış, özelleştirilmiş bir mobil robot tasarımı, imalatı yapılmadığından tasarım ve imalata özel hatalar tespit edilememiştir.

Bu çalışmada, tekstil fabrikalarındaki ortam koşullarına uyum sağlayacak şekilde, maksimum 300 kilogram yükü üzerine alarak taşıyan diferansiyel sürüş sistemine sahip otonom mobil robot mekaniği tasarlanmıştır ve imalatı gerçekleştirilmiştir. Lokalizasyon, haritalama ve navigasyon işlemlerini gerçekleştirilebilmek için ROS paketi geliştirilmiştir. Alt seviye donanımlar için ROS'un sunduğu fonksiyonların içerikleri donanıma özel olduğundan bulunmamaktadır. ROS sadece arayüz fonksiyonlarının başlıklarını sunar. Bu donanımların kontrolü için sunulan arayüzlerin içeriğinin doldurulması gerekmektedir. Kullanılan donanımlara özel iletişim protokolleri, donanımların haberleşme hızları gözönüne alınarak alt seviye fonksiyonlar yazılmıştır.

Gerçek robot üzerinde çalışmalar yapılmadan önce simülasyon ortamı olarak Gazebo'da çalışmalar yürütülmüştür.

Gazebo, donanımlar arasındaki etkileşimi simüle etmek için kullanılmıştır. Gazebo ve ROS'un beraber çalıştırılabilmesi için robotun tüm mekanik ve donanım unsurları xml tabanlı URDF (Universal Robot Description Format) dosyalarında oluşturulmuştur. Bu dosyalar ile birlikte Gazebo için yapılandırma dosyaları oluşturularak ortam modeli sanal ortamda çalıştırılmıştır. Gazebo, sensör gürültüsünü modellediğinden sanal ortamda gerçeğe yakın bir ortam sunmaktadır.

ROS, aynı düğümün sanal ve gerçek dünyada çalıştırılmasına olanak sağlar. Yaklaşma hatalarının tespiti için, verilen 3 nokta arasında toplam 5 döngüde gerçekleştirilen hedefe gitme işlemi gerçekleştirilmiştir. Bu durum, iki dünyayı karşılaştırarak robotun hedefe yaklaşma toleransını neden geçmiş olabileceği konusunda bilgi vermiştir. Yapılan tüm çalışmalar başarıyla gerçekleştirilmiştir. Yapılan testler sonucunda, planlayıcı parametreleri değiştirildiğinde hedefe yaklaşma başarısının değişmediği, sanal ortamda işlemin başarı ile gerçekleştiği görülmüştür. Çalışma sonucunda mekanik imalattaki hatalar tespit edilmiştir. Fabrika içerisinde gerekli lojistiğin sağlanabilmesi için otonom mobil robotun ürüne yaklaşma başarısını etkileyen bir çok faktör bulunmaktadır. Bunlar zeminde gözle görülmeyen seviye farklılıkları, mekanik imalat hataları gibi fiziksel nedenler olabileceği gibi planlayıcı parametrelerinden de kaynaklanabilmektedir.

2. Materyal ve Metot

2.1. Donanım

Bu çalışmada, Şekil 1'de gösterilen diferansiyel sürüş sistemine sahip 300 kg yükü üzerine alarak taşıyan, tekstil fabrikalarındaki ortam koşullarına uyum sağlayacak şekilde tasarlanmış mobil robot mekaniği kullanılmıştır. Bu mekanik üzerine sistem konfigürasyonu sağlanarak Allen Bradley Micro 850 PLC, endüstriyel pc, enkoderler, Sick Microscan3 Pro safety alan tarayıcı, Realsense derinlik kamerası, Realsense takip kamerası ve motor sürücüler kullanılmıştır.



Şekil 1: Diferansiyel AMR

2.1.1. Endüstriyel PC

Endüstriyel pc'ler, tekstil fabrikalarında bulunan aşırı sıcaklık, toz, nem gibi etmenlerden, titreşim ve enerji dalgalanmalarından etkilenmeyecek şekilde tasarlanan bilgisayarlardır. I7 işlemcili, linux işletim sistemine sahip endüstriyel pc üzerine ROS işletim sistemi kurularak kameralar, safety lazer scanner gibi sensörlerden ve PLC haberleşmeleri sağlanarak manual ve otonom kontrol sağlanabilir.

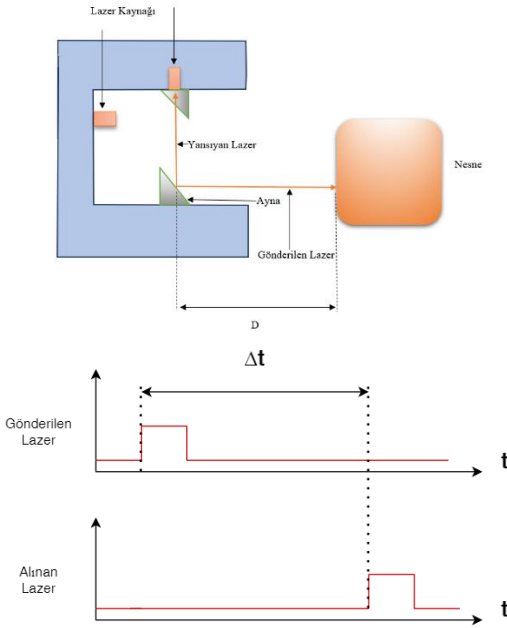
2.1.2. Allen Bradley Micro 850 PLC

Bu uygulamada, Şekil 5'te gösterilen endüstriyel ortam koşullarında çalışabilen Allen Bradley Micro 850 PLC kullanılmıştır. Micro 850 PLC, üzerinde bulunan 14 input, 10 output girişlerine bağlanan enkoder bilgilerini Modbus haberleşmesi üzerinden endüstriyel peye aktarır.

Bu çalışmada, Micro 850 PLC ve endüstriyel pc'nin modbus üzerinden haberleşmesi için C dilinde yazılmış olan libmodbus kütüphanesi ROS ortamında derlenmiştir. 16, 32 ve 64 bitlik tüm data tipleri encode ve decode işlemleri gerçekleştirilerek ROS ortamından Micro 850 PLC'ye okuma ve yazma işlemleri gerçekleştirilmiştir.

2.1.3. Sick MicroScan3 Pro Safety Laser Scanner

Laser scannerlar, yüzeylere yansıtıkları kızılötesi lazer ışınlarının, nesneye çarptıktan sonraki dönüş süresini ölçme ilkesiyle çalışırlar. Laser scanner düzenli ve çok kısa aralıklarla kızılötesi ışın yayılımı yapar. Kızılötesi lazer ışını menzildeki nesneye çarparak geri döner. Laser scanner, yansıyan ışını içerisindeki döner ayna ile alır. Laser scanner, kızılötesi ışının iletiildiği ve alındığı an ile arasındaki zaman aralığına göre nesneye olan mesafeyi hesaplar. Şekil 2'te laser scannerın basit olarak nasıl çalıştığı gösterilmektedir.



Şekil 2: Safety Laser Scanner Mesafe Ölçümü

Bu uygulamada yağlı ve tozlu ortamlara özel olarak üretilen, yağlı ve tozlu ortamlardan ve fabrikalarda oluşabilecek matkaplı kesme testeresi kıvılcımlarına dayanıklı Sick Microscan3 Pro ürünü kullanılmaktadır.

2.1.4. Realsense Deeping Camera, Realsense Tracking Camera

Intel RealSense D435i, diğer tüm Intel kameraların arasında en geniş görüş alanına sahip ve hızlı hareket eden uygulamalar için ideal olan derinlik sensörüdür. Geniş görüş alanı, sahnenin mümkün olduğunca çok görmesinin hayati derecede önemli olduğu robotik gibi uygulamalar için mükemmeldir. 10 metreye

kadar menzile sahip bu küçük form faktörlü kamera, herhangi bir çözüme kolaylıkla entegre edilebilir.

Intel Realsense T265 Tracking Kamera ise balık gözü kamerasına sahip olması ve özellik takibi (feature tracking) için SDK sunması nedeniyle bu kamera da bu çalışmada kullanılmaktadır.

2.1.5. Enkoder

Enkoder (Sinyal üretici), bir milin dönme ya da ötelenme hareketine karşılık, sayısal bir elektrik sinyali üreten elektromekanik bir mekanizmadır. Bu kodlayıcılar konum, sayım, hız veya yönü belirlemek için geri bildirim sinyali gönderir. Enkoderler, sinyal oluşturabilmek için mekanik, manyetik, optik ve dirençli sistemlerini kullanır. [4] Pozisyon ve konum belirleme uygulamalarında 2 tip enkoder kullanılır. Bunlar mutlak ve artımlı enkoderlerdir.

Bu uygulamada artımlı enkoderler kullanılmaktadır. Artımlı enkoder kullanılması sebebi Dead Reckoning algoritmasında artımlı enkoder verilerinin kullanılmasıdır. Bu çalışmada, Dead Reckoning algoritması, Diferansiyel AMR'nin mevcut konumunu hesaplama ve ardından geçen süre boyunca hız, rota yönü ve rota tahminlerini dahil etmek için kullanılmaktadır.

2.1.6. Motor Sürücü

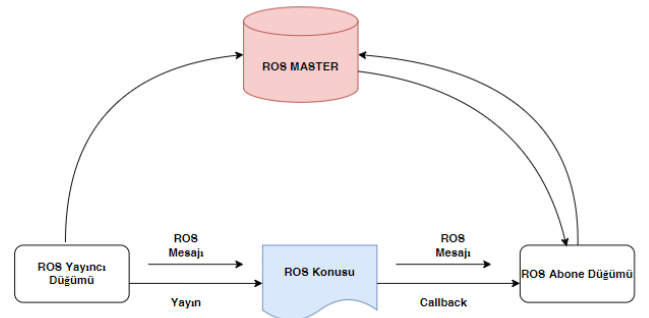
Motor sürücü, temel sabit frekanslı, sabit voltajlı sinüs dalgası gücünü, motorların hızını kontrol etmek için kullanılan değişken frekanslı, değişken voltajlı bir çıkışa dönüştürür. Bu çalışmada motorların hız bilgisi PLC tarafından motor sürücülere aktarılmaktadır.

2.2. Yazılım

Bu bölümde işletim sistemi, simülasyon ortamı ve ROS görselleştirme platformundaki görselleştirme araçlarıyla ilgili bilgiler dahil olmak üzere yazılım uygulaması açıklamıştır.

2.2.1. ROS

Şekil 3'da gösterildiği gibi, Ros işletim sistemi birbiriyle publisher/subscriber modelini kullanarak haberleşen düğümlerden oluşur. Düğümler işlem yapabilen birimlerdir.[6] Düğümlerin haberleşebilmeleri ve birbirlerinden haberdar olabilmeleri için Ros Master'a ihtiyaç vardır. Ros Master merkezi bir XML-RPC sunucusudur. İnsanların anlayabileceği dilde mesajları barındıran ağ tabanlı bir yapıdır.[7]. Düğümler Ros Mastera kaydedilir ve kayıtlı düğümler arasında haberleşme gerçekleşir. Düğümler haberleşmeyi konular(topic) yayınlayıp bu konulara abone olarak gerçekleştirir.[8] Ayrıca bir düğüm birden fazla konuya abone olup birden fazla konu yayınlatabilir.



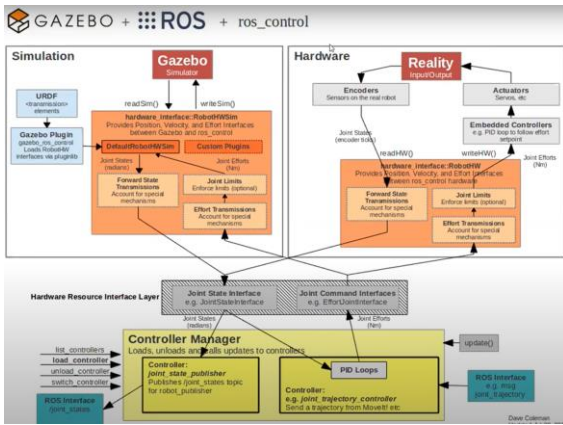
Şekil 3: ROS Çalışma Prensipleri

2.2.2. Gazebo

Gazebo, dış mekanlarda ve iç mekanlarda çalışan çoklu robotlar için geliştirilmiş bir üç boyutlu simülasyon programıdır. Birçok sensörü, nesneyi ve nesne ile robot arasındaki etkileşimi simüle etme yeteneğine sahiptir. Server-client mimarisi ile çalışan bir programdır. Server(Gzserver) somut işlevleri yaparken Client(Gzclient) kullanıcının isteklerinin yerine getirilmesi ve simülasyonu görsele aktarılmasını sağlar. Gazebo programının yüklenmesiyle beraber veritabanından birçok model, robot ve nesneye ulaşmamıza imkân sağlar. Aynı zamanda sadece veritabanındaki robotlarla sınırlı kalmayıp, kendi robot ve dünyamızı yaratma imkânı bize tanınmıştır. Robotların Gazebo simülasyon ortamı ve Ros ile birlikte çalışabilmesi için Urdf dosyasına ve gazebo yapılandırma dosyalarına ihtiyacı vardır. Bu yapılandırma dosyaları ile gerçeğe yakın bir simülasyon ortamı elde edilir. Ros gazeboya entegre edilmiştir ve gazebo simülasyon ortamıyla tam uyumlu çalışmaktadır. Bu sayede projenin prototip aşamasında ROS paketi haline getirilen geliştirilen veri yapısını test etmek için Gazebo simülasyon ortamı hızlı bir geliştirme için kullanılacaktır [5].

2.2.3. ROS Control

ROS Control, farklı eyleyicilere (aktüatör) göre robotların ROS ile etkileşime girebilmesi için geliştirilmiş bir API'dir. APIlere erişim sağlayan ve kaynak çalışmalarını ele alan basit bir yönetici arayüzü; eklemlerin donanım sınırlamasını bilen ve aktüatörlere gönderilen komutların limitleri arasında olmasını sağlayan bir güvenlik arabirimi gibi özellikleri sunabilmektedir. Ros Control tek başına donanım kontrolünün nasıl uygulanacağını (örneğin motorlara akım gönderme) veya donanım durumunun nasıl okunacağını bilmediğinden, Ros Control yönergelerine göre kod yazılarak donanımın kontrol edilmesi gerekmektedir. Ayrıca, pozisyon, hız ve yörünge gibi durumların kontrol edilmesi için Ros Control'e robotun ve robot eklemlerinin mevcut durum bilgilerinin sağlanması gerekir. Şekil 4'de simülasyon ortamında ve gerçek ortamda, simülasyon donanım, kontrolörler ve aktarımlar arasındaki ilişki gösterilmiştir.



Şekil 4: Ros Control Prensipleri [7]

2.2.4. Rviz

Rviz, Ros uygulamaları için bir 3D görselleştirme ortamıdır. Robot modelinin görünümünü sağlamanın yanı sıra sensör bilgilerinin kaydedilmesi ve görselleştirilmesi, geliştirme ve hata ayıklama için önemli bir araçtır. Robotun ne gördüğünü ve ne yaptığını görmemizi sağlar. Resimler ve point cloud verileri dahil olmak üzere kameradan, lidarlardan, 3D ve 2D cihazlardan gelen verileri görüntüleyebilir. Ayrıca Rviz navigation stack ile beraber kullanılabilir. Rviz gui araçları ile robotun başlangıç pozisyonu

belirlenebilir, aynı zamanda harita üzerinden belirlenen bir hedef noktasına hareket etmesi sağlanabilir. Navigasyon sırasında robot tarafından yapılan yol planı da rviz üzerinden görüntülenebilir.

3. Araştırma Sonuçları ve Tartışma

3.1. Gazebo Sanal Ortamın Oluşturulması

Bu çalışmada, diferansiyel AMR üzerinde kullanılan safety laser scanner, kameralar, tekerlekler ve enkoderler ile robot arasındaki etkileşimi simüle etmek için üç boyutlu simülasyon programı Gazebo kullanılmıştır. Gazebo sensör gürültüsünü modellediğinden gerçeğe yakın bir ortam yaratır. Gerçek ortamın birebir ölçüler ile oluşturulan modelinin dae formatı alınarak model.sdf ve model.config dosyaları oluşturulmuştur. Ortamın model dosyasının yanı sıra ortamdaki masa vb. gibi statik nesnelerin de dae dosyası model.sdf ve model.config dosyaları oluşturularak Gazebo ortamında gerçek konumlarına eklenmiştir. Oluşturulan bu dünya hatasız bir şekilde Gazebo ortamında açılmaktadır.

Gazebo'nun ROS ile beraber çalışabilmesi için robotun tüm unsurlarının açıklandığı XML tabanlı bir URDF (Universal Robot Description Format) dosyası ve Gazebo için yapılandırma dosyaları oluşturulmuştur.

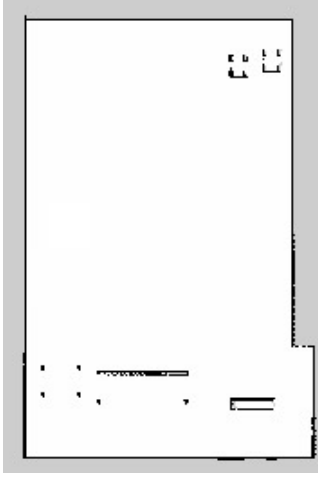
Diferansiyel AMR'nin mekanik tasarım bilgilerine göre kaldırma sistemi, ana tekerlekler, avare tekerlekler, kameralar ve safety laser scanner robot üzerindeki konumlarına göre URDF dosyaları oluşturulmuştur. URDF dosyası oluşturulurken robot ile ilişkili ekipmanların eklem, bağlantı yerleri doğru bir biçimde konumlandırılması gerekmektedir.

Donanım arabirimleri, ROS Control tarafından donanım komut göndermek ve almak için ROS controller'lerden biriyle bağlantılı olarak çalışmaktadır. Komut gönderilecek her eklem için eklem türüne göre controller için yaml dosyaları yazılmıştır. Yaml dosyaları içerisinde oluşturulan controllerlar sayesinde prismatic, revolute gibi eklemler için eklem limitleri, continuous gibi eklemler için hız limitleri vb., robotun maksimum ve minimum hızı ve diferansiyel sürüş tekniğine özel veriler parametre olarak alınarak Diferansiyel AMR sanal ortamda hareket etmeye hazır hale getirilmiştir.

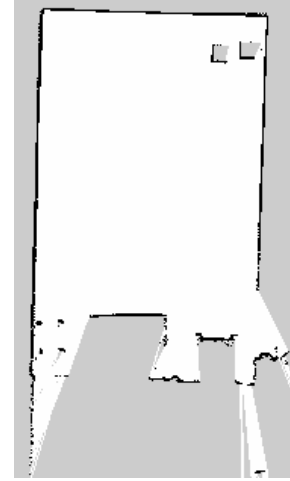
3.2. Gerçek ve Sanal Ortamda Otonom Sürüş

Gerçek ve sanal ortamda otonom sürüş gerçekleştirmek için her iki ortamda da harita çıkarılmıştır. Sanal ortamda çıkarılan harika Şekil 5'te ve gerçek ortamda çıkarılan harita Şekil 6'da gösterilmiştir. Gerçek ortamda hedef noktalar belirlenerek zemine işaretleme yapılmıştır. Gerçek ve sanal ortam sahneleri Şekil 6'da gösterilmiştir. Aynı işaretlemeler, işaretlerin birbirlerine ve engellere olan mesafeleri hesaplanarak sanal ortamda zemin olarak eklenmiştir.

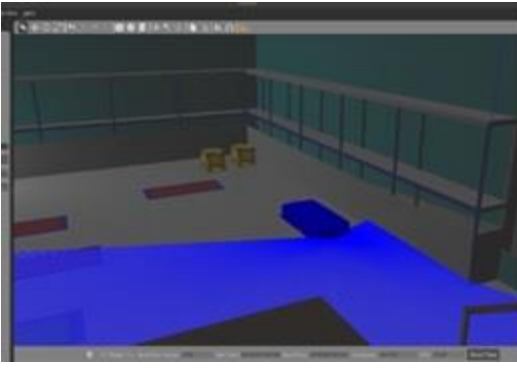
Diferansiyel AMR'nin belirlenen noktalar arası gezmesi için düğüm oluşturulmuştur. Oluşturulan bu düğüm 10 cm toleranslı olarak robotun girilen hedeflere sırasıyla gitmesini, konuma vardığında 10 saniye beklemesini sağlamaktadır. Sonsuz döngü içerisinde robotu gezdirerek hedefe her ulaştığında sapmaları tespit etmek, sapmaların sebebinin mekanik tasarımdan, zeminden ve/veya config dosyalarındaki parametrelerden kaynaklanıp kaynaklanmadığını tespit etmek amaçlanmıştır. Her iki dünyada da aynı düğüm çalıştırılarak uzun süre çalıştırma testleri yapılmıştır.



Şekil 5: Sanal Ortamda Çıkarılan Harita



Şekil 6: Gerçek Ortamda Çıkarılan Harita



Şekil 7: Sanal ve Gerçek Ortamlarda Hedefe Yaklaşma Karşılaştırılması

4. Sonuç

Gerçek ve sanal ortamda çalıştırılan aynı ROS düğümü ile 3 nokta arası gezinme işlemi gerçekleştirilmiştir. Tablo-1’de gerçek ortamda ve sanal ortamda hedefe yaklaşma hataları gösterilmiştir.

Tablo 1. Sanal ve Gerçek Ortamda Yapılan Teste Göre Yaklaşım Hata Miktarları

		1.Hedef Hata (cm)	2.Hedef Hata (cm)	3.Hedef Hata (cm)
GERÇEK ORTAM	1.Tur	1	6	9
	2.Tur	0	6	9
	3.Tur	1	9	11
	4.Tur	0	8	12
	5.Tur	1	11	13
SANAL ORTAM	1.Tur	2	3	3
	2.Tur	0	2	5
	3.Tur	0	2	9
	4.Tur	0	5	11
	5.Tur	7	10	11

Mekanik imalattan kaynaklı sorunlar tespit edilmiştir. Elde edilen bu bilgiler doğrultusunda, max vel theta parametre değeri düşürüldüğünde ve config dosyası içerisinde inflation, yaw tolerance & xy tolerance parametrelerinin değerleri

değiştirildiğinde yaklaşma hataları bir miktar düzeldiği görülmüştür. Bu düzelmeye rağmen hata miktarları Tablo-1’de gösterildiği gibi gerçek ortamdaki hata miktarları simülasyon ortamındaki hata miktarlarından daha fazladır. Bu sorunun odometri hesaplamasında kullanılan merkez ile robotun ağırlık merkezinin aynı eksende olmamasından kaynaklandığı tespit edilmiştir. Buna ek olarak, soruna gerçek ortamdaki zemindeki seviye farklılıklarının da sebep olduğu düşünülmektedir.

Kaynakça

- [1] Köseoğlu, M., Çelik, O. M., & Pektaş, Ö. (2017). Design of an autonomous mobile robot based on ROS. IDAP 2017 - International Artificial Intelligence and Data Processing Symposium. <https://doi.org/10.1109/IDAP.2017.8090199>
- [2] R. Kannan Megalingam, C. Ravi Teja, S. Sreekanth, and A. Raj, “ROS based Autonomous Indoor Navigation Simulation Using SLAM Algorithm,” Int. J. Pure Appl. Math., vol. 118, no. 7, pp. 199–205, 2018.
- [3] Cybulski, B., Wegierska, A., & Granosik, G. (2019). Accuracy comparison of navigation local planners on ROS-based mobile robot. 12th International Workshop on Robot Motion and Control, RoMoCo 2019 - Workshop Proceedings, 104–111. <https://doi.org/10.1109/RoMoCo.2019.8787346>
- [4] <https://www.makyapelektronik.com/blog/encoder-nedir-kullanim-alanlari-nelerdir> Ziyaret Tarihi: 16.05.2021

- [5] <http://yapbenzet.kocaeli.edu.tr/ros-nedir-robot-operating-system/> Ziyaret Tarihi:20.05.2021
- [6] <https://devnot.com/2020/ros-robot-operating-system-nedir> Ziyaret Tarihi:20.05.2021
- [7] <https://ozguradem.net/turkish/coding/2015/03/10/ros-robot-operation-system-nedir/#:~:text=ROS%20Master%2C,da%20di%C4%9Fer%20keyfi%20verileri%20kaydeder.> Ziyaret Tarihi:21.05.2021
- [8] https://raw.githubusercontent.com/osrf/gazebo_tutorials/master/ros_control/Gazebo_ros_transmission.png Ziyaret Tarihi:21.05.2021