JOURNAL of
MATERIALS and MECHATRONICS:A

e-ISSN 2717-8811
http://www.journalmm.com JournalMM, 2020, 1(1), 29-41

Araştırma Makalesi / Research Article

ROS/Gazebo Ortamında Tank Sürüş Özellikli Mobil Bir Robotun Simülasyonu

Yavuz Bahadır KOCA[1]*, Barış GÖKÇE[2], Yılmaz ASLAN[3]

[1]Afyon Kocatepe Üniversitesi, Afyon Meslek Yüksekokulu, Elektronik ve Otomasyon Bölümü, Afyonkarahisar, Türkiye,
ORCID ID: https://orcid.org/0000-0002-0317-1417
[2]Necmettin Erbakan Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü, Konya, Türkiye,
ORCID ID: https://orcid.org/0000-0001-6141-7625
[3]Kütahya Dumlupınar Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Kütahya, Türkiye,
ORCID ID: https://orcid.org/0000-0001-9798-1519

Geliş/ Recieved: 21.05.2020; **Kabul / Accepted:** 25.06.2020

ÖZET: Robot simülatörleri, teorik bulguların performansını gerçek robotlara uygulamadan önce tahmin etme olanağı sağlar. Bu çalışmada, mobil bir robotta sistem performans analizini amaçlayan Gazebo simülatörü ile Robot İşletim Sistemi (ROS) kullanılarak, tank sürüşü özellikli bir mobil robot için 3 boyutlu modelleme ve simülasyon sunulmaktadır. Bu kapsamda holonomik olmayan tank sürüş özellikli mobil bir robot tasarlanmış, tüm fiziksel ve atalet özellikleri tanımlanmış ve simüle edilmiştir. Gazebo, Birleşik Robot Açıklama Formatı (URDF) ve parametrelendirilmiş robot bileşen makro (XACRO) dosyası aracılığıyla robot dünya ortamının, fiziksel modelin, algılayıcıların ve kontrol sisteminin simülasyonunu sağlar. Gerçekleştirilen bu simülasyonla robotta gerçek sistem uygulamasından önce oluşabilecek eksiklikler tespit edilebilir. Bu sayede, gerekli yazılımların geliştirilmesi ve test edilmesi imkânı elde edilir. Sunulan yaklaşım, Gazebo simülatörü ve ROS programı ile tank sürüş mobil bir robotta modelleme ve simülasyonu gerçekleştirilen sistematik yapı sayesinde insansız kara araçları gibi diğer mobil robotik sistemlerinin de çalışmalarının geliştirilmesine katkı sunar. Ayrıca bu alanda çeşitli uygulamaların yapılabilmesine de olanak da sağlar. Bu çalışmada, tank sürüşü özellikli bir mobil robotun konum kontrolü yapılarak bir değerlendirme sunulmuştur.

Anahtar Kelimeler: ROS, Gazebo, Tekerlekli robot, Tank sürüş, Simülasyon, Konum Kontrolü, Kapalı döngü kontrolü.

*Sorumlu yazar / Corresponding author: ybkoca@aku.edu.tr

Bu makaleye atıf yapmak için /To cite this article

Koca, Y.B., Gökçe, B., Aslan, Y. (2020). ROS/Gazebo Ortamında Tank Sürüş Özellikli Mobil Bir Robotun Similasyonu. Journal of Materials and Mechatronics: A (JournalMM), 1(1), 29-41.

Simulation of A Skid Steer Driving Mobile Robot in ROS / Gazebo Environment

ABSTRACT: Robot simulators provide the ability to predict the performance of theoretical findings before applying them to real robots. In this study, 3D modeling and simulation is presented for a mobile robot with skid steer driving feature by using Gazebo simulator and Robot Operating System (ROS), which aims to analyze system performance in a mobile robot. In this context, a mobile robot with non-holonomic skid steer driving is designed, all physical and inertial properties are defined and simulated. Gazebo provides simulation of the robot world environment, physical model, sensors and control system via the Unified Robot Description Format (URDF) and parameterized robot component macro (XACRO) file. With this simulation, deficiencies that may occur before the real system application can be detected in the robot. In this way, it is possible to develop and test the necessary software. The presented approach contributes to the development of other mobile robotic systems, such as unmanned ground vehicles, thanks to the systematic structure modeled and simulated in a mobile driving robot with a Gazebo simulator and ROS program. It also enables various applications in this area. In this study, an evaluation is made by position control of a mobile robot with skid steer driving.

Keywords: ROS, Gazebo, URDF, Wheeled robot, Skid steer drive, Simulation, Velocity Control, Closed loop control.

1. GİRİŞ

Mobil otonom bir robot, çevresi hakkında bilgi elde edebilen ve tanımlı olan çevresel bilgisini anlamlı ve güvenli bir şekilde hareket etmek için kullanabilen bir makinedir (Arkin 1998). Mobil robotlar, modern toplumla birlikte insanoğlunun hayatında giderek artan bir öneme sahip olmaktadırlar. Kara, hava ve su gibi değişik ortamlarda görev alabilen otonom mobil robotlar izleme, değerlendirme, tanımlanan görevleri yapmaya yönelik olarak birçok görevi yerine getirmek üzere tasarlanmaktadırlar. Otonom mobil robotların geliştirilmesi için uygulama türü oluşturan temel robot platformları için büyük önem arz eder. Ayrıca bu uygulamalarda kullanılacak olan diğer aktüatörler, sensörler gibi çeşitli hususlar dikkate alınarak robotlar tasarlanmaktadırlar (Bekey, 2005; Matarić ve ark., 2007). Mobil robotlar kinematik özellikleri açısından farklı şekilde tasarlanabilirler. Bu çalışma da sert bir gövde çerçevesine sahip dört tekerlekli, doğru akım motoru ile çalışan tank sürüş özelliğine sahip bir robot tasarlanmıştır. Tank sürüş robotlar tekerlekleri sağ ve sol taraflarda birbirine senkronize olarak çalışan ve mekanik olarak kilitli sistemlerdir. Tahrik tekerlekleri sağ ve sol taraflarda birbirinden bağımsız olarak sürülebilir. Ayrıca bir direksiyon mekanizmasına sahip olmayan bu araç yapısında araç platformu makinenin gövdesi üzerinde sabit bir düz hizaya sahip olarak manevra yapar.

Karasal uygulamalarda genellikle tekerli mobil robotlar daha yaygın bir kullanım alanına sahiptir. Tekerlekli mobil robot (TMR) tasarlandığı amaç için bir çevrede insansız olarak ve uzaktan herhangi bir kontrol olmaksızın verilen görevleri yerine getirebilme kapasitesine sahip olmalıdır (Rivera ve ark., 2019). Bu açıdan aracın şekli ve kontrol yapısı değişiklikler gösterir. Dolayısıyla bir robotta çok önemli bir bileşen, robotik sistemin uygulamadan önce yazılımsal olarak kontrol, navigasyon ve diğer sistemler açısından tasarlanabilmesi prototip üretimleri ve pratik uygulamalar açısından çok önemlidir (Correa ve ark., 2012). Bu tasarlanan ve oluşturulan mimari ile robota ait tanımlanan özellikleri ve amacını belirleyen hususlar çerçevesinde dünyası hakkında gerekli bilgileri toplayıp derlemesi ve bunları işleyerek görevini eksiksiz ve güvenli bir şekilde tamamlaması beklenir. Burada robotların tercih edilmesinin en önemli faktörü erişilebilirlik, güvenlik ve hayatta kalma

maliyetleri yüksek olan veya insanlar tarafından gerçekleştirilmesi zor ve tehlikeli görevlerde kullanılmalarıdır. Bu zor görevlerde kullanılacak robotların uygulama aşamasından önce bir yazılımsal ortamda tüm parametrelerinin doğru tanımlanabilmesi ve donanımsal yapılarının doğru analiz edilebilmesi çok önemlidir. Robot tasarımı ile ilgili çeşitli program uygulamaları mevcuttur. Bu uygulamalardan son dönemde birçok araştırmacı tarafından tercih edilen yazılım ise robot işletim sistemi (ROS) programıdır.

ROS, Stanford Üniversitesi tarafından STAIR projesinde bir robot işletim sistemi olarak geliştirilmiştir (Joseph, 2015). Karmaşık robotik sistemlerin geliştirilmesi amacıyla kullanılan ücretsiz ve açık kaynaklı bir robot yazılım aracıdır. Donanım soyutlaması, düşük seviye cihaz kontrolü, süreçler arası mesaj geçişi ve paket yönetimi sağladığı için robotlar için bir meta işletim sistemi görevi görür. Esasen ROS' un kendisi bir işletim sistemi olmayıp, bu işletim sistemi altında çalışan bir programdır. Ayrıca, birden fazla bilgisayarda kod elde etmek, oluşturmak, yazmak ve çalıştırmak için araçlar ve kütüphaneler sağlar.

ROS programının en önemli avantajı robot sensör verilerinin, donanım sürücüleri ile uğraşmak zorunda kalmadan uygulanabilmesine ve soyut veri akışı olarak değiştirilmesine izin vermesidir. Bu sayede yazılım geliştiricilerin robotların programlanmasını çok daha kolay hale getirir. Donanım sürücüleri ve arayüzler ile çalışma zorunluluğunun olmaması büyük bir kolaylık sağlamaktadır. Ayrıca ROS, kol denetleyicileri, yüz izleme, haritalama, yerelleştirme ve yol planlama gibi birçok üst düzey uygulama sağlar. Bu sayede program kullanıcılara, programın kendileri için gerekli kısımlarına odaklanma imkânı sunar.

Robotikte, sürekli artan sistem karmaşıklığı ve özerklik düzeyi, araştırmacıları ve yazılımcıları ortaya çıkabilecek entegrasyon sorunlarından kurtarmak ve robotta karar verebilme yeteneklerini artırmak için daha güçlü bir sistem mimarisine ihtiyaç duymaktadır. Buna ek olarak karmaşık tasarım süreci ve farklı robotların davranışlarının entegrasyonu, iyi tanımlanmış bir çevre ortamının desteklenmesini ile ancak mümkün olabilir. Robot çevresi sadece yazılım süreçlerini daha başarılı kılmak için değil, aynı zamanda birbirleri ile uyum farkındalığına dayalı otonom görev planlaması ile yazılı davranışların genişletilmesi açısından da esastır.

Bu çalışma toplam beş başlık altında ele alınmıştır. 2. Bölüm de bir mobil robotu modellemek için kullanılan ROS mimarisi ile birlikte simülasyon ortamında gerçekleştirmek için kullanılan Gazebo programını açıklanmaktadır. Bölüm 3'te tank sürüş özellikli mobil bir robotun ROS programında oluşturulması ve oluşturulan bu modelin Gazebo ortamında uygulanma süreci anlatılmaktadır. Bölüm 4' te ise modeli oluşturulan robotta konum kontrolünün uygulanması, kontrol mekanizması ile elde edilen verilerin değerlendirilmesi sunulmuştur. Son bölümde ise sonuçlar ile birlikte ileriki dönemde yapılması planlanan çalışmalar açıklanmıştır.

2. ROS ve GAZEBO

Robot üretiminin ilk aşaması, tasarımı ve modellemesidir. Bir robot AutoCAD, Solid Works, Blender ve benzeri CAD araçları kullanılarak tasarlanabilir ve modellenebilir. Bir robotu modellemenin temel amaçlarından biri simülasyon programı vasıtasıyla robot davranışlarını, performansını ve doğruluğunu gibi farklı yeteneklerini ölçmektir. Dolayısıyla robotik simülasyon aracı, robot tasarımındaki kritik kusurları kontrol edebilir. Robot üretim aşamasına geçmeden önce çalıştığını doğrulayabilir.

Sanal robot modeli gerçek donanımın tüm özelliklerine sahip olmalıdır. Tasarımı yapılan robotun fiziksel olarak şekli gerçek robot gibi görünebilir veya görünmeyebilir. Ancak gerçek robottaki tüm fiziksel özelliklere sahip soyut bir model olmalıdır. Mobil robotlar tasarlanırken amaç, kararlı mekanik yapılara sahip olurken hassasiyet ve hız gibi istenen işlevleri yerine getirmek için gerekli güvenilirlik ve manevra kabiliyetine sahip modeller elde etmektir. Ayrıca bu analiz, morfolojiye bağlı olarak, robotta amacın yerine getirilmesi için sensörler ve aktüatörler gibi bileşenleri en iyi şekilde düzenlenmesini gerektirmektedir. Bununla birlikte matematiksel olarak tanımlanacak olan kinematik ve dinamik özellikler robotta tanımlanması için ihtiyaç duyulan hesaplamalardır. Ancak aynı mobil robotu simüle etmek için ROS programında istediğimiz işlevselliğe bağlı olarak matematiksel modeller geliştirmeden de tasarlanabilmektedir.

ROS-Gazebo ikilisi genel olarak robot topluluğu tarafından çevrelerinin güvenilirliği ve ortak özelliklerin esnekliği, sağlamlığı ve kullanılabilirliği ile simüle edebilme yeteneği için kullanılan güçlü bir kombinasyondur (Koenig ve Howard, 2004). Bununla birlikte, hem esnekliğe hem de diğer robot ortamlarıyla entegrasyona izin veren bu karmaşıklık yönetiminin, bu Gazebo-ROS platformunun tüm potansiyelinden yararlanmak için sağlam bir bilgi gerektirdiğini anlamak kolaydır (Furrer ve ark., 2016).

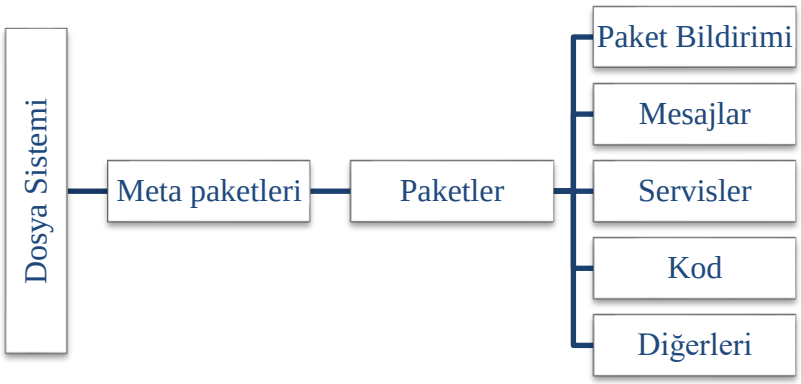
2.1. ROS Programı

Robot teknolojisi son on yılda çok önemli bir gelişim göstermiştir. Özellikle ROS gibi yeni açık kaynaklı platformların ortaya çıkması, robot sistemlerinin hem araştırma imkanlarını hem de nihai kullanıcı uygulamalarında daha erişilebilir hale getirmesine olanak sunmuştur (Alajlan ve Koubâa, 2016). Bu imkanlar neticesinde ROS, robot araştırmaları ve şirketlerin robotları modellemesi, simüle etmesi ve prototip üretmesi için en yaygın kullanılan yazılım çerçevelerinden birisi haline gelmiştir (Mahtani ve ark., 2018). ROS, geliştiriciler açısından çok büyük bir dönüşüm ve geliştirme ortamı sunmuştur. Her geçen gün ROS yeni uygulamalar ve robotlar oluşturmak için bir dizi araç, altyapı ve uygulamaları sağlayarak önemli bir yer edinmektedir.

ROS mimarisi üç bölüme veya kavram seviyesine ayrılmıştır. Bunlar; dosya sistemi seviyesi, hesaplama grafiği seviyesi ve topluluk düzeyi olarak sınıflandırılabilir. ROS kullanarak mobil bir robotta üç boyutlu modelini oluşturmayı simüle etmeyi planlıyorsak, robot tasarımına yardımcı olan bazı ROS paketleri hakkında da bilgi edinmek gerekmektedir. ROS, robot_model adı verilen ve urdf, kdl_parser, robot_state_publisher, collada_urdf vb. paketlerden oluşan robot modelleri tasarlamak ve oluşturmak için standart bir meta pakete sahiptir (Joseph, 2015). Bu paketler, gerçek donanımın kesin özelliklerine sahip üç boyutlu bir robot modeli tanımlamasını oluşturmamıza yardımcı olur. ROS mimarisinin temel kavramlarını aşağıda sırasıyla ele alalım.

2.1.1. ROS Dosya Sistemi

ROS dosya sistemi, bir robot geliştirme sürecini merkezileştirmek ve aynı zamanda bağımlılıklarını merkezden uzaklaştırmak için yeterli esneklik ve araç sağlamaktır. Bir işletim sistemine benzer şekilde bir yapıya sahip olan ROS programı klasörlere ayrılır. Ayrıca bu klasörlerin işlevlerini açıklayan dosyaları vardır. Şekil 1’ de ROS dosya sisteminin genel şeması görülmektedir (Mahtani ve ark., 2018).



Şekil 1. ROS dosya sistemi genel prensip şeması.

ROS, tüm işlemlerin bağlı olduğu bir ağ oluşturur. Sistemdeki herhangi bir düğüm bu ağa erişebilir, diğer düğümlerle etkileşime girebilir, gönderdikleri bilgileri görebilir ve ağa veri iletebilir. Bu seviyedeki temel kavramlar düğümler, master, parametre sunucusu, mesajlar, hizmetler, konular ve depolardır. Bunların hepsi grafiğe farklı şekillerde veri sağlarlar. Aşağıda Şekil 2' de ROS ağ yapısını gösteren şema sunulmuştur (Mahtani ve ark., 2018).

.

Şekil 2. ROS ağ yapısı

2.1.2. Robot modelleme için ROS paketleri

ROS, üç boyutlu robot modelleri oluşturmak için kullanılabilecek bazı paketler sunar. Bunlardan robot modelleri oluşturmak için yaygın olarak kullanılan bazı önemli ROS paketleri şunlardır.

- ***robot_model:*** ROS, üç boyutlu robot modellerinin oluşturulmasına yardımcı olan önemli paketler içeren *robot_model* adında bir meta pakete sahiptir. Bu meta paketin içindeki tüm önemli paketleri görebiliriz. Bunlar sırasıyla aşağıda verilmiştir.

- ***urdf:*** *robot_model* meta paketi içindeki önemli paketlerden birisidir. Birleşik robot tanımlama biçimi (URDF) paketi, bir robot modelini temsil eden bir *xml* dosyasıdır. URDF, *C++* yazılım dilini kullanır. URDF kullanarak bir robot modelinde sensörler, çalışma ortamı tanımlanabilir. Ayrıca URDF çözümleyicileri kullanarak bunları ayrıştırabiliriz. URDF' de yalnızca bir ağaç yapısı şeklinde katı uzuvlar tanımlanır. Robotta katı uzuvlar birbirleriyle eklemler kullanılarak bağlanırlar. Ayrıca esnek uzuvlar URDF kullanılarak temsil edilemez.

- ***joint_state_publisher:*** Bu paket, robot modeli tanımlamasını okuyan, tüm eklemleri bulan ve gui kaydırıcılarını kullanarak sabitlenmemiş tüm eklemlere ortak değerleri yayınlayan joint_state_publisher adlı bir düğüm içerir. Kullanıcı bu aracı kullanarak her bir robot eklemiyle etkileşime girebilir ve RViz kullanarak görselleştirebilir. URDF tasarlarken, kullanıcı bu aracı kullanarak her bir eklemin dönüşünü ve çevirisini doğrulayabilir.

- ***kdl_parser:*** Kinematik ve dinamik kütüphanesi (KDL), URDF kapsamında çözümleyici araçları içeren bir ROS paketidir. Kinematik ağaç, eklem durumlarını yayınlamak için kullanılabilir. Ayrıca ileri ve ters kinematik için de kullanılabilir.

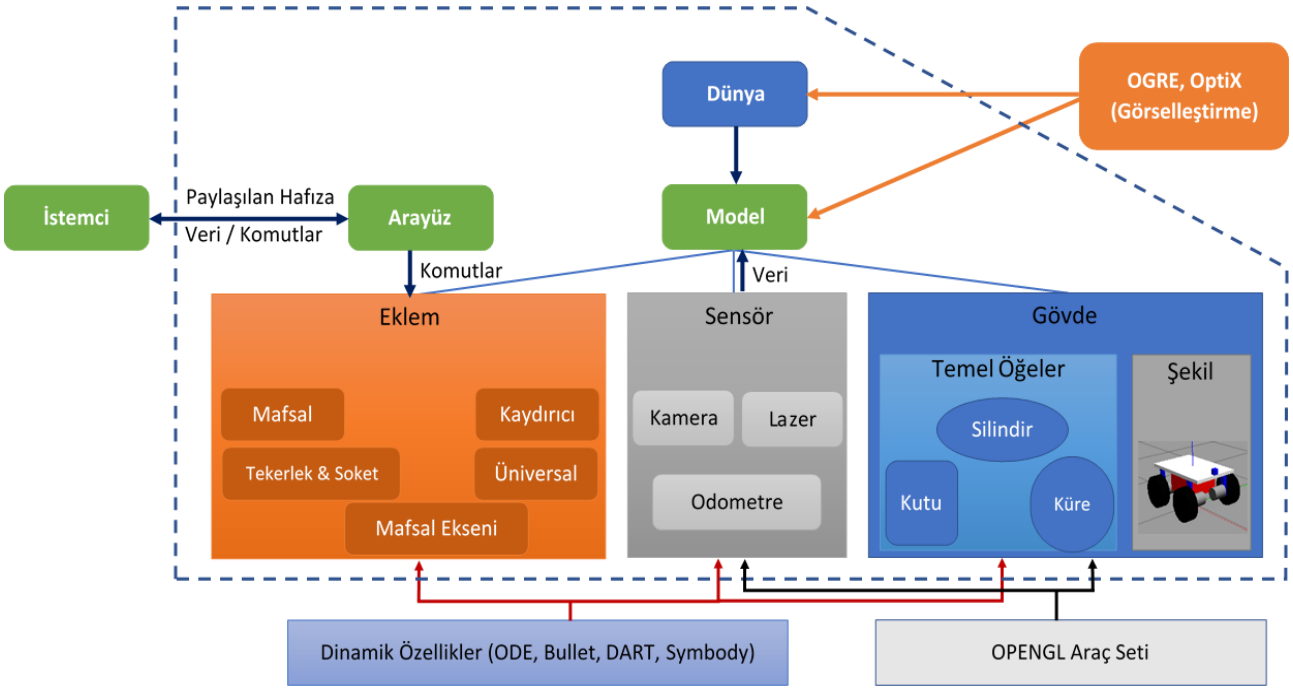
- ***robot_state_publisher:*** Bu paket, mevcut robot eklem durumlarını okur. URDF' den kinematik ağacı yapısını kullanarak her bir robot uzvunun üç boyutlu pozlarını yayınlar. Üç boyut pozu ROS transformasyonu (tf) olarak yayınlanır. ROS tf, bir robotta koordinat çerçeveleri arasındaki ilişkiyi ortaya koyar.

- ***xacro:*** *Xml* makroları anlamına gelir. Xacro URDF ' i daha kısa, okunabilir hale getirmek için bazı eklentiler içerir ve karmaşık robot tanımlamaları oluşturmak için kullanılabilir. Bazı ROS araçlarını kullanılarak xacro her zaman URDF formatına dönüştürülebilir.

2.2. Gazebo Simülasyon Programı

Gazebo, robot gelişimi için gerekli olan üç boyutlu simülasyon için robotlar, sensörler, çevre modelleri sağlayan ve fizik motoru ile gerçekçi simülasyonlar sunan bir simülatördür. Gazebo son yıllarda en popüler simülatörlerden biridir. Ayrıca Gazebo, ROS ve topluluğundan sorumlu Open Robotik tarafından geliştirilip dağıtılmaktadır, bu nedenle ROS ile uyumludur (Koenig ve Howard, 2004). Gazebo tek başına bir program olarak çalışabilir, ancak Gazebo ile farklı türde bir Uygulama Programcısı Arabirimi (API) ve kitaplık kullanarak bağlantı kurmak için kullanılabilirlik de vardır.

ROS, Gazebo' ya bağlanmak için kullanılabilen ve robotik alanda güçlü bir araç oluşturan en popüler uygulamalardan biri olarak kabul edilir. Bunu yapmak için ROS, bağımsız Gazebo ortamında bağıntılar sağlayan ve farklı ROS bileşenleriyle entegrasyon sağlayabilen *gazebo_ros_pkgs* adlı bir paket kullanır. Bu paketler, ROS mesajları, konuları ve hizmetleri kullanarak Gazebo da bir robotu simüle etmek için gerekli arayüzleri sağlar. Ayrıca Gazebo' da bulunan sonar, tarama lazer mesafe bulucuları ve GPS gibi sensör modelleri ile iletişim kurma yeteneği sağlar. Şekil 3’ de Gazebo uygulamasının genel mimarisi görülmektedir.



Şekil 3. Gazebo genel mimarisi

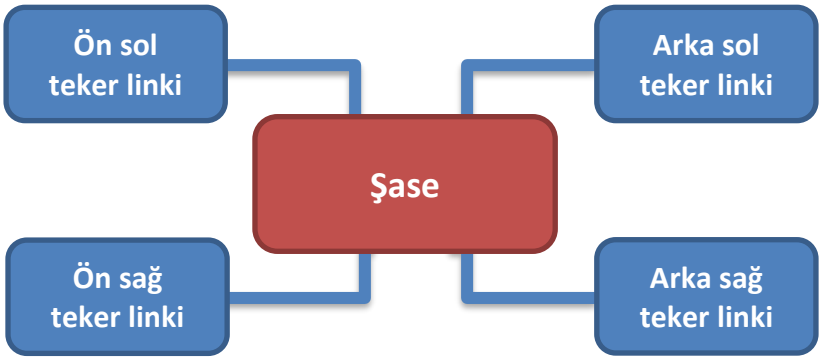
3. Robot Modellemesi / ROS ortamında Tank Sürüş Robot Modeli Oluşturma

Bir simülasyon modeli oluşturmaya başlamadan önce, modelin bireysel bileşenlerinin parametrelerini bilmek gerekir. Bu örnekte, bilinmesi gereken parametreler, robot bireysel bağlantılarının, eklemlerinin, eklem tiplerinin ve robot hareketinin konumu ve geometrisidir. Bu çalışmada, robotik araştırmalarda kullanılmak üzere tasarlanan ve geliştirilen tank sürüş bir robot simülasyonu gerçekleştirilmiştir. Tank sürüş robot simülasyonu ROS ve Gazebo kullanılarak gerçekleştirilmiştir.

Robot simülasyonlarında bir robotu modelleme, bu robottan alınan çeşitli sensör verilerini işleme, aktüatörleri kontrol etme, algoritmaları test etme veya değerlendirme yeteneklerine sahip olması açısından önemlidir. ROS' ta robot kinematik modelini, gerçekleştirebilmek için uygun bir formata çevirmeye izin veren bir yapıya ihtiyaç duyulmaktadır. ROS bu özelliği, robot modellerinin, her bir modelin karşılık gelen serbestlik derecesine göre hareket ettirilebildiği veya çalıştırılabileceği bir üç boyutlu model olarak ifade edilen Birleşik Robot Tanımlama Biçimi (URDF) adlı bir *xml* biçiminde tanımlanabildiği *xml* formatında sağlar. Böylece simülasyon veya kontrol için kullanılabilir. URDF dosyaları, kaç tane tekerleği olduğu, nereye yerleştirildikleri ve hangi yönlere döndükleri gibi robot fiziksel yapılandırmasını tanımlar. Bu bilgiler, Rviz (ROS programının üç boyutlu görselleştirme aracı) tarafından görselleştirmek için kullanılır (Yılmaz ve Bayındır, 2019).

Aşağıdaki model, tank sürüş mobil bir robot için basit bir simülasyon modeli oluşturmanın çeşitli yönlerini açıklamak için kullanılacaktır Dört tekerlekli temel model, ağaç benzeri bir yapıda beş farklı parça ile tanımlanabilir. Şekil 4, tasarlanan robota ait temel bağlantılarının bir blok diyagramını göstermektedir. Robot merkezi şasisi temel çerçeve olacaktır. Kırmızı dikdörtgen kutu taban bağlantısı olacaktır. Tekerleklerimizin dönmesini istediğimiz için, bu robottaki her bir tekerlek için ayrı bağlantılar oluşturmamızı gerektirir. Bunlar alt bağlantılar olarak tanımlanabilir. Oluşturulan bağlantıların geometrisi bağlantı konumu, bağlantı tipi ve bağlantı ekseni tanımlanmalıdır. Bir bağlantı konumu konum ve yönlendirmeye göre sınıflandırılabilirken, bir eklem türü tam, sürekli,

sabit veya prizmatik olarak sınıflandırılabilir. Eklem konumunu ve yönünü tanımlamak için bir eklem ekseni belirtilmelidir.



Şekil 4. Temel nokta dönüşlü robot bağlantı bloğu şeması

Robotun bağlantı bloğu oluşturulduktan sonra, robotun yol almasında gerekli olan özellikler aşağıdaki gibi tanımlanabilir. Buna göre robot için tork ve motor devri hesaplamaları yapılabilir.

- Maksimum taşıma kapasitesi = 40 kg
- Robot ağırlığı = 40 kg
- Maksimum hız = 1 m/s
- Tank sürüş özellikli yapılandırma
- Dikdörtgen taban alanı ve
- Otonom seyir ve engellerden kaçınmadır.

Tasarlanan robot dört tekerlekten tahrik konfigürasyonuna sahiptir. Robotun motor tork değerleri maksimum taşıma kapasitesi ve robotun kendi ağırlığı ile aşağıdaki gibi hesaplanabilir.

- Toplam robot ağırlığı = Robotun ağırlığı + Yük kapasitesi
- Robotun ağırlığı= 40 x 9.8 ≈ 400 N.
- Kapasitesi= 40 x 9.8 ≈ 400 N.
- Toplam ağırlık= 400 + 400 = 800 N.

Robota ait tekerleklerinin konfigürasyonu Şekil 5'de gösterilmektedir. Robot hareketsizse, tekerleklere takılan motorların hareket edebilmesi için maksimum tork uygulanması gerekir. Maksimum tork denklemi,

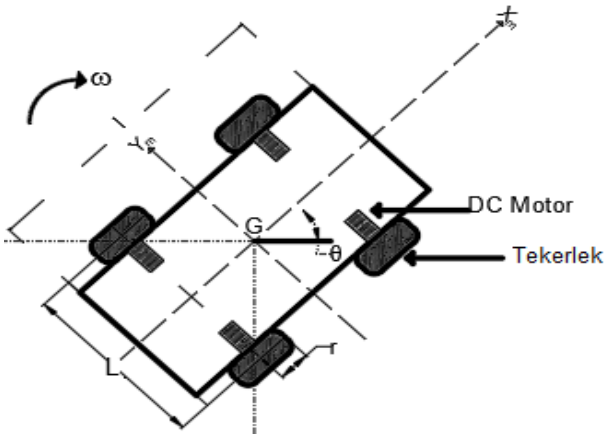
$$\tau = \mu \, x \, N \, x \, r \tag{1}$$

Burada, μ sürtünme katsayısı, N her bir tekerlek üzerinde etki eden ortalama ağırlık, r tekerleklerin yarıçapı ve τ hareket etmek için maksimum torktur. $N = W/4$ yazılabilir. Çünkü robotun ağırlığı dört tekerleğin hepsine eşit olarak dağıtılır. Ayrıca sürtünme katsayısı olarak 0,6 sabit değeri boş dünya ortamı için kabul edilmiştir. Bu durumda,

$$\tau = 0{,}6 \, x \, (\frac{W}{4}) \, x \, r$$

$$\tau = 0{,}6 \, x \, \left(\frac{800}{4}\right) x \, 0{,}225 = 27 \, N.m$$

olarak hesaplanır. Böylece, tork için 270 kg-cm' ye karşılık olarak 300 kg-cm standart bir değer kullanabilir.



Şekil 5. Robot tekerlek konfigürasyonu

Verilen özelliklere göre motorlara ait devir hesaplaması ise aşağıdaki gibi yapılabilir. Buna göre;

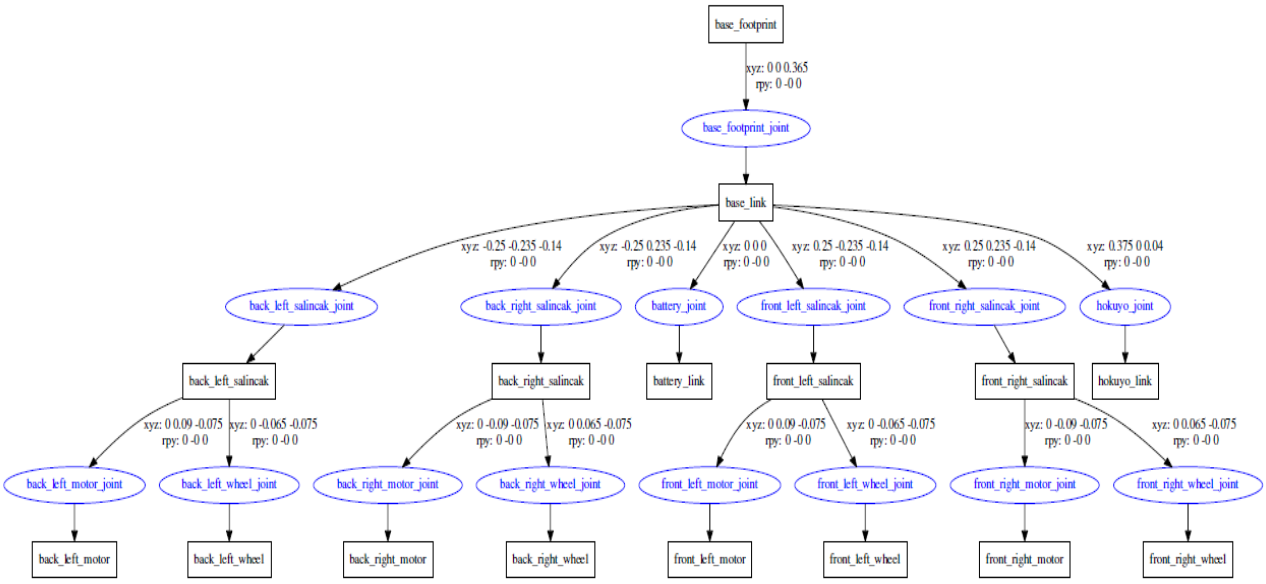
$$RPM = \frac{60\ x\ Hız}{\pi\ x\ r} \tag{2}$$

$$RPM = \frac{60\ x\ 1}{\pi\ x\ 0.225} = 84\ d/dk$$

Bu robot için 100 RPM değerine sahip bir motor seçilebilir. Tüm bu parametreler hesaplandıktan sonra robota ait özellikler ve şasi tasarımı *xml* formatında tanımlanarak robotun modeli oluşturulabilir.

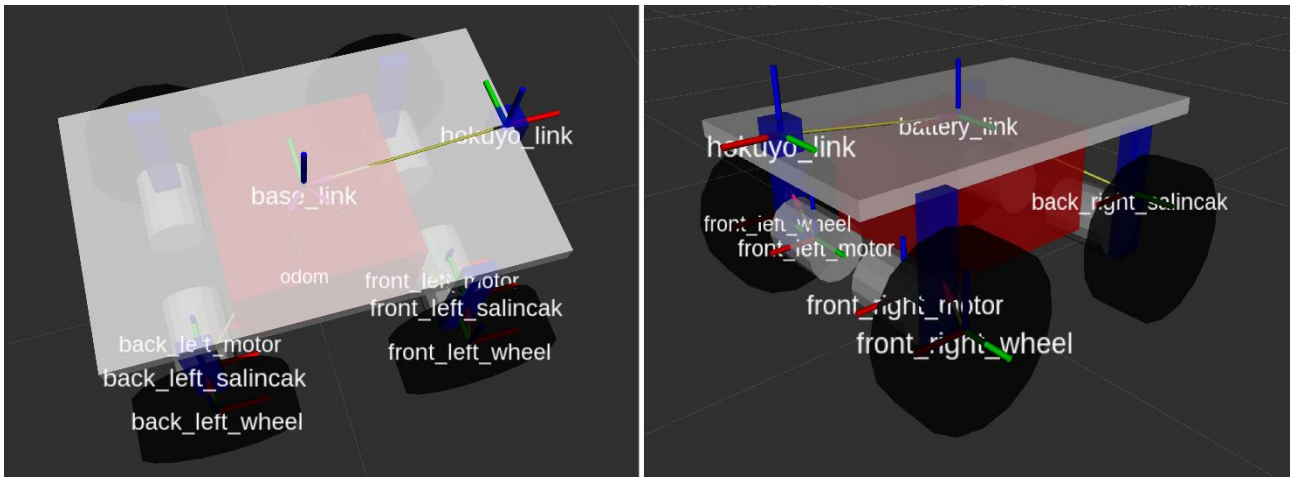
ROS programında gerçekleştirilmek istenen herhangi bir robotik sistem için ilk olarak dosya tanımlamalarının yapılması şarttır. Bu çalışmada da tank sürüş özellikli bir robot sistemi için ilk olarak bir ROS paketi oluşturulmuştur. Oluşturulan bu paket boş bir paket daha sonra ana dosyanın içerisinde özelliklerini oluşturacağımız holonomik olmayan robot tasarımı gerçekleştirilmiştir.

Tank sürüş tasarlanan robota ait fiziksel özellikler urdf dosyası içinde yazılarak robot fiziksel görünümü oluşturulmuştur. Oluşturulan tank sürüş robota ait ağaç şeması Şekil 6’ da verilmiştir.



Şekil 6. Tank sürüş robot ağaç diyagramı.

Oluşturulan robot tanımlamasının Rviz ortamında elde edilen görünümü Şekil 7' de verilmiştir.



Şekil 7. Tank sürüş robotun Rviz ortamında elde edilen model ve simülasyon görünümü.

4. DENEYSEL SONUÇLAR

Tasarlanan robot Gazebo ortamında boş dünya tanımlaması ile gerçekleştirilmiştir. Robotun pozisyon kontrolü lineer hız ve açısal hız için aşağıda tanımlanan ifadelere göre belirlenmiştir. Lineer hız ve açısal hız için tanımlanan maksimum ve minimum hız limitlerinin haricinde oransal bir katsayı tanımlanmıştır. Buna göre (2)' de lineer hız için hedef konuma göre mevcut konumun arasındaki mesafe hesaplanmıştır. Açısal hız içinde robotun yönüne göre (3)' de hedef konum ile mevcut konum arasında açı hesaplanarak bir oransal katsayı ile çarpılmıştır. Burada doğrusal hız için $K_v = 2$ ve açısal hız için $K_h = 0.8$ olarak alınmıştır.

$$V' = K_v . \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (3)$$

$$\theta^{'} = K_h.\left[\tan^{-1}\left(\frac{Y_2 - Y_1}{X_2 - X_1}\right)\right] \quad (4)$$

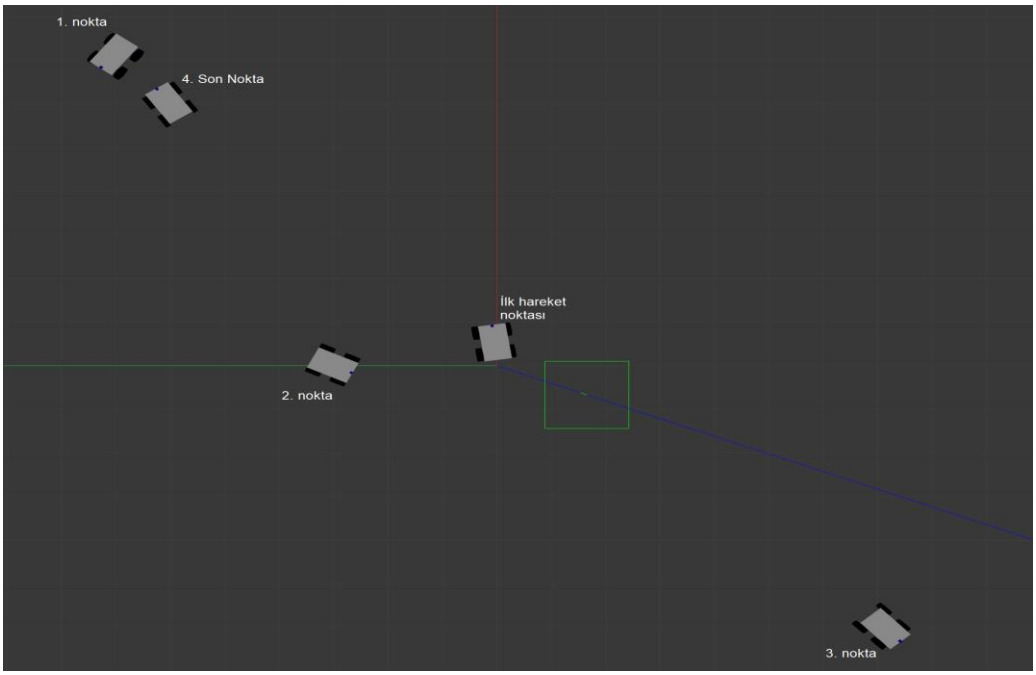
Bu çalışmada, robotun pozisyon kontrolü, Çizelge 1' de verilen koordinat noktalarına göre tanımlanmıştır.

Çizelge 1. Robot güzergâh noktaları

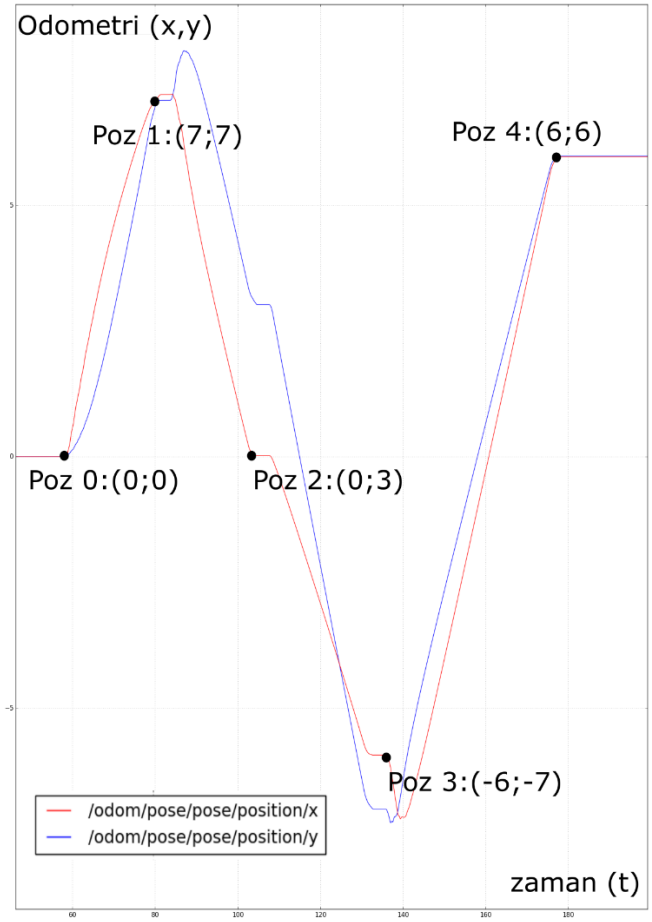
	X	**Y**
İlk hareket noktası	0	0
1. Nokta	7	7
2. Nokta	0	3
3. Nokta	-6	-7
4. Nokta	6	6

Gazebo simülasyon ortamı üzerinde aracın otonom hareketi için bazı yörünge/koordinat tanımlamaları yapılmıştır. Buna göre araç önce odometri merkezinde X ve Y koordinat düzleminde (0;0) konumuna yerleştirilmiştir. Şekil 8' de aracın simülasyon ortamında üstten görünüşü verilmiştir. Bununla birlikte, Şekil 9'da aracın başlangıç noktası ile x ve y eksenindeki oturduğu noktalar grafiksel olarak gösterilmiştir. Grafiklerde görüldüğü gibi aracın ilk hareket noktasından birinci pozisyona sorunsuz bir şekilde oturduğu ve 5 saniyelik bekleme sonrasında ikinci nokta için manevra aldığı görülmektedir. Burada aracın birinci pozisyondan ikinci pozisyona geçerken herhangi bir osilasyona girmediği ve aracın doğrusal ve açısal ivmelenmelerde aracın ataletsel özelliklerini kullanarak sorunsuz bir şekilde hareket ettiği görülmektedir.

Aracın hareket sırasında herhangi bir yanal kaymaların olmadığı da görülmektedir. Buda aradın hem fiziksel, ataletsel ve kontrol özelliklerinin doğru bir şekilde tanımlandığını göstermektedir. Grafik üzerinde birinci güzergâh noktasından ikinci güzergâh noktasına geçiş yaparken keskin bir dönüş yapıldığı görülmektedir. Bu dönüşte robot 'un çok kararlı bir şekilde dönüş yapmış ve belirlenen istikamete doğru yönelim göstermiştir. Aynı şekilde üçüncü ve dördüncü güzergâhlar arasında da robotun belirlenen istikamete kararlı bir şekilde dönüş yaptığı da şekilden gözlenmektedir.



Şekil 8. Tank sürüş robotun tanımlanan noktalara yol alması



Şekil 9. Tank sürüş robotun tanımlanan pozisyon noktalarına ilişkin yol aldığını gösterir grafik.

5. SONUÇLAR

ROS/Gazebo Simülasyon yazılımları çeşitli arazilerde veya ortamlarda robot gerçek zamanlı davranışını tahmin etmek için kullanılır. Bu uygulamalarda robot tasarımının doğru ve verimli bir şekilde kurulduğunda ve test edildiğinde bu gelecekteki gerçek robot davranışı beklentileri hakkında bilgi verecektir. Bu çalışmada, hareketi dört tahrik tekerleğine dayanan tank sürüş dört tekerlekli mobil bir robotun konum kontrolü yapılarak simülasyonu gerçekleştirilmiş ve test edilmiştir. Simülasyon. Gerçek zamanlı bir robot davranışının beklendiği gibi olabileceğini göstermektedir. Robotun simüle edilmiş ortamdaki bu istikrarlı sonuçları ve performansı sayesinde ROS/Gazebo sistemleri ile sağlam ve kararlı olan gerçek zamanlı bir robotta uygulama imkânı elde edilmiştir. Robotun belirlenen noktalara başarı ile gittiği görülmektedir. Daha sonraki çalışmalarda farklı robot tasarımları, yüzey şartları veya tanımlanan görevler için simülasyonlar ile performansları gerçekleştirilerek analizler yapılabilir.

6. KAYNAKLAR

Rivera Z.B., Marco C.S., Domenico G., Unmanned Ground Vehicle Modelling in Gazebo/ROS-Based Environments, Machines, 7(2), 1–21, 2019.

Zahir Y., Bayındır L., Simulation of Lidar-Based Robot Detection Task Using ROS and Gazebo, European Journal of Science and Technology, (October), 513–29, 2019.

Correa D.S.O., Sciotti D.F., Prado M.G., Sales D.O.W., Denis F., Osório F.S., Mobile Robots Navigation in Indoor Environments Using Kinect Sensor, Proceedings - 2012 2nd Brazilian Conference on Critical Embedded Systems, CBSEC 2012, 36–41, 2012.

Koenig N., Andrew H., Design and Use Paradigms for Gazebo an Open-Source Multi-Robot Simulator, In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (IEEE Cat. No. 04CH37566), IEEE, 2149–2154, 2004.

Alajlan M., Anis K., Studies in Computational Intelligence Writing Global Path Planners Plugins in ROS, A Tutorial, 2016.

Arkin R.C., Ronald C.A., Behavior-Based Robotics, MIT press, 1998.

Bekey G.A., Autonomous Robots: From Biological Inspiration to Implementation and Control, MIT press, 2005.

Furrer F., Michael B., Markus A., Roland S., Robot Operating System (ROS): The Complete Reference, Springer International Publishing, Volume-1, 595–625, 2016.

Joseph L., Physiological Research Mastering ROS for Robotics Programming, 2015.

Mahtani A., Joseph L., Fernández E., Martinez A., Sánchez L., ROS Programming: Build Powerful Robots: Design, Build, and Simulate Complex Robots Using the Robot Operating System, Packt Publishing, 2018.

Matarić M.J., Maja J., Ronald C.A., The Robotics Primer, Mit Press, 2007.

İnt. Ky. 1. ROS'a genel bakış, http://gazebosim.org/tutorials?tut=ros_overview (Erişim Tarihi: 11.05.2020).