Eğitim ve Araştırma Amaçlı Gezgin Robot Geliştirilmesi

Mehmet Akçakoca

YÜKSEK LİSANS TEZİ

Elektrik Elektronik Mühendisliği Ana Bilim Dalı

Mayıs 2017

Development of a Mobile Robot for Educational Studies and Research

Mehmet Akçakoca

MASTER OF SCIENCE THESIS

Department of Electric Electronic Engineering

May 2017

Eğitim ve Araştırma Amaçlı Gezgin Robot Geliştirilmesi

Mehmet Akçakoca

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik Elektronik Mühendisliği Ana Bilim Dalı
Kontrol ve Kumanda Sistemleri Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Ahmet Yazıcı

"Bu tez Bilim, Sanayi ve Teknoloji Bakanlığı tarafından, 2014/18 numaralı KOSGEB projesi tarafından desteklenmiştir"

Mayıs 2017

ONAY

Elektrik Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi Mehmet Akçakoca'nın YÜKSEK LİSANS tezi olarak hazırladığı "Eğitim ve Araştırma Amaçlı Gezgin Robot Geliştirilmesi" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek oy birliği ile kabul edilmiştir.

Danışman **:** Doç. Dr. Ahmet YAZICI

İkinci Danışman **:** -

Yüksek Lisans Tez Savunma Jürisi:

Üye : Prof. Dr. Rıfat EDİZKAN

Üye : Yrd. Doç. Dr. Hanife APAYDIN ÖZKAN

Üye :

Üye :

Üye :

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Hürriyet ERŞAHAN
Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Ahmet Yazıcı danışmanlığında hazırlamış olduğum "Eğitim ve Araştırma Amaçlı Gezgin Robot Geliştirilmesi" başlıklı YÜKSEK LİSANS tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 30.05.2017

Mehmet Akçakoca

İmza

ÖZET

Gezgin robotlar artan şekilde üretim, sağlık, uzay ve savunma alanlarında insanlara hizmet vermektedir. Bu alanlarda geliştirme yapan personelin eğitimi veya saha uygulamasından önce geliştirilen sistemlerin testi için eğitim ve araştırma amaçlı gezgin robotlara ihtiyaç duyulmaktadır.

Eğitim ve araştırma amaçlı çok geniş yelpazede gezgin robot platformları geliştirilmiştir. Gezgin robotlar üretim amaçlarına göre sadece eğitim, sadece araştırma ve hem eğitim hem de araştırma amaçlı gezgin robotlar olarak gruplanabilmektedir. Buna rağmen, geliştirilen platformun kullanımı kullanıcıya bağlı olduğundan bu kategoriler arasında katı bir ayrım bulunmamaktadır.

Bu çalışmada, eğitim ve araştırma amaçlı gezgin robot platformu (Evarobot) geliştirilmiştir. Robot üzeri çalışmalarda kolaylık sağlamak ve hızlı uygulamalara imkân vermek için robotun mekanik ve elektronik donanımları ve kontrol yazılım mimarisi modüler ve tekrardan konfigüre edilebilir şekilde tasarlanmıştır. Robotun yazılım mimarisi Robot İşletim Sistemi (ROS) ara katmanı kullanılarak geliştirilmiştir. 3B benzetim ortamı Gazebo için robotun kinematik ve dinamik modeli hazırlanmıştır. Ayrıca Evarobot'un MATLAB üzerinden kontrolünün gerçekleştirilebilmesi için ilgili kütüphaneler geliştirilmiştir. Eğitim ve araştırma amaçlı kullanımlarda kolaylık sağlamak için Evarobot'un geniş kullanıcı dokümanları ve deney dokümanları hazırlanmıştır.

Geliştirilen gezgin robot platformunun testleri Gazebo benzetim ortamı ve gerçek ortamda yapılmıştır. Sistemin bütünleşmiş şekilde çalışırlığının kontrolü için uzaktan kumandalı kontrol, rastgele dolaşma davranışı ve eş zamanlı konumlandırma ve haritalama testleri her iki ortamda da başarı ile tamamlanmıştır.

Anahtar Kelimeler: Eğitim ve Araştırma Robotu, Gezgin Robot, Robot İşletim Sistemi, Gazebo

SUMMARY

Mobile robots are increasingly serving people in the field of production, health, space and defense fields. Mobile robot platforms are required for education of the people who develop products in these areas, or initial testing of developed systems before field application.

A wide range of mobile robot platforms have been developed for educational studies and research. Mobile robot productions can be grouped as the ones for only the purposes of the education, only for the research and both of them. However, there is no strict distinction between these groups since the use of the developed platforms depend on the user.

In this study, a mobile robot for educational studies and research was developed. The robot's mechanic, electronic hardware and control software was designed to be modular and reconfigurable to provide quick application development. The software architecture of the robot was developed using Robot Operating System (ROS) middleware. Kinematic and dynamic model of the robot was prepared for the GAZEBO which is a 3D simulation environment. In addition, related software libraries have been developed so that Evarobot can be controlled via MATLAB. Extensive user and laboratory manual of Evarobot have been prepared for ease of use in educational and research purposes.

The tests of the developed mobile robot platform have been carried out both in Gazebo simulation environment and real environment. In order to verify the integrated operation of the system, the wander, simultaneous localization and mapping, and teleportation tests have been successfully completed in both simulation and real environments.

Key Words: Education and Research Robot, Mobile Robot, Robot Operating System, Gazebo

TEŞEKKÜR

Bu çalışmanın planlanması ve yürütülmesinde yardımlarını esirgemeyen danışmanım Doç. Dr. Ahmet Yazıcı'ya, yüksek lisans eğitimim süresince yanımda olarak manevi desteklerini esirgemeyen aileme, teknik donanım ve alt yapı desteği sağlayan şirketim İnovasyon Mühendislik ve değerli çalışanlarına saygıyla teşekkürlerimi sunarım.

Bu tez Bilim, Sanayi ve Teknoloji Bakanlığı tarafından, 2014/18 numaralı KOSGEB projesi tarafından desteklenmiştir.

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER (devam)

Sayfa

ŞEKİLLER DİZİNİ

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ŞEKİLLER DİZİNİ (devam)

Şekil **Sayfa**

ÇİZELGELER DİZİNİ

Çizelge **Sayfa**

1 GİRİŞ VE AMAÇ

Son on yılda gelişen işlemci ve algılayıcı teknolojisi, robotlar ve otonom sistemler üzerine yapılan çalışmaların hızlanmasında olumlu bir etkiye sahiptir. Robotlar da artan şekilde üretim, sağlık, hizmet, uzay ve savunma alanlarında hizmet vermektedir. Robotik çalışmalarının bir alt alanı olan gezgin robotlar, otonom hareket edebilmekte veya uzaktan kontrol edilebilmektedir. Gezgin robotlar günümüzde birçok alanda tam veya yarı otonom olarak insanlara hizmet etmektedir. Lojistik robotları, temizlik robotları, keşif robotları, bomba imha robotları, arama ve kurtarma robotları ve rehber robotlar bunlardan bazılarıdır.

Günlük hayatta geniş bir yelpazede kullanım alanına sahip olan otonom gezgin robotlar, 21. yy.'ın sanayi devrimi olarak görülen Endüstri 4.0 kapsamında da önemli bir yere sahiptir. Robotların üstlendikleri bu görevleri daha başarılı bir şekilde yerine getirebilmesi için robotların öğrenme kapasitesi, insanlarla ve çevresiyle etkileşme özelliği olacak şekilde tamamen otonom geliştirilmesi önem kazanmaktadır. Bu geliştirmeler için nitelikli iş gücünün yetiştirilmesi önemlidir. Bu kapsamda üniversitelerde öğrencilere güçlü temel robot bilgisi ve gerçek robot sistemler ile uygulamalı deneyim sunulmalıdır. 2013 yılında Amerika'da robotik için çıkarılan yol haritasında (Robotics Technology Consortium, 2013), robotiğin temeline ve ilgili teknolojilere hâkim olan iş gücü ile yapılacak araştırma sonuçlarının ülkeye büyük avantajlar kazandıracağı vurgulanmaktadır. Bu amaçla yapılan 15 yıllık yol haritasında ortaokul seviyesinden doktora seviyesine kadar uzanan robotik çalışmalar üzerine durulmaktadır. Gezgin robotların geliştirilmesi için, mekanik, elektronik, bilgisayar, sinyal işleme ve otomatik kontrol gibi birçok disiplinde çalışmalara ihtiyaç vardır. Gezgin robotların birçok disiplinlerin etkileşimlerine imkân sağlaması nedeni ile mükemmel bir eğitim platformu olma özelliği vardır.

Diğer taraftan, insanın bulunmasının sakıncalı olduğu ortamlar için veya rutin işgücü uygulamaları için insansız araçlar geliştirilmektedir. Geliştirilen insansız araçlar insanlarla aynı ortamda yaşayacağı için gürbüz bir sisteme sahip olması gerekmektedir. Bu yüzden insansız araçlar üzerine yapılan çalışmalar ve geliştirmeler hala devam etmektedir.

İnsansız kara araçları üzerinde çalışmaları yaygınlaştırmak ve geliştirmek amacıyla 2004, 2005 ve 2007 yıllarında Amerika savunma bakanlığına bağlı olan DARPA tarafından yarışmalar düzenlemiştir. Bu yarışmalarda elde edilen bilgi birikimi ile günümüzde Google (Google, 2016), Ford (Ross, 2016), Tesla (Tesla Motors, 2016), Uber (Davies, 2016), BMW (Dillet, 2016), Volvo (Volvo Car, 2016) gibi firmalar tarafından geliştirilen insansız binek araçlarının temeli atılmıştır. Günümüzde sadece binek araçlar değil birçok alanda insansız araçlar üzerine çalışılmaktadır. Örneğin Bosch'un start-up şirketi Deepfield tarafından tarım sanayii için çok amaçlı robotik platform üzerine çalışılmaktadır (Ackerman, 2015). Fabrikalarda tasarruf sağlama amacıyla Toyota (Toyota Forklifts, 2016), Egemin (Egemin Automation, 2016), Swisslog (Swisslog, 2016)'ın aralarında bulunduğu birçok firma tarafından geliştirilmiş otonom forkliftler bulunmaktadır. Ayrıca EasyMile (EasyMile, 2016) isimli firma geliştirdiği EZ10 isimli elektrikli ve insansız otobüsler üzerinde geliştirmelere devam etmektedir. Otto firması ilk uzun mesafeli testini gerçekleştirdiği insansız tırın geliştirme ve testleri üzerine çalışmaktadır (Davies, 2016). Geliştirilen bu insansız araçların algoritmaları ve yazılımı öncelikle benzetim ortamlarında test edilmektedir. Benzetim ortamından sonra gerçek ortama geçilirken yüksek maliyet ile üretilen bu araçlara uygulanmadan önce araştırma amaçlı gezgin robotlar üzerinde test edilmektedir. Böylece kontrollü bir laboratuvar ortamı kurularak gerçek ortam testlerinin ilk aşaması gerçekleştirilmektedir.

Gezgin robotlara eğitim ve araştırma sektöründe duyulan ihtiyaçlar doğrultusunda çok geniş yelpazede gezgin robot platformları geliştirilmiştir. Gezgin robotlar üretim amaçlarına göre gruplanmak istendiğinde platformun kullanımı kullanıcıya bağlı olduğundan tam olarak keskin bir ayrım elde edilememektedir. Buna rağmen robotlar sadece eğitim, sadece araştırma ve hem eğitim hem de araştırma amaçlı gezgin robotlar olarak gruplanabilmektedir. Sadece eğitim ya da sürü robotiği gibi çalışmalarında kullanılmak üzere tasarlanan robotlarda düşük maliyet büyük önem arz etmektedir. Bu amaçla geliştirilen robotlarda maliyeti düşürmek için daha küçük boyutlarda mekanik yapı ve daha az işlem kapasitesine sahip donanımlar kullanılmaktadır. E-puck (Mondada, 2015) benzeri küçük ve düşük maliyetli robotlar daha çok orta öğretim seviyesinde robotik kullanılarak yapılan eğitimler ve sürü robotiği çalışmalarında kullanılmaktadır. Geliştirilen robotlar incelendiğinde robotlar çalışma ortamları iç ortamdan dış ortama ve eğitimden araştırma amaçlı robotlara geçerken robotların ebatlarının, işlem kapasitelerinin ve bunlara

bağlı olarak maliyetinin arttığı gözlemlenmektedir. Yüksek işlem kapasitesine sahip olan PatrolBot (Adept Technology, 2011 a) ve PowerBot (Adept Technology, 2011 b) gibi robotlar ise yüksek maliyetinden dolayı eğitimden daha çok araştırma amaçlı çalışmalarda kullanılmaktadır. Düşük ve yüksek maliyet gerektiren bu iki uç nokta arasında ise eğitim ve araştırma amaçlı geliştirilen gezgin robotlar bulunmaktadır. Turtlebot (Clearpath Robotics, 2014), Pioneer 3-DX (Adept Technology, 2011 c), Youbot (Kuka Robotics, 2016) ve Freight (Fetch Robotics, 2016) eğitim ve araştırma amaçlı kullanılan gezgin robotlardan bazılarıdır. Bu robotların ülkemizde temin maliyeti oldukça yüksektir.

Bu çalışmada, uluslararası standartlarda eğitim ve araştırma amaçlı gezgin robot platformu olan Evarobot geliştirilmiştir. Eğitim amaçlı kullanılan robotların düşük maliyete sahip olması büyük önem arz etmektedir. Çalışma kapsamında geliştirilen gezgin robot platformunun mekanik, elektronik ve yazılım bileşenleri tamamen yerli kaynaklar kullanılarak üretilmesiyle maliyeti azaltmak amaçlanmıştır. Robotun mekanik ve elektronik donanımları modüler ve tekrardan konfigüre edilebilir şekilde tasarlanmıştır. Eğitim ve araştırma çalışmalarında ihtiyaç duyulan uzun süreli çalışmalara olanak sağlayacak şekilde güç tasarımı yapılmıştır. Robot üzerinde akan bütün verinin görselleştirilmesi ve robotun çevre birimlerinin hata kontrolünü gerçekleştiren diyagnostik yapısı kurulmuştur. Evarobot'un yazılım kütüphaneleri robotik çalışmalarda standartlaşmış hale gelen Robot İşletim Sistemi (ROS) ara katmanı kullanılarak geliştirilmiştir. Buna ek olarak yazılım kütüphanelerinin açık kaynak olarak internet üzerinden paylaşılmasıyla dünya çapında geliştirilebilir bir yapı kurulmuştur. 3B benzetim ortamı GAZEBO için robotun kinematik ve dinamik modeli hazırlanmıştır. Ayrıca Evarobot'un MATLAB üzerinden kontrolünün gerçekleştirilebilmesi için ilgili kütüphaneler geliştirilmiştir. Eğitim ve araştırma amaçlı kullanımlarda kolaylık sağlamak için Evarobot için deney dokümanları da hazırlanmıştır.

Takip eden bölümde, literatürde bulunan eğitim ve araştırma amaçlı geliştirilmiş gezgin robot platformları ve robotik ara katmanlar hakkında bilgi verilmektedir. Bölüm 3'te ise geliştirilen Evarobot'un mekanik modeli, matematiksel modeli, elektronik ve yazılım mimarisi anlatılmaktadır. Bölüm 4'de ise tasarlanan ve gerçeklenen gezgin robot platformu ile ilgili testlere yer verilmektedir. Bölüm 5'te ise sonuç ve öneriler sunulmaktadır.

2 LİTERATÜR ARAŞTIRMASI

Robotik eğitimi ve insansız araçlar üzerine yapılan çalışmalarda test amaçlı gezgin robot platformlarına ihtiyaç duyulmaktadır. Bu bölümde literatürde eğitim ve araştırmada yaygın olarak kullanılan gezgin robot platformları ve bu platformların farklı amaçlar için yaygın bir şekilde kullanılmasında önemli bir yere sahip olan ara katmanlar incelenmektedir.

2.1 Eğitim ve Araştırma Amaçlı Gezgin Robot Platformları

Gezgin robotlara eğitim ve araştırma sektöründe duyulan ihtiyaçlar doğrultusunda farklı özelliklerde gezgin robot platformları geliştirilmiştir. Literatür incelendiğinde boyutu küçük ve işlem kapasitesi düşük olan ve buna bağlı olarak maliyeti düşük olan robotların daha çok eğitim ve sürü robotiği çalışmalarında tercih edildiği görülmektedir. Araştırma amaçlı robotlara doğru gidilirken robotların boyutları, işlem kapasiteleri ve maliyetleri artmaktadır. Bu bölümde bahsedilecek olan gezgin robotların anlatımı yapılırken eğitimden araştırmaya doğru gidilerek aradaki farkların gözlemlenmesi amaçlanmıştır.

E-puck (Mondada vd., 2006), temelde mühendislik eğitimi için geliştirilen bir robottur (Şekil 2.1). Ayrıca düşük maliyetinden dolayı sürü robotları çalışmalarının yapılabileceği bir gezgin platformdur. E-puck'un mekanik tasarımı yapılırken küçük boyutlarda olması ve esnek bir yapıya sahip olmasına dikkat edilmiştir. Amaç eğitim amaçlı bir robot geliştirmek olduğu için düşük maliyetli bir robot geliştirmeye özen gösterilmiştir. Bu kapsamda ucuz komponentler kullanılmış ve seri üretim tekniklerine uygun bir mekanik tasarım yapılmıştır. Seri üretim sayesinde ürünün birim fiyatını düşürmek amaçlanmıştır. Robot 75mm çapta ebatlara ve üzerine konulan eklentilere bağlı bir yüksekliğe sahiptir. Robot ana gövde, led yüzüğü ve iki tekerlek olmak üzere dört plastik yapıdan oluşmaktadır. Robot, genel amaçlı ve DSP olmak üzere iki tip işlemciye sahiptir. DSP işlemci olarak Microchip dsPIC 30F6014A mikro denetleyicisi kullanılmıştır. İşlemci 8kB RAM ve 144kB flash hafızaya sahiptir. Ayrıca işlemci, GCC C

derleyicinin değiştirilmiş halini desteklemektedir. E-puck, 3 tane mikrofon, 8 tane kızılötesi algılayıcısı, 3D ivmeölçeri ve renkli CMOS kamerayı desteklemektedir. Robot üzerinde dönüş başına 1000 adım çözünürlüğe sahip olan iki tane adım motoru bulunmaktadır. Kullanıcıya bilgi verme amaçlı robot üzerinde ledler bulunmaktır. Bilgisayar ile robotun bağlantısı RS232 ya da Bluetooth üzerinden yapılabilmektedir. Ayrıca Bluetooth üzerinden yedi robot aralarında konuşabilmektedir. Robot üzerine ek donanımlar eklendiği zaman I2C üzerinden haberleşebileceği veri yolu ayrılmıştır. Webots ve Enki, E-puck'un desteklendiği sırasıyla 3D ve 2D benzetim ortamıdır. E-puck (Cianci vd., 2007), (Francesca vd., 2014) ve (Prieto vd., 2010) çalışmalarında da görüleceği üzere daha çok sürü robotların üzerinde yapılan araştırma ve geliştirme çalışmalarında kullanılmaktadır.

Şekil 2.1 E-puck robot görseli

Turtlebot 2 (Clearpath Robotics, 2014), eğitim ve araştırma amaçlı geliştirilmiş dünyada en popüler açık kaynak robotlarından biridir (Şekil 2.2). Robot 35,4 cm çapında ve 42 cm yüksekliğindedir. Robot 6,3 kg ağırlıkta olup 5 kg taşıma kapasitesine sahiptir. Diferansiyel sürüşe sahip olan robot 0,65 m/s maksimum hıza ulaşabilmektedir. Turtlebot temel olarak Kobuki robotunu kullanmaktadır. Bünyesinde bir adet çift çekirdekli netbook bilgisayar, Kobuki temeli, jiroskop ve ASUS Xtion Pro derinlik kamerası içermektedir. Robot ile haberleşme seri port üzerinden gerçekleşmektedir. Robot üzerinde bulunan tam programlanabilir butonlar üzerine istenilen komut girilebilmektedir. Robot üzerinde kızılötesi algılayıcı, cayroskop ve bumper bulunmaktadır. TurtleBot, açık kaynak kodlu ROS tabanlı uygulamalar geliştrilmesine olanak tanımaktadır. TurtleBot SDK, TurtleBot ile ilgili ihtiyaç olunabilecek her türlü yazılım ile etkileşim halindedir. ROS'un yanı sıra

OpenCV ve PCL kütüphaneleri ile de TurtleBot üzerinde uygulamalar geliştirilebilmektedir. Universitat Jaume I, Özyeğin Üniversitesi, University of Southern California, University of Washington laboratuvarlarında eğitim ve araştırma amaçlı Turtlebot'u kullanan üniversitelerden bazılarıdır. (Gritti vd., 2014), (Wu vd., 2015) ve (Barber vd., 2015) çalışmaları Turtlebot üzerinde yapılan araştırma çalışmalardan birkaç örnektir.

Şekil 2.2 Turtlebot 2 görseli

Robotino (Festo, 2016), Festo-Didactic tarafından eğitim ve araştırma amaçlı geliştirilmiş gezgin robot platformudur (Şekil 2.3). 45 cm çapta ve 29 cm yükseklikte olan ve paslanmaz çelikten üretilen Robotino, üç adet omnidirectional sürüşe sahip tekerlere sahiptir. Robot 20 kg ağırlıkta olup 30 kg taşıma kapasitesine sahiptir. İki adet 12V batarya ile çalışan robotun dört saate kadar çalışma süresi bulunmaktadır. Robotun motorlarını sürmek için 32 bit mikro denetleyici kullanılmıştır. Robot üzerindeki enkoderleri okumak ve işlemek için bir adet FPGA kullanılmıştır. Mikro denetleyici ile haberleşen ve robotun kontrol yazılımlarının bulunduğu bütünleşik bir bilgisayar bulunmaktadır. Bu bilgisayar için Intel ya da Atom işlemcili olmak üzere iki seçenekten birisi kullanılabilmektedir. Robot üzerine kızılötesi mesafe algılayıcısı, endüktif yakınlık algılayıcısı ve optik algılayıcı takılabilmektedir. Robot üzerine eklenecek ek bileşenler için USB, Ethernet, PCI

Express girişleri de bulunmaktadır. Robot ile haberleşme WiFi üzerinden gerçekleştirilmektedir. Robot üzerinde geliştirme yapılacağı zaman C/C++, JAVA ve .NET dilleri kullanılarak yapılabilmektedir. LabVIEW ve MATLAB/Simulink ortamında geliştirmeler yapılabilmektedir. Ayrıca Robotino, ROS uyumludur. Robotino üzerindeki mikro denetleyici için yazılım geliştirmekte mümkündür. Robotino-SIM ile robotun 3D simülasyonu üzerinde çalışılabilmektedir. Robotino eğitim ve araştırma amaçlı birçok çalışmada kullanılmaktadır. Türkiye'de Niğde Üniversitesi ve Erciyes Üniversitesi robotik laboratuvarlarında robotik eğitiminde kullanılmaktadır. Robotino ile Güney Afrika'da yapılan mühendislik eğitiminde ve yarışmalarda kullanılmıştır (Weinert ve Pensky, 2011). Ayrıca, (Straßberger vd., 2014) çalışması Robotino ile yapılan araştırma çalışmalarına örnektir.

Şekil 2.3 Robotino görseli

X80SV (Dr Robot, 2013), DrRobot tarafından uzaktan görüntüleme, uzaktan konferans, otonom navigasyon gibi gelişmiş robotik uygulamalarında araştırma ve geliştirme için gezgin robot platformudur (Şekil 2.4). Robot, 38 cm çapında ve 25,5 cm yüksekliğinde ebatlara sahiptir. 18 cm çapında iki adet diferansiyel sürüş tekerine sahiptir. 22 kg.cm tork sağlayabilen motorlar ile maksimum 1,0 m/s doğrusal hıza ulaşabilmektedir. Robotun ağırlığı 3,5 kg'dır ve 10 kg yük taşıma kapasitesine sahiptir. Robot üzerinde dijital sinyal işleme için bir mikro denetleyici bulunmaktadır. Bu mikro denetleyici ile

WiFi üzerinden tanımlanmış bir protokol ile haberleşilmektedir. Bu haberleşme protokolü üzerinden algılayıcılardan okunan bilgiler ve motorları kontrol etmek için gereken ilgili bilgiler akmaktadır. Tanımlanmış olan protokol üzerinden haberleşme gerçekleştiğinden herhangi bir yazılım platformuna bağlı kalınmadan robot üzerinden geliştirme yapmak mümkündür. Bu protokol için hazırlanmış bir SDK'sı bulunmaktadır. Robotun kontrolü için harici bilgisayar kullanılması gerekmektedir. Robot üzerinde 128x64 grafik LCD bulunmaktadır. Sonar algılayıcı, kızılötesi algılayıcı, sıcaklık algılayıcısı, lazer mesafe algılayıcısı, kamera gibi birçok algılayıcıyı robot desteklemektedir. (Chang vd., 2012) X80SV ile yapılmış araştırma çalışmasına bir örnektir.

Şekil 2.4 X80SV robot görseli

Pioneer 3-DX (Adept Technology, 2011 c), Adept Mobile Robots grubu tarafından geliştirilmiş dünyanın en popüler gezgin araştırma robotudur (Şekil 2.5). Robot 45,5 cm uzunlukta, 38,1 cm genişliğindedir. Temel platformun yüksekliği 23,7 cm'dir ve bu yükseklik robot üzerinde kullanılan lazer, kamera ya da robot kolu gibi donanımlara bağlı olarak değişmektedir. Robotun mekaniği 1,6 mm kalınlıkta alüminyum gövdeden oluşmaktadır. Temel platformun ağırlığı 9 kg iken 23 kg'a kadar taşıma kapasitesine sahiptir. İki tane sürücü dolma tekere ve diferansiyel sürüşe sahip olan robotun ön kısmında bir adet sarhoş teker bulunmaktadır. Bu tasarımdan kaynaklı ani frenlemelerde devrilme riski taşımaktadır. Güçlü motorlara sahip olan Pioneer 3-DX'in maksimum doğrusal hızı 1,2 m/s'dir. Robot üzerinde bulunan mikro denetleyicide üretici firma tarafından geliştirilmiş olan ARCOS isimli yerleşik donanım yazılımı bulunmaktadır. Mikro denetleyici 32 dijital girdiye, 8 dijital çıktıya ve 7 tane analog girdiye sahiptir.

Düşük seviyede robot üzerinde ARCOS çalışırken, yüksek seviyede robot üzerinden verilerin alınması ve kontrol edilmesi için geliştirilmiş olan ARIA isimli yazılım çerçevesi bulunmaktadır. Pioneer 3-DX'in ROS (Robot İşletim Sistemi) ile haberleşebilmesi için ilgili sürücüler mevcut olup, ROS kütüphanelerinin robot üzerinde kullanılmasına olarak sağlamaktadır. Ayrıca Gazebo benzetim ortamında robotun modeli bulunmaktadır. Robotun simülasyonun gerçekleştirilmesi için açık kaynaklı benzetim ortamı olan MobileSim'de robotun modeli hazırlanmıştır. Robot temel setinde 8 tane sonar algılayıcı, enkoderler ve mikro denetleyici içermektedir. Bunlara ekstra olarak robot üzerine dual core 2,26 GHz işlemcili bilgisayar da eklenebilmektedir. Ayrıca opsiyonel olarak, lazer algılayıcı, kamera, arka sonar algılayıcılar, jiroskop, bumper, mikrofon gibi algılayıcılarda eklenebilmektedir. Türkiye ve yurt dışındaki birçok üniversite laboratuvarlarında bu robotlardan bulunmakta ve hem eğitim hem de araştırma amaçlı kullanılmaktadırlar. Eskişehir Osmangazi Üniversitesi, İstanbul Şehir Üniversitesi, Kocaeli Üniversitesi, İstanbul Teknik Üniversitesi ülkemizde Pioneer 3-DX'i kullanan üniversitelerden bazılarıdır. Pioneer 3-DX robotik üzerinde yapılan birçok araştırma çalışmalarında gezgin platform olarak kullanılmıştır (Yazici vd., 2014; Chang vd., 2007; Guzel ve Bicker, 2012).

Şekil 2.5 Pioneer 3-DX görseli

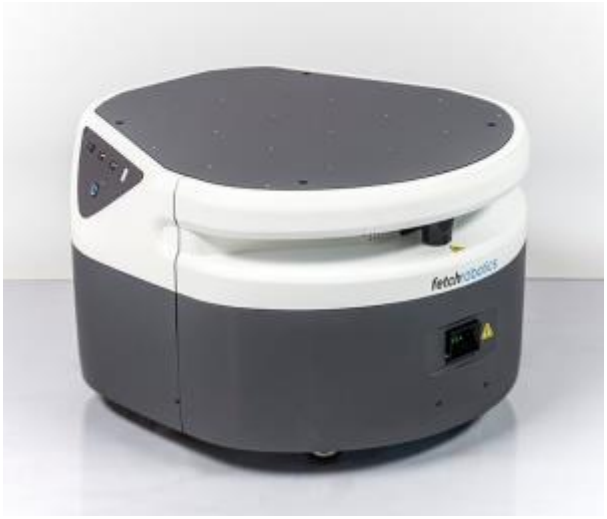
Youbot (Kuka Robotics, 2016), Kuka Robotics tarafından eğitim ve araştırma amaçlı bir gezgin platformdur (Şekil 2.6). Robot gezgin platform olan temel platform ve üzerinde bulunan robot kolu olmak üzere iki platformdan oluşmaktadır. Temel platformun ebatları 58x38x14 cm'dir. Robot 20 kg ve 20 kg taşıma kapasitesine sahiptir. Omni-

directional sürüşe sahip dört tekerlek ile kontrol edilen gezgin platform maksimum hız olarak 0,8 m/s'ye ulaşabilmektedir. Robot içerisinde bütünleşik atom işlemcili mini bilgisayar kullanılmıştır. Platform ile haberleşme EtherCAT üzerinden gerçekleşmektedir. Bunun için ilgili sürücüler KUKA youBot API altında geliştirilmiştir. YouBot, ROS uyumlu bir platformdur ve Gazebo, OpenRAVE, V-REP ve Webots benzetim ortamlarında kullanılabilmektedir. Ayrıca OROCOS ve Labview ile de robotun kontrolü gerçekleştirilebilmektedir. Robot üzerinde 3D Kamera, lazer algılayıcı, kızılötesi algılayıcı ve renkli kamera gibi algılayıcıları desteklemektedir. Literatürde YouBot'un eğitim (Bischoff vd., 2011; Duc vd., 2012) ve araştırma (Systems, 2014; Lopez-Franco vd., 2014) amaçlı kullanıldığı çalışmalar bulunmaktadır. Ayrıca YouBot Türkiye'de Bahçeşehir Üniversitesi robotik laboratuvarında eğitim ve araştırmalarda kullanılmaktadır.

Şekil 2.6 YouBot görseli

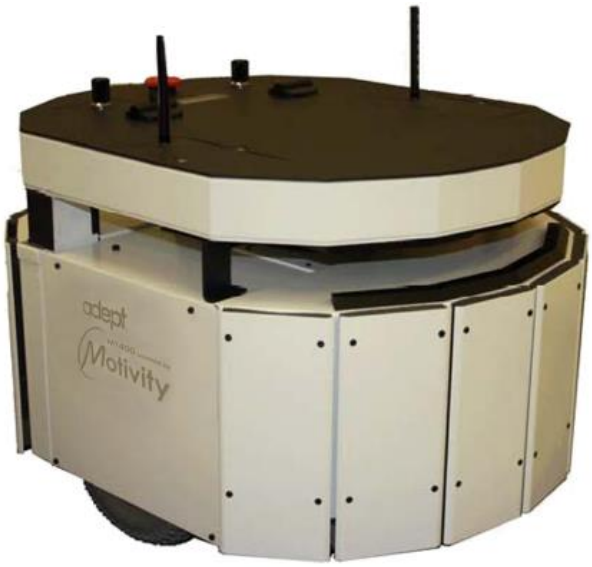
Freight (Fetch Robotics, 2016), Fetch Robotics tarafından lojistik amaçlı yapılan çalışmalarda kullanılması geliştirilen otonom gezgin platformdur (Şekil 2.7). 68 kg ağırlığa sahip olan robotun 100 kg'a kadar taşıma kapasitesi bulunmaktadır. Robot 50.8x55.9x35.9 cm ebatlarındadır. Maksimum doğrusal hızı 2,0 m/s'dir. 9 saat gibi uzun bir nominal çalışma süresine sahiptir. Robot içerisinde algılayıcılardan alınan bilgileri işlemek ve robotun kontrolü için Intel işlemcili güçlü bir bilgisayar kullanılmıştır. Robot üzerinde 2D lazer algılayıcı, 3D derinlik algılayıcısı bulunmaktadır. Robot ile haberleşme WiFi ya da Ethernet üzerinden gerçekleştirilebilmektedir. Ayrıca HDMI ve iki tane USB bağlantısı üzerinden direk olarak robot bilgisayarı üzerinden de geliştirmeler yapılabilmektedir.

Robot şimdilik yüksek maliyetinden kaynaklı olarak eğitimden daha çok araştırma çalışmalarında kullanılmaktadır. Freight, ROS uyumlu bir gezgin robot platformudur. Robotun otonom navigasyonu, ortam haritasının çıkartılması gibi çalışmalarda ROS kütüphanelerinin kullanılabilmesi için ilgili sürücüler geliştirilmiştir.



Şekil 2.7 Freight görseli

PatrolBot (Adept Technology, 2011 a) ‘da Adept Mobile Robots grubu tarafından geliştirilmiş olan, diferansiyel sürücülü, orta dereceli yükleri taşıyabilen ve 7/24 çalışabilen bir araştırma robotudur (Şekil 2.8). Robotun ebatları 48x59x38 cm ‘dir. PatrolBot 2,3 mm kalınlıkta alimünyüm kasadan oluşmaktadır IP42 standartlarını desteklemektedir. Robotun toplam ağırlığı 45 kg’dır ve düz zeminde 40 kg’lık taşıma kapasitesine sahiptir. Differensiyel sürüşe sahip olan PatrolBot üzerinde iki tane 20 cm çapında sürücü tekere ve dört tane dengeleyici sarhoş teker bulunmaktadır. Sürücü tekerlerin ortada bulunması ve sarhoş tekerlerin ön ve arka tarafta bulunması ile Pioneer 3-DX bulunan denge problemine karşı tedbir alınmıştır. İç ortamlar için tasarlanmış robotun maksimum doğrusal hızı 1,8 m/s’dir. Pioneer 3-DX’deki gibi PatrolBot üzerinde ARCOS yazılımı içeren mikro denetleyici bulunmakta ve opsiyonel olarak intel işlemcili bilgisayar da entegre edilebilmektedir. Düşük seviye donanım ile haberleşme RS232 üzerinden yapılmaktadır. Yazılım platformu olarak ARIA ve ROS uyumludur ve MobileSim ve Gazebo ortamlarında simüle edilebilmektedir. PatrolBot, eğitimden daha çok araştırma ve geliştirme projelerinde kullanılmaktadır (Ray vd., 2008).



Şekil 2.8 PatrolBot görseli

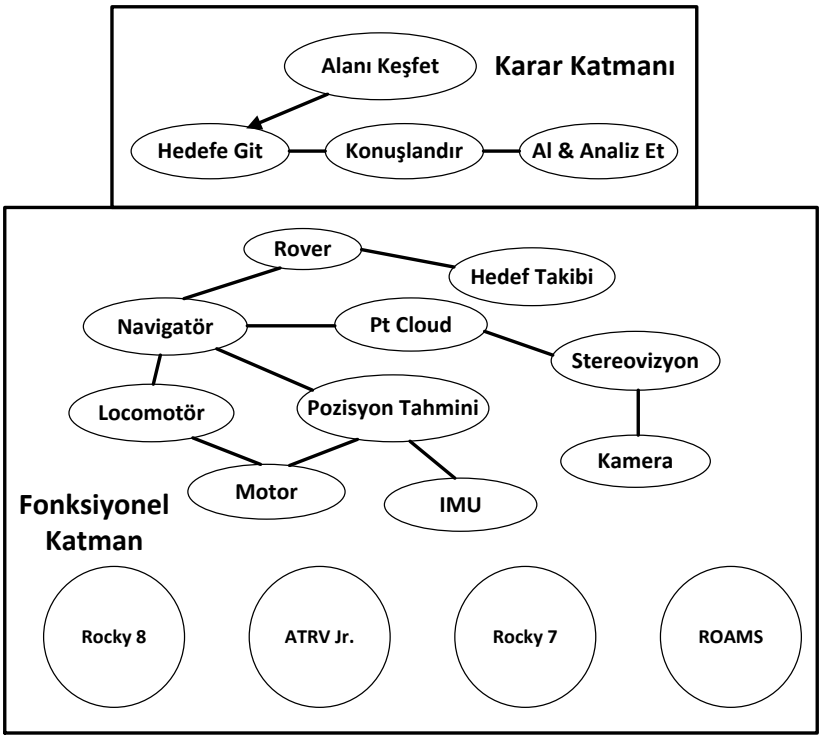
PowerBot (Adept Technology, 2011 b), Adept Mobile Robots tarafından geliştirmiş olan ağır yükler taşıyabilecek araştırma ve hızlı prototip robotudur (Şekil 2.9). Robotun üst platformu 4.75 mm kalınlıkta ve diğer kısımlar ise 2,3 mm kalınlıkta alüminyum ile üretilmiştir. Robot 62,6 cm genişlikte, 83,2 cm uzunlukta ve 48,4 cm yükseklikte ve 120 kg ağırlığa sahip iken 100 kg taşıma kapasitesine sahiptir. Bu robotta da sürücü tekerler ortada bulunmakta ve iki tane dengeleyici sarhoş teker bulunmaktadır. İç ortamlar için tasarlanmış olan PowerBot 2,1 m/s doğrusal hıza ulaşabilmektedir. Diğer Adept Mobile Robots tarafından geliştirilen robotlardaki mikro denetleyici, yazılım mimarisi ve ek algılayıcılara sahiptir. Diğer platformlardan farkı mekanik farkıdır. Robot yüksek yük taşıma kapasitesine sahip olmasından kaynaklı taşıma ve navigasyon görevleri için ideal bir robottur. Büyük bir mekanik platforma sahip olan robot, ayrıca yüksek fiyata sahiptir. Bu yüzden eğitimden daha çok araştırma amaçlı kullanılan gezgin platformdur. (Karasalo vd., 2009) ve (Abdessemed vd., 2014) çalışmaları PowerBot kullanılarak yapılan çalışmalardan bazılarıdır.

Şekil 2.9 PowerBot görseli

2.2 Robotik Ara Katmanlar

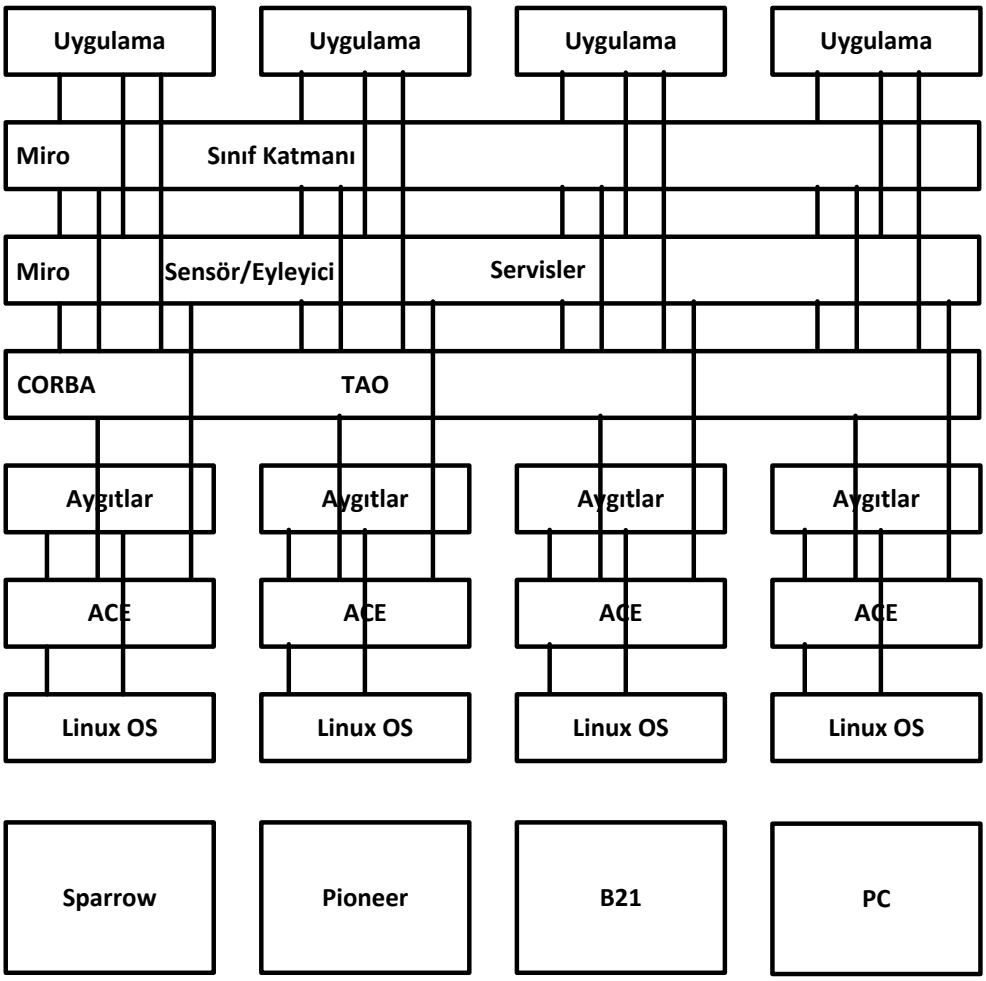
Robotik sistemlerde uygulama alanına bağlı farklı çevre birimleri kullanılmaktadır. Bu birimlerin ürettikleri bilgiler ve haberleşme yöntemleri arasında da farklılıklar bulunmaktadır. Donanım ve yazılımının işletim sistemlerindeki gibi soyutlanarak geliştiricilere kolaylık sağlanması adına birçok robotik ara katman geliştirilmiştir. Bu ara katmanlar her bir birim için yeni bir yazılım geliştirilmesi yerine nesne tabanlı bir yapıda kolay geliştirilebilir bir yapı sunmayı amaçlamaktadır. Bu bölümde literatürde robotik çalışmalarında yaygın olarak kullanılan ara katmalar incelenecektir.

CLARAty (Nesnas vd., 2003), fonksiyonel katman ve karar katmanı olmak üzere iki katmandan oluşmaktadır (Şekil 2.10). Fonksiyonel katman, gerçek ya da simule edilmiş robota uyum sağlamak için gerekli olan temel işlevselliği sağlamaktadır. Bu katman düşük ve orta seviyede otonomluk içerebilmektedir. Algılayıcıların okunması, motorların kontrolü, algılayıcı tabanlı kontrol gibi görevler Fonksiyonel katmana aittir. Karar katmanı, planlama ve çalıştırma sistemlerinin birleşmesinden oluşmaktadır. Bu katmanda sistem kaynakları, istenilen hedef ve sistemin durumunu değerlendirerek global çözümler üretilmektedir. Zaman çizelgelemesi, global planların çıkarılması gibi işlemler Karar katmanın gerçekleştirilmektedir.



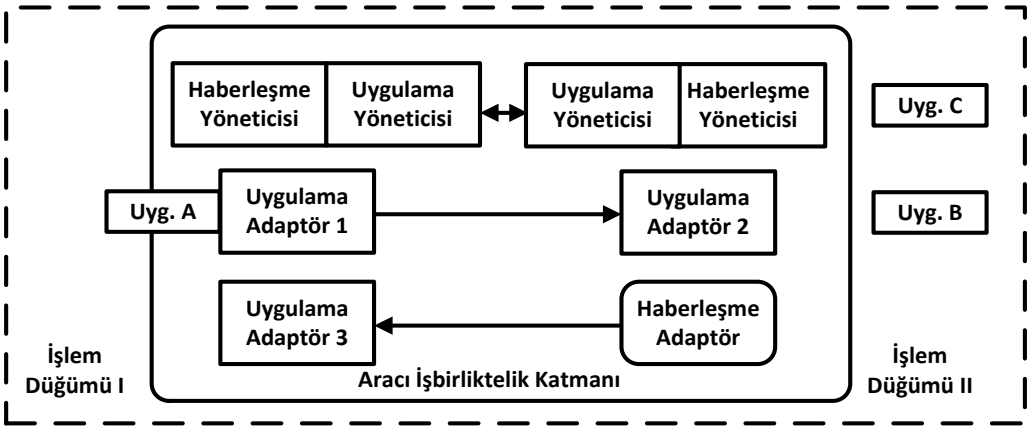
Şekil 2.10 CLARAty katmanları

Miro (Utz vd., 2002), dağınık robot kontrolünü sağmak için haberleşme alt yapısında CORBA standartlarına bağlı kalınarak geliştirilen bir robotik ara katmanıdır. Miro üç tane mimari katman üzerine oturtulmuş bir yapıya sahiptir (Şekil 2.11). Miro Aygıt Katmanı robot üzerindeki algılayıcı ve yürütücüleri nesne tabanlı bir yapıya büründürerek ara katmanı platformdan bağımsız hale getiren katmandır. Aygıt Katmanının üzerinde ise Miro Servis Katmanı bulunmaktadır. Servis Katmanı, Aygıt Katmanındaki algılayıcı ve yürütücü bilgilerini CORBA haberleşme ara yüzüne dönüştüren katmandır. Bu sayede herhangi bir aygıta yerel ya da uzaktan erişeme imkân verilmektedir. Bu katmandaki servisler klasik tabanlı haberleşme yerine olay tabanlı haberleşme ile çalışmaktadır. Miro'nun üçüncü katmanı ise Sınıf Katmanıdır. Bu katman robotun kontrolü, konumlandırma, yol planlama, görselleştirme gibi fonksiyonel modüller sağlamaktadır. Bu ara katman Linux ve Windows ortamında çalışılmasına imkân vermektedir ve gerçek zamanlı çalışmamaktadır.



Şekil 2.11 Miro mimarisi

Marie (Côté vd., 2006), merkezi kontrol modeline sahip ara katmandır. Heterojen bileşenleri kullanarak dağınık uygulamalar geliştirilmesi için Aracı Kalıbı (Mediator Pattern) kullanılmıştır. Böylece bileşenler birbirleri ile bağımsız olarak iletişime geçebilmektedir. Ara katman, Çekirdek, Bileşen ve Uygulama olmak üzere üç katmandan oluşmaktadır (Şekil 2.12). Çekirdek katmanında girdi/çıktı kontrolü, hafıza işlemleri gibi düşük seviyeli işletim sistemi fonksiyonları, haberleşme v.b. işlemlerin gerçekleştiği katmandır. Bileşenleri eklenmesi için gerekli yazılım çerçevelerinin tanımlandığı ve uygulandığı yer ise Bileşen katmanıdır. Geliştirilen uygulamaların çalıştırılması ve kontrolünü üstlenen katman Uygulama katmanıdır. Marie katmanı gerçek zamanlı çalışmamaktadır ve Windows işletim sistemi için desteği bulunmamaktadır.



Şekil 2.12 Marie uygulama tasarım çerçevesi

OpenRTM-aist (Ando vd., 2008), CORBA üzerine geliştirilmiş gerçek zamanlı çalışan bir robotik ara katmandır. Dağınık kontrolü destelemektedir. Ara katmanın amacı yazılım seviyesinde modüler bir yapıda robotu ve fonksiyonel kısımlarını geliştirmek ve seçilen modülleri basit bir şekilde birleştirerek robot geliştirme işlemini kolaylaştırmaktır. RT-Komponent yapısı kullanılarak çeşitli amaçlar için farklı konfigürasyonlarda maliyet etkin robotlar geliştirilmesine olanak sağlanmaktadır. RT-Komponentler C++, Pyhton ve Java dilleri ile geliştirilebilmektedir. OpenRTM-aist, açık kaynak kodlu ve Linux ve Windows ortamında çalışabilen bir ara katmandır.

Orocos (Bruyninckx, 2001), genel amaçlı ve açık kaynaklı robot kontrol yazılımlarını içermektedir. Orocos yazılım kütüphaneleri gelişmiş teknik dokümantasyona sahiptir. Orocos'taki amaç ticari robot platformlarından bağımsız çalışan ve açık kaynaklı olması sayesinde diğer kullanıcılar tarafından da geliştirilebilecek robotik kütüphanelerinin oluşturulmasıdır. Orocos temelde dört tane c++ kütüphanesinden oluşmaktadır: RTT (The Orocos Real-Time Toolkit), OCL (The Orocos Components Library), KDL(The Orocos Kinematics and Dynamics Library) ve BFL (The Orocos Bayesian Filtering Library). RTT kütüphanesi, geliştiricilere konfigüre edilebilir ve etkileşimli komponent tabanlı gerçek zamanlı kontrol uygulamaları geliştirmesine olanak sunmaktadır (Bruyninckx ve Soetens, 2007). OCL kullanılma hazır komponent sunmaktadır. Bütün bileşenler RTT üzerinde geliştirilmiştir ve bazıları KDL ya da BFL kütüphanelerini kullanmaktadır. KDL ise modelleme ve kinematik zincirleri gerçek zaman hesaplama gibi uygulamalar içermektedir. BFL kütüphaneleri Bayes kuralını temel alan Kalman, Partikül gibi filtreleri içermektedir.

Player (Kranz vd., 2006), gezgin robot uygulamaları için altyapı, sürücü ve bazı algoritmaları sağlayan bir ara katmandır. Yürütücüler, algılayıcılar ve robotlar Player'da aygıt olarak tanımlanmaktadır. Player'daki aygıtlar sürücü ve ara yüzün birleşmesinden oluşmaktadır. Arayüz, geliştiricinin algılayıcılardan bilgilerin alınması, işlenmesi ve kontrolünü gerçekleştireceği uygulamaları yazacağı kısımdır. Sürücü ise donanım ile yazılım arasında geçişi sağlamaktadır. Aygıtlardan bilgilerin alınması ya da onlara bilgilerin gönderilmesi için aradaki haberleşmeyi sağlamak sürücünün görevidir. Collett v.d. (2005) çalışmasında kuyruk tabanlı haberleşme yapısına sahip olan bu katmanın haberleşmesinin çekirdek ve taşıyıcı olmak üzere iki kısımdan oluştuğundan bahsetmektedir. Çekirdek katmanında mesaj sözdizimleri tanımlanmakta ve mesajların geçiş koordinasyonu sağlanmaktadır. Taşıyıcı katmanı ise aygıttan bağımsızdır ve soketler üzerinden TCP haberleşme protokolü tabanlıdır.

Robot İşletim Sistemi (ROS) (Quigley vd., 2009), robotlar için geliştirilmiş açık kaynak kodlu meta işletim sistemidir. Bu tez çalışmasında da tercih edilen bu sistem takip eden alt bölümde detaylı verilmektedir.

2.3 Robot İşletim Sistemi (ROS)

ROS, bir işletim sisteminden beklenen donanımdan soyutlama, düşük seviyeli aygıt kontrolü, yaygın olarak kullanılan işlevselliği gerçekleştirme, prosesler arasında mesajlaşma ve paket yönetimi işlemlerini gerçekleştirebilmektedir. ROS'un haberleşme topolojisi peer-to-peer ağ yapısına dayanmaktadır. Böylece heterojen bir ağ ile birbirine bağlı olan bilgisayarların haberleşmesinde ortaya çıkan yavaşlıktan kaynaklı veri kayıpları önlenmektedir. Peer-to-peer haberleşmesinin ihtiyaç duyduğu eşlerin ve birbirleriyle ilişkilerinin tutulduğu tablo için ROS'ta XML-RPC'i içeren master olarak isimlendirilen yapı kullanılmaktadır. ROS açık kaynaklı diğer projelerden birçok kodu tekrardan kullanabilmektedir. Bunlara örnek olarak Project projesinden sürücüler, navigasyon sistemi, simulator, OpenCV'den görüntü işleme, Orocos'dan Bayesian filtreleri verilebilir. ROS yapı olarak düğüm, mesaj, topik ve servis olmak üzere dört ana konseptten oluşmaktadır. Düğüm, proseslerin gerçekleştirildiği yerdir. Bu düğümler arasındaki haberleşme mesajlar üzerinden gerçekleşmektedir. Mesajlar topikler üzerinden akmaktadır. Topikler publish/subscribe yapısında haberleşmektedir. Bundan dolayı topikler asenkron

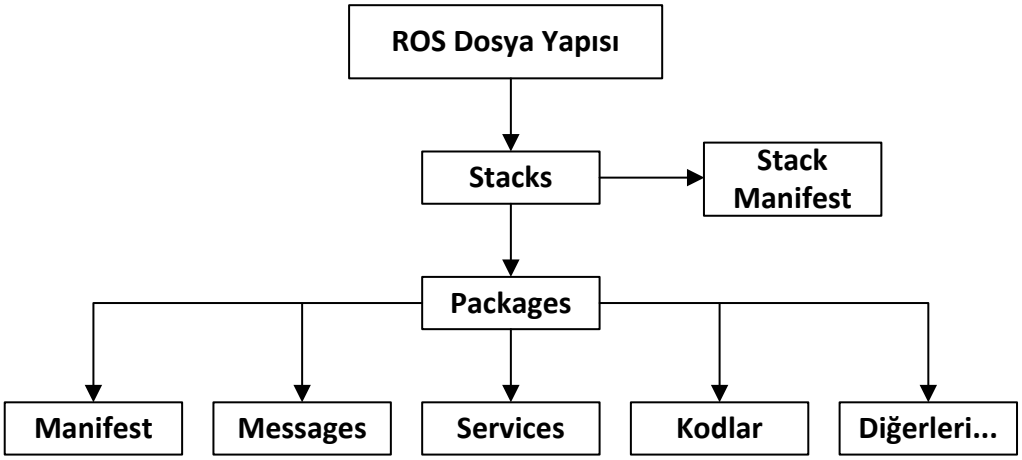
haberleşme sağlamaktadır. Senkron bir haberleşme için ise topiklerin yerine servisler kullanılmaktadır. ROS piyasada bulunan birçok robot ve robotik çalışmalarında aktif olarak kullanılmaktadır. Açık kaynak kodlu olması ve yaygın olarak kullanılmasından dolayı artık standartlaşma seviyesine ulaşmıştır. Yeni üretilen herhangi bir robot ya da algılayıcıların ROS sürücüleri ile birlikte piyasaya sürülecek kadar robotik çalışmalarında yaygın olarak kullanılmaktadır. Evarobot'un kontrol yazılımları tamamen ROS uyumlu olacak şekilde geliştirilmiştir. İlerleyen bölümlerde bahsedilecek olan kontrol yazılımlarının kolay anlaşılması adına bu bölümde ROS'un mimarisinden ve bir robotun ROS ortamı için modelinin oluşturulmasından kısaca bahsedilecektir.

2.3.1 ROS mimarisi

ROS ortamında çalışan prosesler arasındaki haberleşme peer-to-peer yapısında gerçekleşmektedir. Prosesler arasındaki haberleşme asenkron ve senkron olarak iki tipte gerçekleştirilebilmektedir. Anlık olarak işlemler arasındaki mesajlar görüntülenebilmekte ve dışardan müdahale edilebilmektedir. ROS'un sağladığı bu avantajlardan faydalanılabilmesi için ROS'un dosya ve işlevsel yapısı olarak iki grupta incelenen ROS mimarisi hakkında detaylı bilgiye sahip olunması gerekmektedir.

2.3.1.1 ROS dosya yapısı

ROS'un dosya yapısında ana yapıyı paketler oluşturmaktadır. Paketler birleşerek yığınları oluşturmaktadır. Paketlerin de kendisine özel alt klasörleri ve dosyaları bulunmaktadır. **Şekil 2.13**'te ROS'un standart dosya yapısı verilmiştir.



Şekil 2.13 ROS dosya yapısı

Paketler (Packages): Paketler ROS'un temel yapısını oluşturmaktadır. ROS ile birlikte bir program geliştirmek istendiğinde

- bin/: Derleme sonun elde edilen dosyaların tutulduğu klasördür.
- include/paket_adı/: Paket içerisinde ihtiyaç duyulan kütüphanelerin header dosyalarının bulunduğu klasördür.
- msg/: Standart mesaj tiplerinin dışında tanımlanan mesaj tiplerinin tutulduğu klasördür.
- scripts/: Bash, Python gibi çalıştırılabilir script kodlarının içermektedir.
- src/: Paket içerisinde geliştirilen kaynak kodların bulunduğu klasördür.
- srv/: Standart servislerin dışında tanımlanan servis tiplerini içermektedir.
- CMakeLists.txt: CMake derleme dosyasıdır.
- manifest.xml: Paket hakkındaki bilgilerin tutulduğu dosyadır.

Manifesto (Manifests): Bir paket hakkında lisans bilgileri, bağımlılıklar, derleyici bayrakları gibi bilgileri içermektedir. Manifestolar, manifests.xml isimli dosyalar tarafından kontrol edilmektedir.

Yığınlar (Stacks): Aynı amaca sahip birden fazla paketin bir araya gelmesiyle yığınlar oluşmaktadır. ROS bünyesinde navigasyon yığını gibi birçok yığın bulunmaktadır.

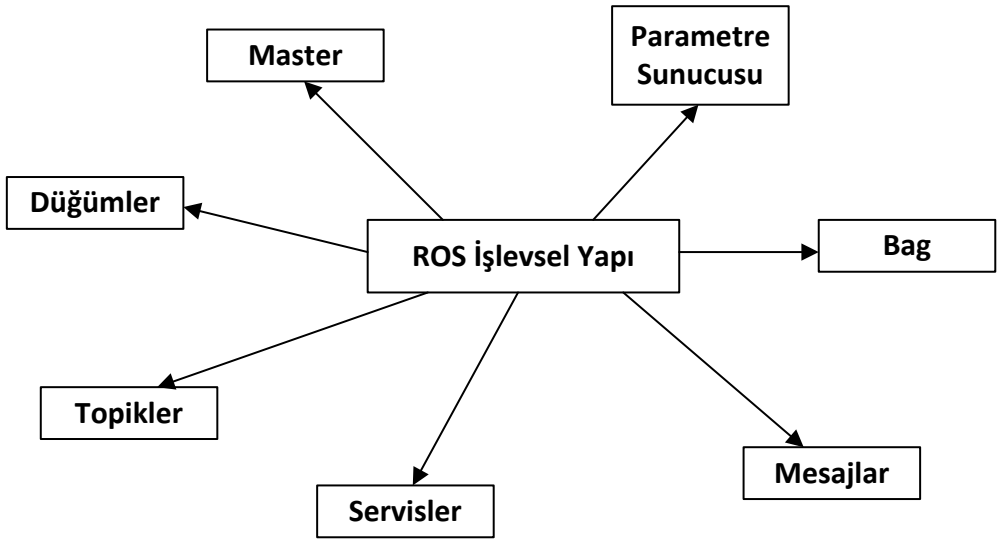
Yığın Manifestoları (Stack Manifests): Manifestoların paketler hakkında içerdiği bilgiler gibi yığınların lisans, bağımlılık gibi bilgilerinin içermektedirler. Stacks.xml isimli dosyada bu bilgiler saklanmaktadır.

Mesaj (msg) Tipleri: Düğümler arasındaki haberleşmeler dah önceden tanımlanmış mesaj tipleri ile yapılmaktadır. ROS bünyesinde standart olarak tanımlanmış birçok mesaj tipleri bulunmaktadır. Bu mesaj tiplerinin dışında bir mesaj tanımlanacağı zaman paket içerinde msg klasörünün altında .msg uzantılı mesaj tipleri tanımlanabilmektedir.

Servis (srv) Tipleri: ROS üzerindeki servislerin haberleşmesi için kullanılan servis tipleridir. Mesaj tiplerinde olduğu gibi standart olarak tanımlanmış servis tipleri bulunmaktadır. Yeni tanımlanacak mesaj tipleri paket içerisinde srv klasörünün altında .srv uzantılı olacak şekilde tanımlanabilmektedir.

2.3.1.2 ROS'un işlevsel yapısı

ROS'taki birimler arasındaki haberleşme ve bunların kontrolünü sağlayan yapısı **Şekil 2.14**'te verilmiştir.



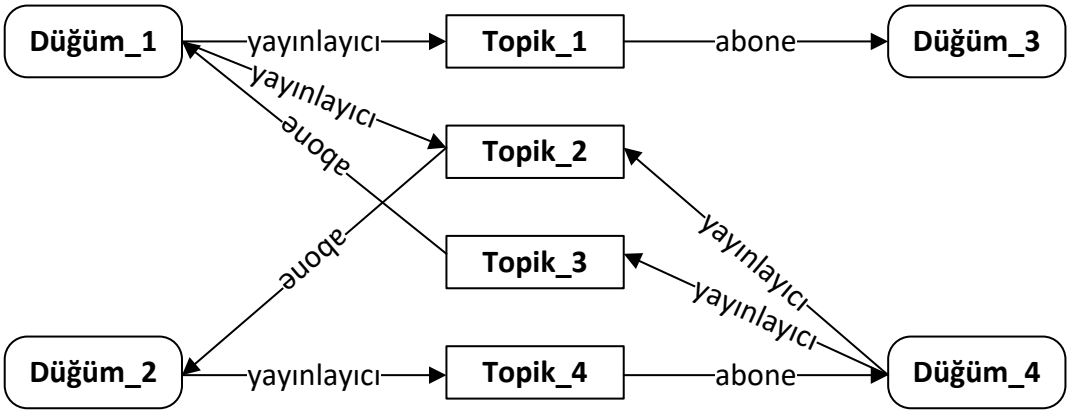
Şekil 2.14 ROS işlevsel yapısı

Düğümler (Nodes): Düğümler hesaplamaların gerçekleştirildiği proseslerdir. Düğümler kullanılarak geliştirilen sistemler sayesinde modüler bir yazılım mimarisi elde edilmektedir. Farklı fonksiyonların kontrolü birçok düğüm ile gerçeklendiğinde, sistemdeki her şeyi yapan bir düğüme göre daha fonksiyonel bir yapı oluşturulmaktadır. Düğümler arasındaki haberleşme ROS ağı kullanılarak sağlanmaktadır.

Master: ROS için isim ve kayıt servisidir. Master, ROS ortamındaki düğümlerin port bilgileri, haberleşmedeki mesaj bilgileri gibi bilgileri bünyesinde barındırmaktadır. Master olmaz ise düğümler birbirlerini bulamamakta, mesaj alışverişi yapamamakta ya da servisleri çağıramamaktadır.

Parametre Sunucusu (Parameter Server): Düğümlerin parametrelerini merkezi bir ortamda tutan ROS servisidir. ROS Master içerisinde çalışmaktadır. Düğümler çalışırken parametre değişikliği yapılması da parametre sunucu sayesinde gerçekleştirilmektedir.

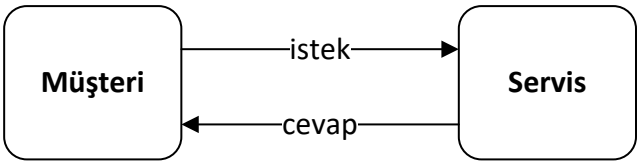
Mesajlar: Düğümlerin birbirleri arasındaki haberleşme mesajlar üzerinden gerçekleşmektedir. Bir mesaj diğer düğümlere gönderilecek bilgileri içermektedir. ROS içerisinde standart olarak tanımlanmış mesaj tipleri bulunmaktadır. Buna ek olarak, özel mesaj tipi de tanımlanabilmektedir.



Şekil 2.15 ROS düğüm ve topik ilişkisi

Topikler: ROS ortamında düğümler arasındaki mesajlar yayınla/abone ol yapısında gerçekleşmektedir. Topik, mesajın içeriğini tanımlamak için kullanılan isimdir. Bir düğüm

mesaj göndereceği zaman topiğe mesaj yayınlamakta, okuyacağı zaman ise topiğe abone olmaktadır. Bir topiğin birden fazla yayınlayıcı ya da abone düğümü olabilirken, bir düğüm birden fazla topiğe yayın yapabilmekte veya abone olabilmektedir (**Şekil 2.15**). Genelde yayınlayıcı ve abone düğümler iletişimde oldukları düğüm hakkında bilgileri olmamaktadır.



Şekil 2.16 ROS'ta servis yapısı

Servisler: Yayınla/abone ol modeli esnek bir haberleşme paradigmasıdır ve dağıtık sistemlere ihtiyaç duymaktadır. İki düğüm arasında istek/cevap şeklinde senkron bir yapı kurulmak istendiğinde servisler kullanılmaktadır. Yayınla/abone ol modelinin aksine servislerde bir istek ve bu isteğe karşı dönen bir cevap bulunmaktadır (**Şekil 2.16**). Bu yapıda müşteri servise istek göndermekte ve servis cevap dönene kadar müşteri beklemektedir.

Bags: ROS mesaj bilgilerinin kaydedilmesi ve tekrardan oynatılması sağlayan bir formattır. Geliştirme ve test aşamasında gerekli olan algılayıcı bilgileri gibi bilgilerin kaydedilmesi ve gerçek zamanlı kullanılmasını imkân veren mekanizmadır.

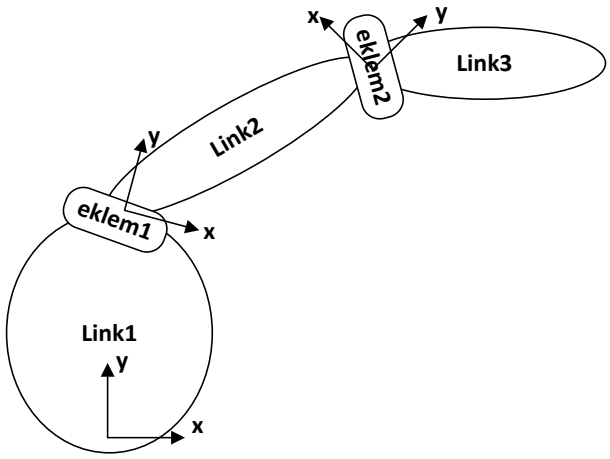
2.3.2 3B modelleme ve benzetim

Robot üzerinde geliştirilecek uygulamaların direk olarak gerçek ortamda denenmesi yerine geliştirme çevrimini hızlandırmak adına benzetim ortamlarına ihtiyaç duyulmaktadır. ROS ile haberleşme kütüphaneleri geliştirilmiş ve açık kaynak kodlu 3B benzetim ortamı olan Gazebo robotik çalışmalarında yaygın olarak kullanılmaktadır. Bu başlık altında robotun simülasyonda da kullanılacak olan 3B sanal modelinin çıkartılması ve Gazebo ile entegrasyonundan bahsedilecektir.

2.3.2.1 3B modelleme

Bir gezgin robot üzerinde birçok alt bileşen ve algılayıcı bulundurmaktadır. Bu bileşenlerin birbirleri ile etkileşimi veya hepsinin beraber değerlendirilerek işlem yapılması gerekmektedir. Örneğin, bir algılayıcıdan okunan değerler algılayıcının robot üzerindeki konumunu orijin kabul ederek elde edilen verilerden oluşmaktadır. Fakat robot üzerinde her bir bileşen için ayrı bir koordinat sistemi bulunmaktadır. Robot üzerindeki bileşenlerden alınan veriler üzerinde işlem yapabilmek için bu koordinat sistemleri arasındaki ilişkilerin tanımlanmış olması gerekmektedir. Koordinat sistemlerinin ilişkileri kullanılarak koordinat dönüşümleri gerçekleştirilmektedir.

ROS ortamında koordinat sistemleri arasındaki ilişkiler URDF (Unified Robot Description File) formatında tanımlanmaktadır. URDF, XML formatında robotun kinematik ve dinamiklerinin, görsellerinin ve çarpışma alanlarının tanımlanmasına olanak sağlayan robot tanımlama dosyasıdır. URDF dosyası kullanılarak gerçek ve simülasyon ortamında anlık koordinat dönüşümleri gerçekleştirilirken, Gazebo ve rviz gibi ortamlarda görsellik de sağlanabilmektedir.



Şekil 2.17 Eklem ve link arasındaki ilişki

Robotlar linkler ve eklemler olmak üzere iki ana kategoriden oluşmaktadır (Şekil 2.17). Linkler robot üzerindeki katı cisimlerdir. Linkler hareket etmemektedir. Linkleri birbirine bağlamak için eklemler kullanılmaktadır. Eklemler robot üzerindeki hareket eden

bölgelerdir. URDF dosyasında linkler ve eklemler tanımlanırken aynı isme sahip (link, joint) XML elementleri kullanılmaktadır.

<link> elementi katı cismin eylemsizliğini ve görsel özelliklerini tanımlamaktadır. **Çizelge 2.1**'de örnek bir link tanımlaması yapılmıştır. Örnekte de görüleceği üzere <link> elementi <collision> (çarpışma), <visual> (görsel) ve <inertial> (atalet) olmak üzere 3 alt elementten oluşmaktadır. Çarpışma elementi linkin çarpışma alanlarının tanımlandığı kısımdır. Tanımlanan geometri kutu, silindir gibi hazır elementler ile tanımlanabilirken daha karmaşık yüzeyler de tanımlanabilmektedir. Burada tanımlanan çarpışma geometrisi ile görselde tanımlanan aynı tanımlanabildiği gibi farklı olacak şekilde de tanımlanabilmektedir. Görsel elementinde ise robota görsel ara yüz tasarlanmaktadır. Bu kısımda robota sadece görsellik katılmaktadır. Yapılan işlemler robotun kinematik ve dinamiğinde herhangi bir etkiye sebep olmamaktadır. Çarpışmada elementinde olduğu gibi burada da geometri elementi kullanılmaktadır. Silindir, kutu gibi geometrilerinin yanında katı cismin stl ya da dae formatındaki katı modelleri de kullanılabilmektedir. Ayrıca cismin renklendirilmesi de bu elementin içerisinde yapılmaktadır. Atalet elementi linkin kütlesi ve ataleti gibi özelliklerinin tanımlandığı alandır. Atalet 3x3 eylemsizlik matrisi şeklinde tanımlanmaktadır. Bu matris simetrik bir matris olduğundan 6 değişkenin tanımlanması yeterli olmaktadır.

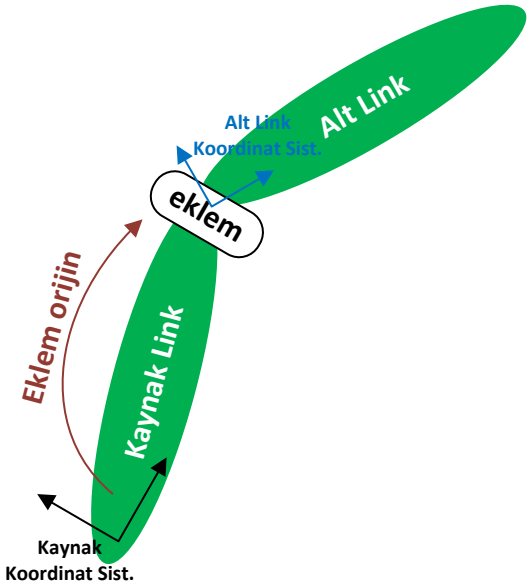
Çizelge 2.1 Kinect derinlik algılayıcısı için link tanımlaması

```
<link name="kinect_link">
        <collision>
                <origin xyz="0 0 0" rpy="0 0 0"/>
                <geometry>
                        <box size="0.1 0.1 0.1"/>
                </geometry>
        </collision>

        <visual>
                <origin xyz="${-55/1000} ${-110/1000} ${-50/1000}" rpy="${-pi/2} ${pi} 0"/>
                <geometry>
                        <mesh filename="package://evarobot_description/meshes/rplidar.stl"
scale="0.001 0.001 0.001" />
                </geometry>
```

```
                <material name="Black" />
        </visual>

        <inertial>
                <mass value="1e-5" />
                <origin xyz="0 0 0" rpy="0 0 0"/>
                <inertia ixx="1e-6" ixy="0" ixz="0" iyy="1e-6" iyz="0" izz="1e-6" />
        </inertial>
</link>
```



Şekil 2.18 Alt linki, kaynak linke bağlama

Eklem elementinde, eklemin kinematik ve dinamik özellikleri tanımlanmaktadır. Ayrıca eklemin kısıtları da bu elementte belirtilmektedir. **Çizelge 2.2**'de örnek bir eklem tanımlaması yapılmıştır. <origin> isimli element ile kaynak link ile alt link arasındaki koordinat ilişkisi tanımlanmaktadır. Şekil 2.18' de görüleceği üzere eklem, alt link koordinat sisteminin üzerine yerleştirilmektedir. <parent> elementi ile alt linkin bağlandığı kaynak link belirtilmektedir. <child> elementi ise alt linki göstermektedir. <axis> elementi ile eklemin alt link koordinat sistemi üzerindeki eksenlerden hangisinin üzerinde hareket kabiliyetine sahip olduğu tanımlanmaktadır. <dynamics> elementi altında eklemin fiziksel sönümlendirme değeri ve fiziksel statik sürtünme kuvveti belirtilmektedir. <limit> ise eklemin limitlerinin belirlendiği elementtir.

Çizelge 2.2 Kinect algılayıcısı için eklem tanımlaması

```
<joint name="kinect_joint" type="floating">
        <origin xyz="0 0 1" rpy="0 0 3.1416"/>
        <parent link="base_link"/>
        <child link="kinect_link"/>

        <calibration rising="0.0"/>
        <dynamics damping="0.0" friction="0.0"/>
        <limit effort="30" velocity="1.0" lower="-2.2" upper="0.7" />
        <safety_controller k_velocity="10" k_position="15" soft_lower_limit="-2.0"
soft_upper_limit="0.5" />
 </joint>
```

2.3.2.2 Benzetim

Gazebo, iç ve dış ortamlar için geliştirilmiş çoklu robot benzetim ortamıdır. Gazebo içerisinde çalışabilen dört adet fizik motoru içermektedir. 3B dünyada gelişmiş robot ve algılayıcı modelleri içermektedir. Açık kaynak kodlu olan Gazebo, robotik çalışmalarında yaygın olarak kullanılmaktadır. ROS ara katmanı ile kullanılabilmesi için sürücü kütüphaneleri bulunmaktadır. Önceki bölümde bahsedilen robotun 3B modeli üzerinde geliştirmeler yapılarak Gazebo için kişisel robot modeli oluşturmak mümkündür.

URDF dosyasında Gazebo için eklenen özellikler <gazebo> elementi içerisinde yer almaktadır. <material> elementi kullanılarak linkler Gazebo ortamında renklendirilmektedir. Gazebo içerisinde robottan bağımsız algılayıcı ve motor eklentileri bulunmaktadır. Bu eklentiler kullanılarak algılayıcıların veri üretmesi ve robotun sürüş tipinin belirlenmesi benzetim ortamında gerçekleştirilebilmektedir. Gazebo eklentileri paylaşılmış kütüphane şeklinde derlenmiş ve simülasyona gömülmüş kod yığınıdır. Eklentiler Gazebo'nun bütün fonksiyonelliğine direk erişime izin veren standart C++ kütüphanelerinden oluşmaktadır. Eklentiler, Gazebo'nun nerdeyse bütün unsurlarının kontrolüne izin vermesiyle, kolayca paylaşılabilen kendine yeten rutinleriyle ve çalışan bir sisteme eklenebilir ve çıkartılabilir yapısıyla kullanımda kolaylıklar sağlamaktadır. Gazebo 6 adet eklenti türüne sahiptir (Gazebo, 2016). Her bir eklenti Gazebo'nun farklı bir bileşeni tarafından kontrol edilmektedir. Dünya eklentisi Gazebo dünyasının kontrol edilmesini sağlayan eklenti türüdür. Fizik motorları ya da ortam ışığının parlaklığı üzerinde yapılacak

değişiklikler dünya eklentileri ile yapılacaklara verilebilecek örneklerden birkaçıdır. Model ve algılayıcıların kontrolü için geliştirilen eklentiler ise model ve algılayıcı tipi eklentiler sınıfına girmektedir. Sistem eklentisi ise komut satırı üzerine özelleşmiş eklentilerdir. Gazebo açılırken çalışan sistem eklentileri kullanılarak açılışta kullanıcının sisteme dahil edilmesi sağlanmaktadır. Görsel eklenti kullanarak robotun izlediği rotanın Gazebo üzerine çizilmesi gibi görsel çalışmalar yapılabilmektedir. Ayrıca GUI eklentisi kullanılarak Gazebo'ya ek olarak kullanıcı arayüzü tasarlanabilmektedir. Bu eklentilerin robot ile bağlama işlemi robot modeli içerisinde <gazebo> elementi altında <plugin> elementi ile gerçekleştirilmektedir. Derinlik kamerası için oluşturulan Gazebo modeli örneği aşağıdaki **Çizelge 2.3**'te verilmiştir. <plugin> elementi içerisinde tanımlanan parametreler algılayıcılara ve eklentilere göre değişiklik göstermektedir. Eklentiler sayesinde gerçek ortamdaki ile benzetim ortamdaki robotun yayınladığı ve abone olduğu topiklerin özellikleri arasında fark kalmayacak şekilde model oluşturulabilmektedir. Böylece geliştirilen kontrol yazılımlarında hiçbir değişiklik yapılmadan iki ortamda da çalışılabilmektedir.

Çizelge 2.3 Kinect algılayıcısı için Gazebo modeli

```
<gazebo reference="camera_link">
 <sensor type="depth" name="camera">
        <always_on>true</always_on>
        <update_rate>20.0</update_rate>
        <camera>
          <horizontal_fov>${60.0*pi/180.0}</horizontal_fov>
          <image>
                <format>R8G8B8</format>
                <width>640</width>
                <height>480</height>
          </image>
          <clip>
                <near>0.05</near>
                <far>8.0</far>
          </clip>
        </camera>
        <plugin name="kinect_camera_controller" filename="libgazebo_ros_openni_kinect.so">
                <cameraName>camera</cameraName>
                <alwaysOn>true</alwaysOn>
```

```
                <updateRate>10</updateRate>
                <imageTopicName>rgb/image_raw</imageTopicName>
                <depthImageTopicName>depth/image_raw</depthImageTopicName>
                <pointCloudTopicName>depth/points</pointCloudTopicName>
                <cameraInfoTopicName>rgb/camera_info</cameraInfoTopicName>
        <depthImageCameraInfoTopicName>depth/camera_info</depthImageCameraInfoTopicName>
                <frameName>camera_depth_optical_frame</frameName>
                <baseline>0.1</baseline>
                <pointCloudCutoff>0.4</pointCloudCutoff>
                <distortionK1>0.00000001</distortionK1>
                <distortionK2>0.00000001</distortionK2>
                <distortionK3>0.00000001</distortionK3>
                <distortionT1>0.00000001</distortionT1>
                <distortionT2>0.00000001</distortionT2>
                <CxPrime>0</CxPrime>
                <Cx>0</Cx>
                <Cy>0</Cy>
                <focalLength>0</focalLength>
                <hackBaseline>0</hackBaseline>
        </plugin>
  </sensor>
</gazebo>
```

3 MATERYAL VE YÖNTEM

Evarobot için yapılan elektro-mekanik sistem tasarımı, elektronik sistem tasarımı, yazılım kontrol mimarisi tasarımı ve gerçeklemeleri takip eden alt bölümlerde verilmektedir.

3.1 Evarobot Modeli

Tez çalışmasına Evarobot için üç katmanda model geliştirilmiştir. Bunlardan ilki robotun mekanik üretiminin yapılabilmesi için geliştirilen mekanik modeldir. Belirlenen isterleri sağlayacak şekilde yapılan tasarım sonucunda oluşan mekanik model üzerinden oluşturulan teknik çizimler kullanılarak robotun mekanik parçalarının üretimi sağlanmıştır. Geliştirilen diğer bir model ise robotun matematiksel modelidir. Robotun üzerindeki motorlarının da dahil edildiği kinematik ve dinamik modelinin çıkartılmasının amacı özellikle düşük seviyede yapılacak kontrol çalışmalarına kaynak sağlamaktır. Ayrıca Matlab, Octave gibi ortamlarda robotun simülasyon ortamının kurulmasında robotun matematiksel modeline ihtiyaç duyulmaktadır. Evarobot için geliştirilen sonuncu model ise robotun 3B modelidir. Bu model robot üzerindeki koordinat sistemleri arasında anlık dönüşümlere imkân vermektedir. Geliştirilen sanal model kullanılarak 3B benzetim ortamı olan Gazebo'da robotun kullanılmasını sağlanmaktadır. Benzetim ortamı sayesinde geliştirilen otonom davranışların testi yapılabilmektedir.

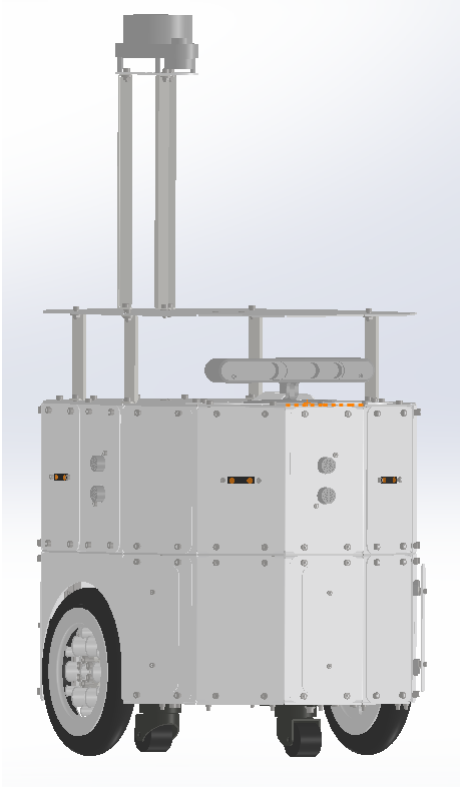
3.1.1 Mekanik model

Eğitim ve araştırma amaçlı gezgin robot platformu geliştirilirken robotun mekanik tasarımı büyük bir önem teşkil etmektedir. Geliştirilecek platformun modüler bir yapıya ve geliştirilebilir bir yapıya sahip olması gerekiyorken maliyetinin de yüksek olmaması gerekmektedir. Robotun tasarımı yapılmadan önce hedef kitlenin ihtiyaçları ve literatürde yapılan eğitim ve araştırmalar incelenmiştir. Çalışma kapsamında geliştirilen gezgin robot platformunun kullanım alanı olarak eğitim çalışmalarına ek olarak araştırma amaçlı çalışmalarda da kullanılması hedeflenmiştir. Bu bölümde robotun mekanik tasarımı

geliştirilirken dikkat edilen noktalardan ve üretim yöntemlerinden bahsedilecektir. Mekanik tasarım ve gerçeklenmesi destek alınan KOSGEB projesindeki diğer araştırmacılar tarafından yapılmıştır.

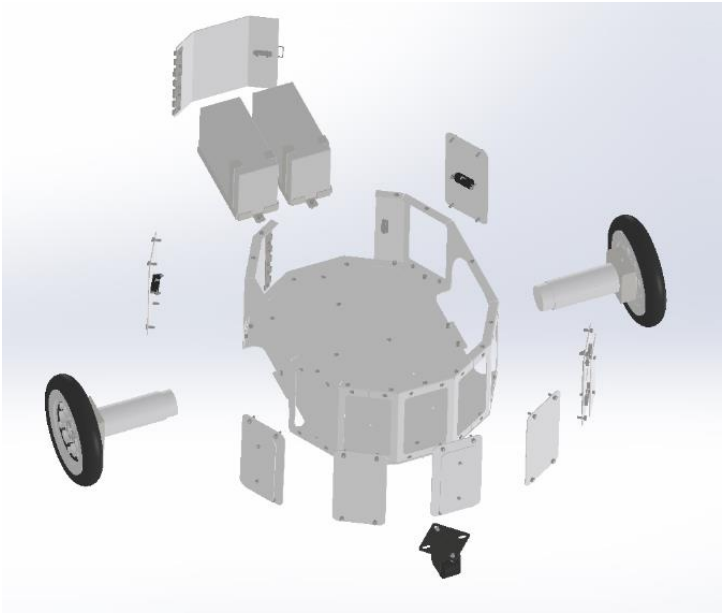
Robotun tasarımı yapılırken dikkat edilen noktaların en başında eğitim ve araştırma amaçlı kullanımlarda kolaylık sağlamasına dikkat edilmiştir. Bu amaç kapsamında modüler bir mekanik tasarım yapılmıştır. Gezgin robot platformu üzerindeki algılayıcıların yeri değişebilir yapıda tasarım yapılmıştır. Üç kattan oluşan robotun ikinci ve üçüncü katlarının kullanımı opsiyonel olmak ile birlikte bunlardan biri ya da ikisi isteğe bağlı kullanılabilmektedir. Bu özellikleri ile Evarobot kişiye özel konfigürasyonlara olanak sağlamaktadır. Tasarımda dikkat edilen diğer önemli bir nokta ise robotu düşük maliyette üretmektir. Bu amaç doğrultusunda robotun parçalarının seri üretime uygun bir şekilde tasarımı yapılmıştır. Ayrıca kalıp kullanılarak yapılan üretim tekniğine ihtiyaç doğuracak tasarımlardan uzak durulmuştur. Diferansiyel sürüşe sahip olacak şekilde tasarlanan robotta sürücü tekerler ile robotun ağırlık merkezi çakışacak şekilde yapılan tasarım sayesinde robotun kontrolünde kolaylık sağlanması hedeflenmiştir. Robotun eğitim ve araştırma amaçlı kullanımlarda uzun süreli çalışmalara imkân sağlaması için iki adet akünün yerleştirilmesine olanak sağlayan mekanik tasarım yapılmıştır. Ayrıca robot ile dışarıdan daha kolay etkileşim sağlamak ve bir sorun olduğunda hızlı müdahaleye imkân sunmak için çok yönlü bir kontrol paneli geliştirilmiştir.

Geliştirilen gezgin robot 335 mm genişlikte, 425 mm derinlikte ve 760 mm yükseklikte ebatlara sahiptir. Robotun mekanik modeli Şekil 3.1'de verilmiştir. Evarobot'un geliştirilen mekanik sistemi farklı yükler altında laboratuvar ortamında test edilmiştir.



Şekil 3.1 Evarobot'un mekanik modeli

Robotun mekanik tasarımı üç katlı yapıda olacak şekilde tasarlanmıştır. Birinci kat, sürücü motorlar ve tekerlerle birlikte sarhoş tekerleri içeren bölümdür. Ayrıca bu kat iki adet batarya ve isteğe bağlı kullanılabilecek algılayıcıların takılabileceği alanları da içermektedir. Bu kata toplamda 7 adet algılayıcı ya da kontrol paneli modülü takılabilmektedir. Katın detay tasarımı Şekil 3.2'de verilmiştir. Bu bölüm robotun temel platformudur. İstenirse kontrol kartı (EKB) da bu bölüme yerleştirilerek robot olarak sadece bu bölüm kullanılabilmektedir. Bundan sonra bahsedilecek olan ikinci ve üçüncü katlardan biri ya da ikisi de bu platformun üzerine eklenebilir bir yapıda tasarıma sahiptir. Robotun ikinci katı, toplamda 14 algılayıcı modül takabilme kapasitesine sahiptir ve Elektronik Kontrol Kartı (EKB)'nı içeren bölümdür. Bu bölümün içerisine ayrıca IMU algılayıcısı ve modem takılmasına olanak sağlanmaktadır. En üst kat olan üçüncü katta ise lazer algılayıcı, RGB ve RGBD kameraların konumlandırıldığı bölümdür. Bu katta bilgisayarların konulabileceği bölüm de bulunmaktadır. Dizüstü bilgisayarlı kullanımlarda kullanımı kolaylaştırmak için bilgisayarın monitörü açıkken lazer algılayıcının görüş açısını engellemeyecek bir tasarıma gidilmiştir. Kullanıcıyı bilgilendirme amaçlı geliştirilen RGB algılayıcının takılabileceği bölüm de bu katta yer almaktadır.



Şekil 3.2 Evarobot birinci katın detay mekanik modeli

Robotun ana iskeleti üzerinde 30’ar derecelik farklı oryantasyonlarda takılabilecek plaka modülleri tasarlanmıştır. Bu modüller ebat ve montaj delikleri bakımında hiçbir farkı olmayacak şekilde tasarlanmıştır. Böylece bu plakaların istenilen yere montajında kolaylık sağlanması amaçlanmıştır. Bu kapsamda dört adet plaka modül geliştirilmiştir. Sonar ve kızılötesi algılayıcılarının takıldığı modüller bunlardan iki tanesidir. Diğer plaka ise tasarımının bu çalışmada yapıldığı bumper algılayıcısıdır. Dördüncü modül ise üzerinde güç butonu, şarj girişi, 3 adet USB çıkışı, 1’er adet HDMI, eternet ve seri port haberleşme birimlerini içermektedir.

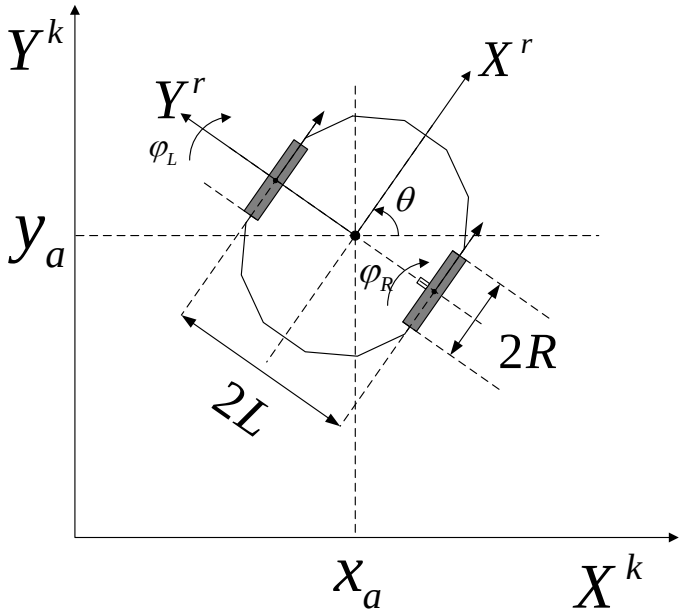
3.1.2 Matematiksel model

Geliştirilen gezgin robot platformu üzerinde yapılacak yeni kontrol ve simülasyon çalışmalarına kolaylık sağlamak adına robotun kinematik ve dinamik modeli çıkartılmıştır. Bu bölümde robotun kinematik ve dinamik modelin çıkartılması aşamaları anlatılmaktadır. Evarobot için çıkartılan matematiksel modelin doğruluğu gerçek robot ile aynı referans girdilerine verdikleri tepkilerin karşılaştırılmasıyla karar verilmiştir.

3.1.2.1 Koordinat sistemi

Gezgin robot platformunun pozisyonunun belirlenebilmesi için iki adet koordinat sistemine ihtiyaç vardır. Bunlardan ilki küresel koordinat sistemidir. Bu koordinat sistemi robotun gezeceği ortam üzerinde sabitlenmiştir. Referans koordinat sistemi olarak kullanılmaktadır ve $\{X^k, Y^k\}$ olarak gösterilecektir. İkinci koordinat sistemi ise yerel koordinat sistemi olarak adlandırılmaktadır. Robot üzerinde tanımlanan koordinat sistemidir ve robot ile birlikte hareket etmektedir. Yerel koordinat sistemi robotun sürüş tekerlerinin orta noktası alınacaktır ve $\{X^r, Y^r\}$ ile gösterilecektir. Şekil 3.3'te verilen robotu küresel koordinat sisteminde pozisyon ve oryantasyonu denklem (3.1)'de tanımlanmıştır.

$$q^I = \begin{bmatrix} x_a \\ y_a \\ \theta \end{bmatrix} \tag{3.1}$$



Şekil 3.3 Evarobot serbest cisim diyagramı

İlerleyen bölümlerde bu iki koordinat sistemi arasında dönüşümler gerçekleştirilecektir. Kolaylık sağlaması için bu iki koordinat sistemi arasındaki

oryantasyon matrisi hesaplanacaktır. Küresel ve yerel koordinat sistemlerindeki herhangi bir noktanın yerinin sırasıyla $X^k = \begin{bmatrix} x^k \\ y^k \\ \theta^k \end{bmatrix}$ ve $X^r = \begin{bmatrix} x^r \\ y^r \\ \theta^r \end{bmatrix}$ şeklinde olduğunu varsayalım. Bu iki nokta arasındaki ilişkiyi tanımlayan oryantasyon matrisi denklem (3.2)' de verilmiştir.

$$X^k = R(\theta) X^r \tag{3.2}$$

Burada,

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.1.2.2 Kinematik kısıtlar

Diferansiyel sürüşlü gezgin robotların hareketi holonomik olmayan kısıt eşitlikleri ile nitelendirilmiştir. Bu kısıt eşitlikleri elde edilirken iki varsayım kullanılmaktadır.

Yanal kayma hareketi: Bu varsayımda robotun sadece ileri ve geri hareketleri gerçekleştirebildiği ve yanal hareket yapmadığı kabul edilmiştir. Bu sebepten robot koordinat sistemindeki y eksenindeki robotun hızı sıfırdan başka bir değer almayacaktır.

$$\dot{y}_a^r = 0 \tag{3.3}$$

Önceki bölümde elde edilen oryantasyon matrisi (denklem 3.2) kullanılarak robotun y eksenindeki hızının küresel koordinat sistemindeki değeri aşağıdaki gibi hesaplanmıştır.

$$
\begin{aligned}
&X^r = R(\theta)^T X^k \\
&\begin{bmatrix} x^r \\ y^r \\ \theta^r \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_a \\ y_a \\ \theta \end{bmatrix} \\
&y^r = -x_a \sin\theta + y_a \cos\theta = 0
\end{aligned} \tag{3.4}
$$

<u>Yuvarlanma kısıtı:</u> Yuvarlanma kısıtı her bir tekerin yüzeye sadece bir P noktasından temas etmesini temsil etmektedir. Bu kapsamda robotun kaymadığı ve savrulmadığı varsayılmaktadır. Bu varsayımlar ışığında robotun sağ ve sol tekerinin P noktasındaki hızları yerel koordinat sisteminde denklem (3.5)' teki gibi ifade edilmektedir.

$$
\begin{aligned}
v_{pR} &= R\dot{\varphi}_R \\
v_{pL} &= R\dot{\varphi}_L
\end{aligned} \tag{3.5}
$$

Bu hızların küresel koordinat sistemindeki eşitliklerinin elde edilmesi için öncelikle sağ ve sol tekerlerin yüzeye temas ettikleri P noktalarının küresel koordinat sistemindeki yeri bulunur.

$$
\begin{cases} x_{pR} = x_a + L\sin\theta \\ y_{pR} = y_a - L\cos\theta \end{cases} \tag{3.6}
$$

$$
\begin{cases} x_{pL} = x_a - L\sin\theta \\ y_{pL} = y_a + L\cos\theta \end{cases} \tag{3.7}
$$

Denklem (3.6) ve (3.7) eşitliklerin türevleri alınır:

$$
\begin{cases} \dot{x}_{pR} = \dot{x}_a + L\dot{\theta}\cos\theta \\ \dot{y}_{pR} = \dot{y}_a + L\dot{\theta}\sin\theta \end{cases} \tag{3.8}
$$

$$
\begin{cases} \dot{x}_{pL} = \dot{x}_a - L\dot{\theta}\cos\theta \\ \dot{y}_{pL} = \dot{y}_a - L\dot{\theta}\sin\theta \end{cases} \tag{3.9}
$$

Oryantasyon matrisi kullanılarak yuvarlanma kısıtı eşitlikleri denklem (3.10)' daki gibi elde edilir:

$$\begin{aligned} \dot{x}_{pR}\cos\theta + \dot{y}_{pR}\sin\theta &= R\dot{\varphi}_R \\ \dot{x}_{pL}\cos\theta + \dot{y}_{pL}\sin\theta &= R\dot{\varphi}_L \end{aligned} \tag{3.10}$$

Tekerlerin P noktası için hızları, yuvarlanma kısıtı eşitliklerinde yerine yazılırsa küresel koordinat sistemindeki yuvarlanma kısıtları elde edilmiş olur.

$$\begin{aligned} \dot{x}_a\cos\theta + \dot{y}_a\sin\theta + L\dot{\theta} &= R\dot{\varphi}_R \\ \dot{x}_a\cos\theta - \dot{y}_a\sin\theta + L\dot{\theta} &= R\dot{\varphi}_L \end{aligned} \tag{3.11}$$

İki kısıttan elde edilen üç eşitlik denklem (3.12)' deki gibi matris formatında yazılabilir.

$$\begin{aligned} &\Lambda(q)\dot{q} = 0 \\ &\Lambda(q) = \begin{bmatrix} -\sin\theta & \cos\theta & 0 & 0 & 0 \\ \cos\theta & \sin\theta & L & -R & 0 \\ \cos\theta & \sin\theta & -L & 0 & -R \end{bmatrix} \\ &\dot{q} = \begin{bmatrix} \dot{x}_a & \dot{y}_a & \dot{\theta} & \dot{\varphi}_R & \dot{\varphi}_L \end{bmatrix}^T \end{aligned} \tag{3.12}$$

3.1.2.3 Kinematik model

Kinematik model, harekete etki eden kuvvetleri hesaba katmadan mekanik sistemin hareket eşitliklerinden oluşmaktadır. Kinematik modellemedeki amaç robotun geometrik parametreleri ile birlikte sürüş tekerlerinin hızları ile robotun hızları arasındaki ilişkiyi tanımlamaktır.

Robotun doğrusal hızı, robotun koordinat sistemindeki sürüş tekerlerinin ortalamasıdır.

$$v = \frac{v_R + v_L}{2} = R\frac{\dot{\varphi}_R + \dot{\varphi}_L}{2} \tag{3.13}$$

Robotun açısal hızı ise sürüş tekerlerinin farkının bu iki teker arasındaki uzunluğa bölünmüş halidir.

$$\omega = \frac{v_R - v_L}{2L} = R\frac{\dot{\varphi}_R - \dot{\varphi}_L}{2L} \tag{3.14}$$

Tanımlanan robot koordinat sistemine göre robotun doğrusal hızı robotun x eksenindeki hızına, açısal hızı ise koordinat sistemindeki robotun açısının değişimine eşittir. Buna göre yukarıda belirtilen doğrusal ve açısal hızın matris formatında gösterimi aşağıdaki gibi olmaktadır.

$$\begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{2L} & \frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \tag{3.15}$$

Oryantasyon matrisi kullanılarak robotun hızları küresel koordinat sisteminde denklem (3.16) gibi ifade edilmektedir.

$$\dot{q}^k = \begin{bmatrix} \dot{x}_a^k \\ \dot{y}_a^k \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}\cos\theta & \frac{R}{2}\cos\theta \\ \frac{R}{2}\sin\theta & \frac{R}{2}\sin\theta \\ \frac{R}{2L} & \frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \tag{3.16}$$

3.1.2.4 Dinamik model

Dinamik model, robotun hareketini etkileyen çeşitli kuvvetlerin hesaba katılarak mekanik sistemin hareket denklemlerinin elde edilmesidir. Dinamik model robotun hareketinin benzetim ortamında analizinin gerçekleştirilmesi için imkân verdiği gibi çeşitli hareket kontrol algoritmalarının tasarlanmasına da olanak sağlamaktadır.

Holonomik olmayan bir robotun n adet koordinatı $(q_1, q_2, \ldots q_n)$ ve m adet kısıtı içeren dinamik modeli denklem (3.17) gibi ifade edilmektedir.

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - \Lambda^T(q)^{\lambda} \tag{3.17}$$

Bu denklemde $M(q)$ nxn simetrik pozitif tanımlı atalet matrisi, $V(q,\dot{q})$ koriyolis matrisi, $F(\dot{q})$ yüzey sürtünme matrisi, $G(q)$ yerçekimi vektörü, τ_d sınırlandırılmış bilinmeyen gürültüler, $B(q)$ girdi matrisi, τ girdi vektörü, $\Lambda^T(q)$ kinematik kısıt matrisi ve λ Lagrange çarpan vektörlerini temsil etmektedir.

Bu çalışmada, robotun matematiksel modeli çıkartılırken Lagrange yaklaşımı kullanılmıştır. Lagrange yönteminde verilen sistemin kinetik ve potansiyel enerjileri kullanılarak hareket denklemleri elde edilmektedir.

Lagrange eşitliği denklem (3.18)' deki formda yazılmaktadır:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) + \frac{\partial L}{\partial q_i} = F - \Lambda^T(q)^{\lambda} \tag{3.18}$$

Lagrange fonksiyonu $L = T - V$, toplam kinetik enerji (T) ve potansiyel enerjinin (V) farkından oluşmaktadır. Elde edilen Langrange fonksiyonu her bir koordinat (q_i) için yukardaki verilen formülün uygulanması gerekmektedir. Yukarıda verilen denklemdeki F kuvvet vektörünü, Λ kısıt matrisini ve λ ise kısıtlarla ilişkili olan Lagrange çarpan vektörlerini belirtmektedir.

Robotun dinamik modeli elde edilirken ilk olarak Lagrange fonksiyonu için gerekli olan kinetik ve potansiyel enerjilerin hesaplanması gerekmektedir. Geliştirilen robot, iki boyutlu x, y düzleminde hareket ettiği için potansiyel enerji sıfır olarak düşünülmüştür ve hesaba katılmamıştır.

Oluşturulacak dinamik modeldeki durum vektörü denklem (3.19) gibidir.

$$q = \begin{bmatrix} x_a & y_a & \theta & \varphi_R & \varphi_L \end{bmatrix}^T \tag{3.19}$$

Sistemin kinetik enerjisi hesaplanması üç parçada gerçekleştirilecektir. Bunlardan ilki tekerleri olmadan robot platformunun kinetik enerjisidir. Denklem (3.20)' de verilen robotun kinetik enerjisi denklemindeki ilk kısım doğrusal hızdan kaynaklanan kinetik enerjidir. Eşitlikteki ikinci kısım ise robotun açısal hızından meydana gelen açısal kinetik enerjidir.

$$T_c = \frac{1}{2} m_c v_c^2 + \frac{1}{2} I_c \dot{\theta}^2 \tag{3.20}$$

Yukarıda verilen denklemde m_c robotun sürüş tekerleri ve motorlarının hesaba katılmadığı kütlesi, I_c ise robotun ağırlık merkezine dik olan eksenindeki eylemsizlik momentidir.

Sağ ve sol tekerlerdeki kinetik enerji denklemleri denklem (3.21-3.22)' de verilmiştir. Eşitliklerdeki ilk bölüm tekerin doğrusal kinetik enerjisini, ikinci bölüm ise tekerin ekseninde oluşan açısal kinetik enerjiyi, son bölüm ise tekerin çevresinde meydana gelen açısal kinetik enerjiyi belirtmektedir.

$$T_{wR} = \frac{1}{2} m_w v_{wR}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\varphi}_R^2 \tag{3.21}$$

$$T_{wL} = \frac{1}{2} m_w v_{wL}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\varphi}_L^2 \tag{3.22}$$

m_w teker ve motorların toplam kütlesi, I_m her bir tekerin teker eksenindeki eylemsizlik momenti ve I_w ise teker çevresindeki eylemsizlik momentidir.

Kinetik enerji denklemlerindeki hızların global koordinat sistemindeki ifadesini elde etmek için denklem (3.23)' deki fonksiyon kullanılacaktır.

$$v_i^2 = \dot{x}_i^2 + \dot{y}_i^2 \tag{3.23}$$

Denklem (3.23)' i kullanmak için robotun ağırlık merkezi ve tekerlerin global koordinat sistemindeki koordinatları aşağıdaki gibi hesaplanmıştır.

$$\begin{cases} x_c = x_a \\ y_c = y_a \end{cases} \tag{3.24}$$

$$\begin{cases} x_{wR} = x_a + L\sin\theta \\ y_{wR} = y_a - L\cos\theta \end{cases} \tag{3.25}$$

$$\begin{cases} x_{wL} = x_a - L\sin\theta \\ y_{wL} = y_a + L\cos\theta \end{cases} \tag{3.26}$$

Eşitlik (3.23) kullanılarak aşağıdaki gibi ağırlık merkezi ve tekerlerin hızları hesaplanmıştır.

$$v_c^2 = \dot{x}_a^2 + \dot{y}_a^2 \tag{3.27}$$

$$v_{wR}^2 = \dot{x}_a^2 + \dot{y}_a^2 + 2\dot{x}_a L\dot{\theta}\cos\theta + 2\dot{y}_a L\dot{\theta}\sin\theta + L^2\dot{\theta}^2 \tag{3.28}$$

$$v_{wL}^2 = \dot{x}_a^2 + \dot{y}_a^2 - 2\dot{x}_a L\dot{\theta}\cos\theta - 2\dot{y}_a L\dot{\theta}\sin\theta + L^2\dot{\theta}^2 \tag{3.29}$$

Hesaplanan üç kinetik enerji denklemlerindeki hızların yukardaki eşitlikler yazılmış ve bu üç kinetik enerji denklemi toplanarak aşağıdaki toplam kinetik enerji bulunmuştur.

$$T = \frac{1}{2}m(\dot{x}_a^2 + \dot{y}_a^2) + \frac{1}{2}I_w(\dot{\varphi}_R^2 + \dot{\varphi}_L^2) + \frac{1}{2}I\dot{\theta}^2 \tag{3.30}$$

Yukarıdaki denklemdeki $m = m_c + 2m_w$ toplam kütle ve $I = I_c + 2m_w L^2 + 2I_m$ toplam atalet olacak şekilde gruplanarak yazılmıştır.

Toplam kinetik enerjinin bulunmasının ardından $L = T$ olmasından kaynaklı Lagrange fonksiyonu elde edilmiştir. Lagrange denkleminde bu eşitlik yerine konularak

durum matrisindeki her bir eleman ve onların türevi için kısmi türevler alınarak denklem (3.31-3.35)'deki hareket denklemleri bulunmuştur.

$$m\ddot{x}_a = C_1 \tag{3.31}$$

$$m\ddot{y}_a = C_2 \tag{3.32}$$

$$I\ddot{\theta} = C_3 \tag{3.33}$$

$$I_w\ddot{\varphi}_R = \tau_R + C_4 \tag{3.34}$$

$$I_w\ddot{\varphi}_L = \tau_R + C_5 \tag{3.35}$$

Denklem (3.31-3.35)' lerde kullanılan $C_1, C_2, \ldots, C_5$ katsayıları kinematik kısıtlarla ilişkilidirler ve Lagrange çarpan vektörü ve kinematik kısıt matrisi formatında yazılabilmektedirler.

$$\Lambda^T(q) = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \end{bmatrix} \tag{3.36}$$

Bulunan hareket denklemleri denklem (3.37) gibi matris formatında yazılabilmektedir.

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} = B(q)\tau - \Lambda^T(q)\lambda \tag{3.37}$$

Burada;

$$M(q) = \begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix},\ V(q,\dot{q}) = 0,\ B(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$\Lambda^T(q)\lambda = \begin{bmatrix} -\sin\theta & \cos\theta & \cos\theta \\ \cos\theta & \sin\theta & \sin\theta \\ 0 & L & -L \\ 0 & -R & 0 \\ 0 & 0 & -R \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \end{bmatrix}$$

İlerleyen bölümde denklem (3.37)'de verilen dinamik model kontrol ve simülasyon çalışmalarında kolaylık sağlaması adına başka bir formda yazılacaktır. Buradaki amaç bilinmeyen Lagrange çarpan vektörlerinden kurtulmaktır.

Bu kapsamda ilk olarak denklem (3.38) gibi bir vektörümüz olduğunu varsayalım.

$$\eta = \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \tag{3.38}$$

Önceki bölümlerde elde edilen kinematik model denklem (3.39)' daki formatta yazılır.

$$\begin{bmatrix} \dot{x}_a \\ \dot{y}_a \\ \dot{\theta} \\ \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} = \frac{1}{2} \begin{bmatrix} R\cos\theta & R\cos\theta \\ R\sin\theta & R\sin\theta \\ \frac{R}{L} & -\frac{R}{L} \\ 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \tag{3.39}$$

Denklem (3.39) verilen kinematik model denklem (3.40) gibi yazılabilir.

$$\dot{q} = S(q)\eta \tag{3.40}$$

Dönüşüm matrisi $S(q)$, kısıt matrisinin $\Lambda(q)$ sıfır uzayını oluşturmaktadır.

$$S^T(q)\Lambda^T(q) = 0 \tag{3.41}$$

Eşitlik (3.40)'ın türevi alınır.

$$\ddot{q} = \dot{S}(q)\eta + S(q)\dot{\eta} \tag{3.42}$$

(3.40) ve (3.42) eşitlikleri, ana eşitlik (3.37) de yerine konulur.

$$M(q)\left[\dot{S}(q)\eta + S(q)\dot{\eta}\right] + V(q,\dot{q})\left[S(q)\eta\right] = B(q)\tau - \Lambda^T(q)\lambda \tag{3.43}$$

Eşitlik soldan $S^T(q)$ matrisi ile çarpılmış ve yeniden düzenlenerek aşağıdaki denklem (3.44) elde edilmiştir. Böylece eşitlikteki Lagrange çarpanını içeren bölüm çıkartılmıştır.

$$\begin{aligned} &S^T(q)M(q)S(q)\dot{\eta} + S^T(q)\left[M(q)\dot{S}(q) + V(q,\dot{q})S(q)\right]\eta \\ &= S^T(q)B(q)\tau - S^T(q)\Lambda^T(q)\lambda \end{aligned} \tag{3.44}$$

Yeni dinamik model matrisleri denklem (3.45)' teki gibi tanımlanmıştır.

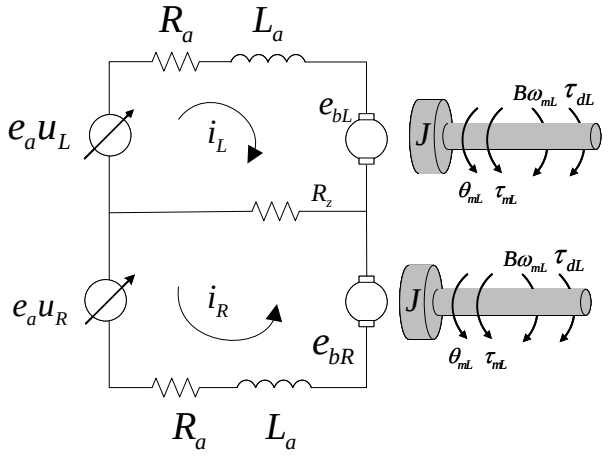
$$\begin{aligned} &\bar{M}(q) = S^T(q)M(q)S(q) \\ &\bar{V}(q,\dot{q}) = S^T(q)M(q)\dot{S}(q) + S^T(q)V(q,\dot{q})S(q) \\ &\bar{B}(q) = S^T(q)B(q) \end{aligned} \tag{3.45}$$

Lagrange çarpan vektörünün silinmesi ile elde edilen indirgenmiş yeni dinamik model denklem (3.46)' da elde edilmiştir.

$$\begin{aligned} &\bar{M}(q)\dot{\eta} + \bar{V}(q,\dot{q})\eta = \bar{B}(q)\tau \\ &\bar{M}(q) = \begin{bmatrix} I_w + \dfrac{R^2}{4L^2}(mL^2 + I) & \dfrac{R^2}{4L^2}(mL^2 - I) \\ \dfrac{R^2}{4L^2}(mL^2 - I) & I_w + \dfrac{R^2}{4L^2}(mL^2 + I) \end{bmatrix} \\ &\bar{V}(q,\dot{q}) = 0 \\ &\bar{B}(q) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned} \tag{3.46}$$

3.1.2.5 DC motor modeli

Önceki bölümlerde geliştirilen dinamik modelin girdisi olan tork, robot üzerinde bulunan iki tane DC motor tarafından üretilmektedir. Bu bölümde motorlara girdi olarak verilen gerilim ile çıktı olan tork arasındaki ilişkiyi tanımlayan motorların matematiksel modeli üretilecektir.



Şekil 3.4 Sağ ve Sol için dc motor modeli

Motor modeli elektrik ve mekanik olmak üzere iki kısımdan oluşmaktadır. Şekil 3.4'de verilen motor modelinde şeklin sol tarafı motorun elektriksel kısmını, sağ tarafı ise mekanik kısmını göstermektedir.

Motor modeli çıkartılırken ilk olarak motorun elektriksel kısmının modeli çıkartılmıştır (Denklem 3.47). Sağ ve sol motorların elektriksel modeli göz akım yöntemi kullanılarak çıkartılmıştır.

$$\begin{aligned} e_a u_L - e_{bL} &= L_a \dot{i}_L + \left(R_a + R_z\right) i_L + R_z i_R \\ e_a u_R - e_{bR} &= L_a \dot{i}_R + \left(R_a + R_z\right) i_R + R_z i_L \end{aligned} \tag{3.47}$$

Verilen eşitlerdeki e_a sistemin toplam beslemesi, u_L ve u_R sırasıyla sol ve sağ motorlara uygulanan PWM sinyalin çalışma doluluk oranı, L_a motorların endüvi

endüktansı, R_a motorların endüvi direnci, e_{bL} ve e_{bR} sırasıyla sol ve sağ motorun ters emk'sı ve i_L ve i_R sırasıyla sol ve sağ motorlar üzerinden geçen akımdır. Motorların beslemesi ortak olduğundan gerçek ortamda bir motorun çektiği akım diğer motor üzerinden geçen akımı etkilemektedir. Bu durumu modele dahil etmek için Şekil 3.4'de görüldüğü üzere modelin elektriksel kısmına R_z isimli direnç eklenmiştir.

Motor üzerindeki ters emk (e_{bL} , e_{bR}) ve motor mili dönüş hızı (ω_{mL} , ω_{mR}) arasındaki ilişkinin tanımlanması için denklem (3.48) gibi ters emk katsayısı (K_b) kullanılmıştır.

$$\begin{aligned} e_{bL} &= K_b \omega_{mL} \\ e_{bR} &= K_b \omega_{mR} \end{aligned} \tag{3.48}$$

Motorların mekanik modeli denklem (3.49) gibi çıkartılmıştır. Eşitliklerde verilen τ_{mL} , τ_{mR} sol ve sağ motor milinde üretilen tork değerleri, J motorların eylemsizlik momenti, B dönüş direnci katsayısı, ω_{mL} ve ω_{mR} motor millerinin dönüş hızları ve τ_{dL} , τ_{dR} bilinmeyen gürültülerdir.

$$\begin{aligned} \tau_{mL} &= J\dot{\omega}_{mL} + B\omega_{mL} + \tau_{dL} \\ \tau_{mR} &= J\dot{\omega}_{mR} + B\omega_{mR} + \tau_{dR} \end{aligned} \tag{3.49}$$

Tork sabiti K_t , motorların ürettiği tork ve motorların üzerinden geçen akım arasındaki ilişkiyi tanımlamaktadır.

$$\begin{aligned} \tau_{mL} &= K_t i_L \\ \tau_{mR} &= K_t i_R \end{aligned} \tag{3.50}$$

Geliştirilen gezgin platform üzerinde motor milini direk olarak tekerlere bağlamak yerine motorlar tarafından üretilen torku artırmak için dişli kutusunda geçirilerek bağlanmıştır. Bu dişli kutusundan kaynaklı motor milinde üretilen tork ile tekere uygulanan tork arasındaki ilişki tanımlanırken N dişli oranı kullanılmıştır.

$$\begin{aligned} \tau_L &= N\tau_{mL} \\ \tau_R &= N\tau_{mR} \end{aligned} \tag{3.51}$$

Elektriksel ve mekanik olmak üzere iki kısımda incelenerek üretilen motor eşitlikleri, kontrol ve simülasyon çalışmalarında kolaylık sağlamak adına matris formunda yazılmıştır. Geliştirilen motor modelinin çıktısı olan torklar, robotun dinamik modeline girdi olarak uygulanabilmektedir.

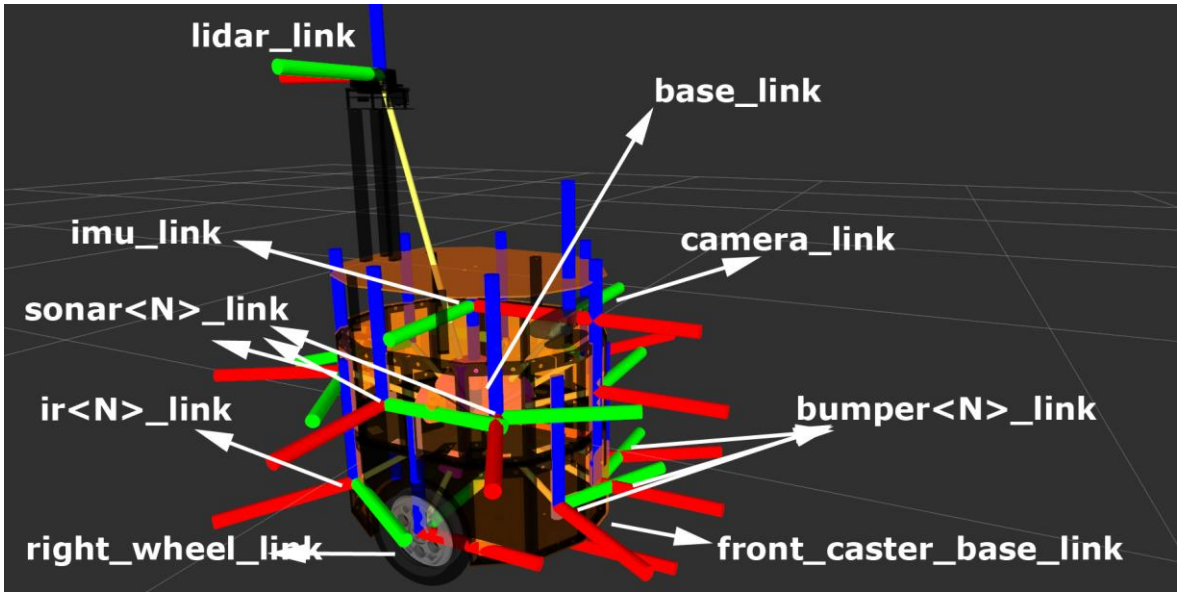
$$\begin{bmatrix} \dot{\omega}_{mR} \\ \dot{\omega}_{mL} \\ \dot{i}_R \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} -\frac{B}{J} & 0 & \frac{K_t}{J} & 0 \\ 0 & -\frac{B}{J} & 0 & \frac{K_t}{J} \\ -\frac{K_b}{L_a} & 0 & -\frac{R_a+R_z}{L_a} & -\frac{R_z}{L_a} \\ 0 & -\frac{K_b}{L_a} & -\frac{R_z}{L_a} & -\frac{R_a+R_z}{L_a} \end{bmatrix} \begin{bmatrix} \omega_{mR} \\ \omega_{mL} \\ i_R \\ i_L \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{e_a}{L_a} & 0 \\ 0 & \frac{e_a}{L_a} \end{bmatrix} \begin{bmatrix} u_R \\ u_L \end{bmatrix} - \begin{bmatrix} \tau_{dR} \\ \tau_{dL} \\ 0 \\ 0 \end{bmatrix} \tag{3.52}$$

$$\tau = \begin{bmatrix} \tau_R \\ \tau_L \end{bmatrix} = N \begin{bmatrix} 0 & 0 & K_t & 0 \\ 0 & 0 & 0 & K_t \end{bmatrix} \begin{bmatrix} \omega_{mR} \\ \omega_{mL} \\ i_R \\ i_L \end{bmatrix} \tag{3.53}$$

3.1.3 Evarobot 3B modeli

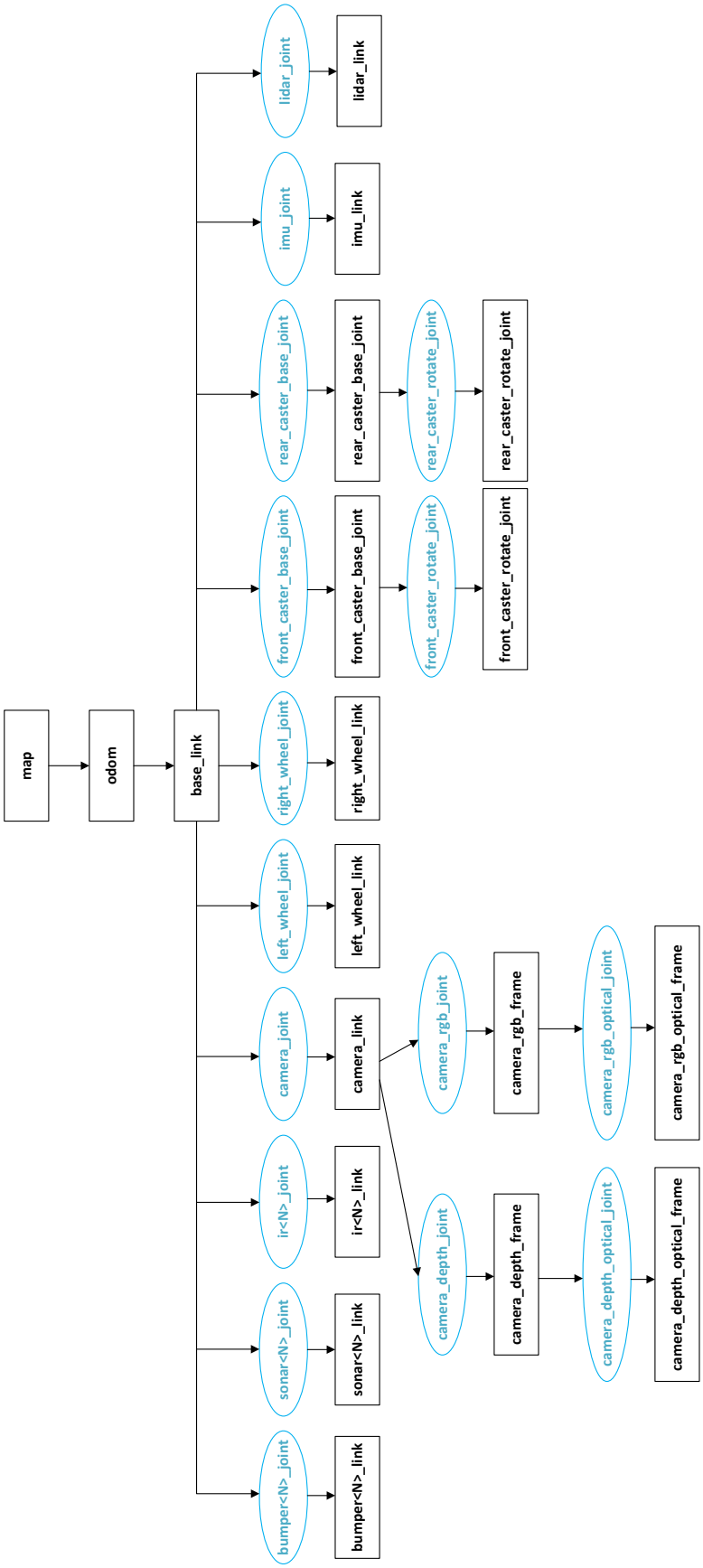
Evarobot üzerindeki her bir algılayıcıya ait koordinat sistemi bulunmaktadır. Bu koordinat sistemleri arasında anlık koordinat dönüşümleri, görselleştirilmesi ve robotun benzetim ortamı modelini oluşturmak için robotun sanal modeli oluşturulmuştur. Evarobot'un 3B modeli ROS araçları ve Gazebo ile birlikte kullanılabilmesi için URDF formatında geliştirilmiştir. Evarobot için tanımlanan linklerin robot üzerindeki konumlarının görselleştirilmiş hali Şekil 3.5'de verilmiştir. Bu linkler ve eklemler tanımlanırken XML üzerine makroların tanımlanmasına imkân veren Xacro kullanılmıştır. Xacro kullanılmasındaki amaç makroların fonksiyonelliği sayesinde gerçek ortamda

yapılan algılayıcı yerlerinin ve adetlerinin değiştirilmesi gibi durumlarda robotun sanal modelinde de yapılacak değişiklikler için kolaylık sağlamaktır.



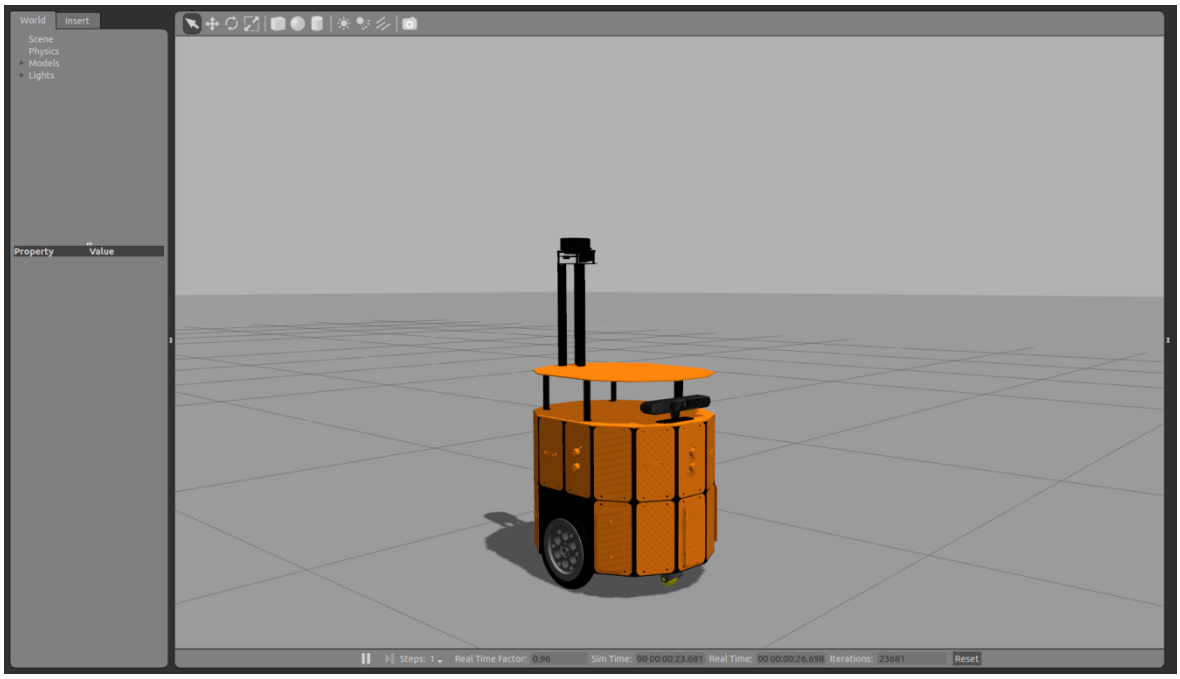
Şekil 3.5 Linklerin Evarobot üzerinde görselleştirilmiş hali

Evarobot'un sanal robot modeli oluşturulurken geometrik merkezine base_link adında link atanmıştır ve robot üzerindeki diğer algılayıcıların bu link ile olan ilişkileri link ve eklemler kullanılarak tanımlanmıştır (Şekil 3.6). 'base_link' evarobot'un yerel koordinatının koordinat sistemini oluşturmaktadır. Bu link sayesinde algılayıcılardan gelecek bilgilerin dönüştürüleceği ortak bir koordinat sistemi oluşturulmuştur. Ayrıca, diğer algılayıcıların arasındaki ilişki bu link sayesinde sağlanmaktadır. Motorlarda bulunan enkoder ile elde edilen robotun mutlak konumu ile base_link arasındaki koordinat dönüşümünün tanımlanması için 'odom' isminde link oluşturulmuştur. Ayrıca robotun AMCL (Adaptive Monte Carlo Localization) kullanılarak elde edilen global konumu 'map' isimli link üzerinden yayınlanmaktadır. Robot üzerindeki koordinat sistemleri arasındaki ilişkiler URDF formatında tanımlanarak robotun sanal modeli oluşturulmuştur. İlerleyen bölümlerde detaylı olarak bahsedilen 'evarobot_state_publisher' isimli düğüm bu modeli kullanarak anlık koordinat dönüşümlerini gerçekleştirmektedir. Ayrıca bu düğüm sayesinde rviz üzerinde robotun üzerindeki algılayıcılar ve robotun konumu gibi bilgiler görselleştirilebilmektedir.



Şekil 3.6 Evarobot link ve eklemleri

Yukarıda bahsedilen Evarobot'un sanal modeli Gazebo benzetim ortamı üzerinde çalıştırıldığında sadece görsellik elde edilmektedir (Şekil 3.7). Algılayıcılardan verilerin okunması, robotun ortamda sürülebilmesi ve benzetin ortamı ile ROS üzerinden haberleşebilmek için Gazebo eklentilerinin robotun sanal modeline URDF formatında yerleştirilmesi yapılmıştır. Evarobot geliştirilirken hazır Gazebo eklentileri kullanılmasının yanında evarobot için ek Gazebo eklentileri geliştirilmiştir. Evarobot üzerinde lazer, derinlik kamerası, IMU ve diferansiyel sürüş eklentileri kullanılmıştır. Ayrıca bunlara ek olarak bumper, sonar ve kızılötesi algılayıcılar için eklentiler geliştirilmiştir.

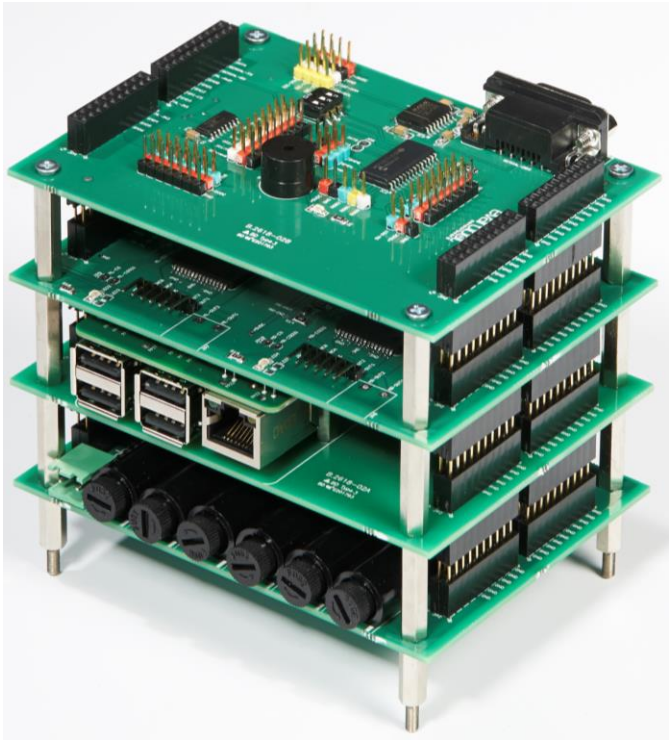


Şekil 3.7 Evarobot'un Gazebo ortamındaki görüntüsü

3.2 Evarobot Elektronik Kontrol Birimleri

Evarobot üzerindeki çevre birimlerinin ve motorların kontrolü için elektronik kontrol birimi (EKB) donanımı geliştirilmiştir (Şekil 3.8). EKB mobil robot uygulamaları için geliştirilmiş ROS uyumlu kontrol kartıdır. Kart; güç, motor sürücü, Raspberry Pi ve algılayıcı kartları olmak üzere 4 katlı yapıdadır. Değiştirilebilir katlı yapısı sayesinde farklı konfigürasyonlarda kullanıma uygundur. Robotik uygulamalarda sıklıkla kullanılan algılayıcılar için güç ve sinyal girişleri bulunmaktadır. Farklı düzeyde robotik uygulamaları için konfigüre edilebilir, kullanımı kolay ve geliştirilebilir yapısı ile çok

yönlü bir kontrol kartıdır. Bu bölümde EKB kartları ve bu kartlar için geliştirilmiş çevre birimleri donanımlarından bahsedilecektir.



Şekil 3.8 Evarobot Elektronik Kontrol Birimi (EKB)

Güç kartı, Evarobot üzerindeki diğer kart ve algılayıcılar için giriş gerilimini 9V, 5V ve 3.3V gerilimlere dönüştürerek giriş gerilimi dahil toplam 16 kanaldan 80 W gücünde çıkış vermektedir. Giriş gerilimi 7-40V aralığında olabilmektedir. Evarobot üzerinde giriş gerilimi olarak 12V kullanılmaktadır.

Motor sürücü kartı, PWM (Darbe Genişlik Modülü) ile sürülen iki kanallı DC motor sürücü kartıdır. Bu kart, 5.5-24V besleme gerilimi aralığında kanal başına sürekli 12A verebilmektedir. 20 kHz değerine kadar PWM sinyal frekansı desteklenmektedir. Ayrıca kart üzerinde enkoderli motorlar için enkoder bağlantı ara yüzü bulunmaktadır.

Raspberry Pi kartı, içerisinde düşük seviyeli kontrol yazılımının çalıştığı Raspberry Pi isimli mini bilgisayarı içermektedir. Raspberry Pi kartının görevi B+/2/3 modellerinin pimlerini EKB kartının standart pimlerine uyumlu hale getirmektedir.

Algılayıcı kartı, I2C, SPI ve RS232 haberleşmesi kullanarak çeşitli algılayıcılar için giriş ve çıkış bağlantıları sağlamaktadır. Bu kart ile kullanılacak elektronik kartın I/O bağlantısı için de bağlantılara sahiptir. Farklı seviyelerde robotik/elektronik uygulamalar için kullanıma uygundur.

RGB led modülü elektronik ve robotik uygulamalarda kullanıcıya görsel bilgilendirme sağlamaktadır. PWM ile kontrol edildiğinde daha fazla renk kombinasyonu oluşturulabilmektedir.

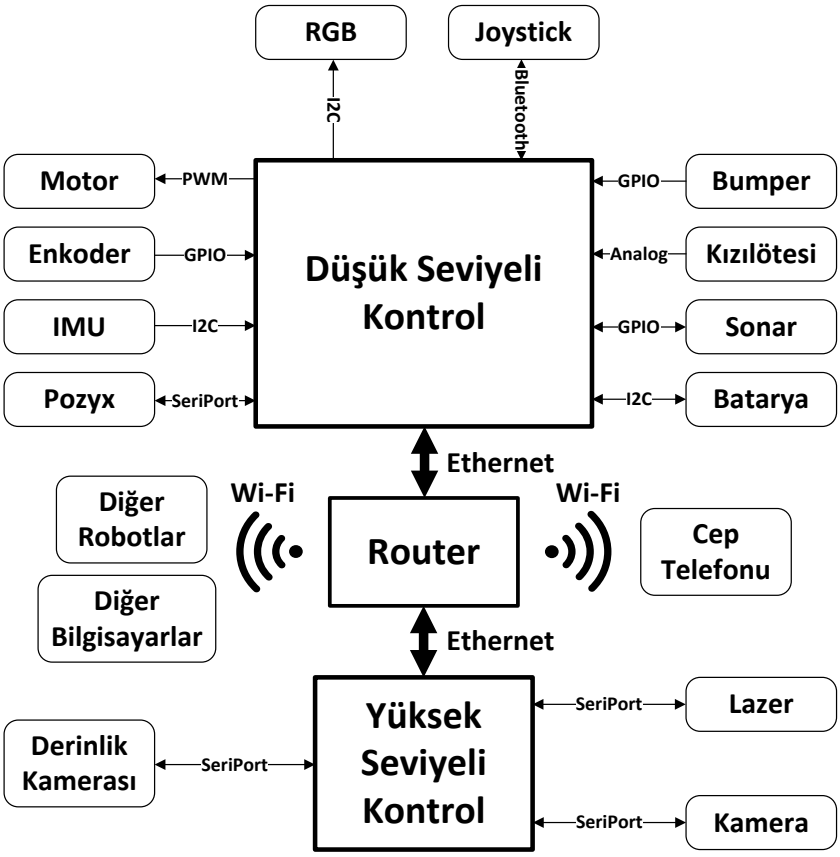
Batarya sensörü, batarya gerilimi, akımı ve şarj durumu %1 hassasiyet ile kullanıcıya bilgi sağlamaktadır. Bataryanın kimyasal yapısı ve kapasitesi gözetilmeden herhangi bir batarya ile kullanılabilmektedir. Bu algılayıcının kontrolü için I2C haberleşme alt yapısı kullanılmaktadır.

Çalışma kapsamında geliştirilen diğer algılayıcı ise ultrasonik mesafe algılayıcısıdır. Algılayıcı 2 cm ile 5 m aralığında mesafe ölçümü yapılmasını sağlamaktadır. Algılayıcının çalışma mantığı 40 kHz frekanslı sinyal dizisi yayınlanarak sinyalin aldığı yol süresi kadar çıkış sinyali üretmesidir. Çıkış sinyalinin genişliği ölçülerek mesafe hesaplaması yapılmaktadır.

3.3 Evarobot Yazılım Kontrol Mimarisi

Evarobot yazılım kontrol mimarisi Düşük Seviyeli Kontrol (DSK) ve Yüksek Seviyeli Kontrol (YSK) olmak üzere iki seviyeli bir yapıda kurulmuştur. DSK yazılımı EKB donanımı üzerinde çalışmaktadır. EKB donanımı içerisinde RT-Linux konfigürasyonunda çekirdeği tekrardan derlenen Raspbian işletim sistemi kullanılmaktadır. DSK'nın görevi robot üzerindeki algılayıcılar ve motorlar ile ilgili işlemlerin yapılmasıdır. DSK'nın etkileşimde olduğu donanımlar ve bunlarla olan haberleşme için kullanılan ara yüzler **Şekil 3.9**'da verilmiştir. DSK'da ana kontrol bileşeni olarak ROS kütüphaneleri çalışmaktadır. Bu kütüphanelerin çıktıları YSK ve diğer birimler ile paylaşılmaktadır. YSK donanımı robot üzerine yerleştirilen taşınabilir bilgisayardır ve üzerinde Ubuntu 14.04 işletim sistemi çalışmaktadır. YSK'nın görevi lazer, derinlik kamerası gibi yüksek veri üreten algılayıcıların okunması ve robotun otonom davranışları için yüksek seviyeli

kontrolün gerçekleştirilmesidir. Evarobot'un çalışması sırasında robota uzaktan bağlanılabilmektedir. DSK, YSK ve dışarıdan bağlanan diğer bilgisayarlar ROS alt yapısı sayesinde büyük bir düğüm gibi çalışmaktadır. Bu sayede uzaktan bağlanan bilgisayarlar kullanılarak geliştirilen MATLAB kütüphaneleri ya da ROS kütüphaneleri ile robotun anlık kontrolüne müdahale edilebilmekte ve kullanıcı ile robot arasında bir arayüz oluşturulabilmektedir. Uzaktan bilgisayar bağlamak dışında sisteme diğer robotların bağlanması da mümkündür. Ayrıca üzerinde ROS ara katmanı çalışmayan cep telefonu gibi donanımların TCP portları kullanılarak JSON formatında sisteme entegrasyonu gerçekleştirilebilmektedir.

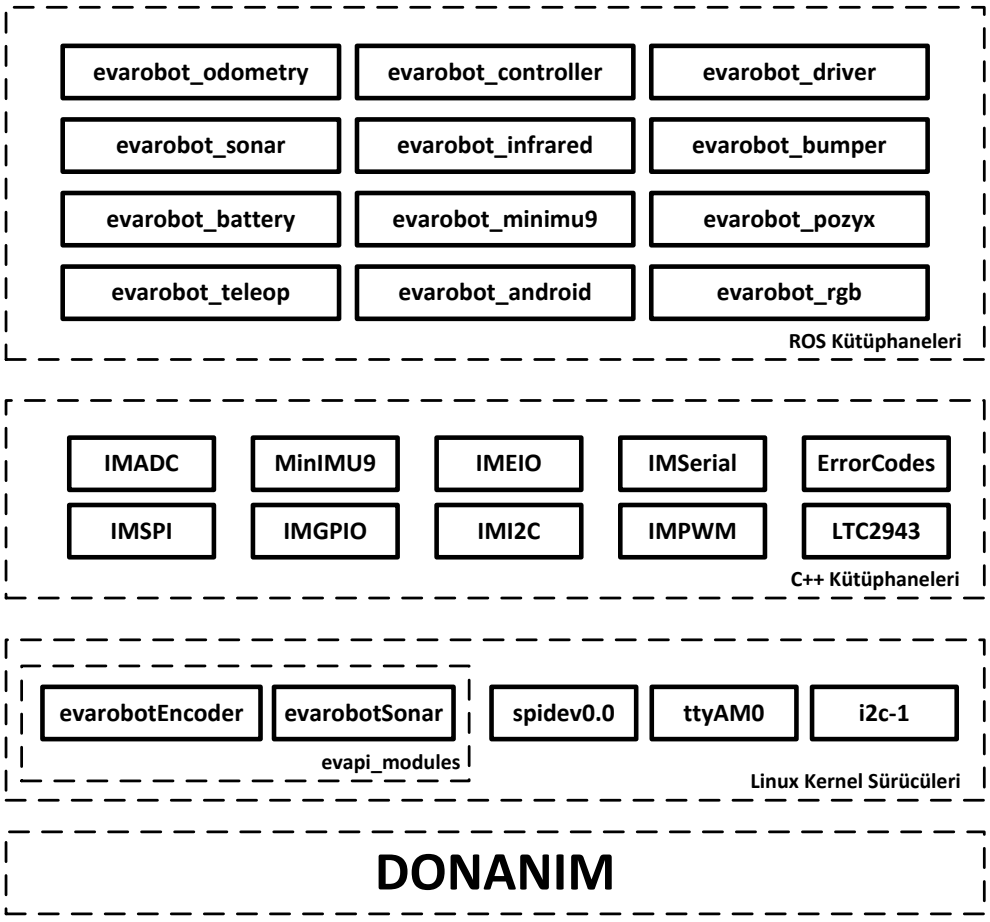


Şekil 3.9 Evarobot çevre birimleri ve haberleşme ara yüzleri

3.3.1 Düşük seviyeli kontrol

Düşük seviyeli kontrolün en üst katmanında ROS kütüphaneleri çalışmaktadır. Fakat bu düğümlerin robot donanımları ile haberleşebilmesi için arada C++ kütüphaneleri ve Linux çekirdek sürücülerinin bulunduğu katmanları kullanılmaktadır. Bahsedilen üç

katmanlı DSK yapısı **Şekil 3.10**'da verilmiştir. Üç katmanlı bu yapıda donanım ile ilk temasta bulunan katman Linux çekirdeklerinin bulunduğu katmandır. Çalışma kapsamında SPI, I2C ve Seri haberleşme için Linux'un içerisinde bulunan sürücülere ek olarak enkoder ve sonar mesafe algılayıcıları için sürücüler geliştirilmiştir. Orta katmanda bulunan C++ kütüphanelerinin görevi ise donanım sürücüleri ile haberleşmektir. Bu kütüphaneler sınıflardan oluşmaktadır. Bu sayede birden fazla ROS düğümünde bu sınıfların kullanımına ve geliştirilebilir bir yapıda olmasına olanak sağlanmıştır. C++ kütüphaneleri ile iletişimde olan ROS kütüphanelerinin görevi ise donanımların kontrolünü gerçekleştirmektir. Bu bölümde DSK'nın çalışma yapısı aktarımı en tepede bulunan ROS düğümleri ve bu düğümlerin diğer katmanlar ile olan ilişkileri üzerinden gerçekleştirilecektir.

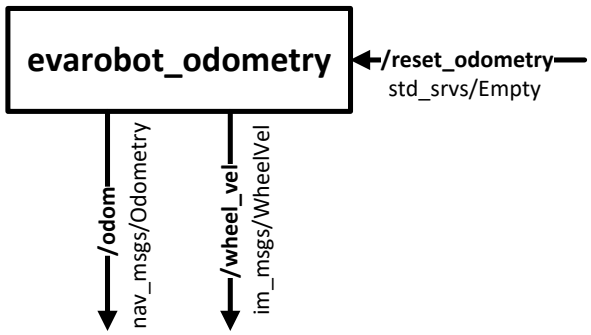


Şekil 3.10 Düşük seviyeli kontrol mimarisi

Takip eden alt bölümlerde Şekil 3.10' da verilen ROS kütüphaneleri detaylı olarak incelenmektedir.

3.3.1.1 Evarobot_odometry

Bu düğüm, tekerlerin dönüş sayısı bilgisinden robotun mutlak konum bilgisini ve hız bilgilerini hesaplamaktadır. Dönüş sayısı bilgisi enkoderlerin ürettiği darbeler sayılarak elde edilmektedir. Yüksek frekansta darbe sayımları yaparak yanlış dönüş bilgisi elde etme olasılığını düşürmek amaçlanmıştır. Bu kapsamda normal GPIO kütüphanelerini kullanmak yerine Linux çekirdek sürücüsü geliştirilmiştir. Geliştirilen "evarobotEncoder" isimli Linux çekirdek sürücüsü, sağ ve sol tekerler üzerindeki enkoderlerin ürettiği darbeleri GPIO pimlerinden okuyarak saymaktadır. Sayılan bu değerler, "evarobot_odometry" düğümü tarafından alınmaktadır. Robotun kinematik modeli kullanılarak mutlak konum bilgisi hesaplanmaktadır.



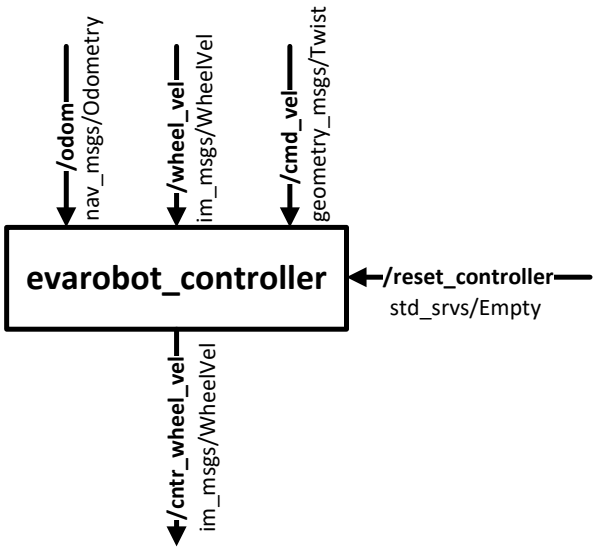
Şekil 3.11 evarobot_odometry düğüm yapısı

"evarobot_odometry" düğümü iki tane topik yayınlamaktadır ve bir adet servise sahiptir (Şekil 3.11). "/odom" düğümü üzerinden robotun hesaplanan mutlak konumu ve açısal ve doğrusal hızı paylaşılmaktadır. Ayrıca "/wheel_vel" topiği ile sağ ve sol tekerlerin dönüş hızları yayınlanmaktadır. Düğüm içerisinde hesaplanan konum robotun harekete başlangıç noktasını merkez kabul etmektedir. "/reset_odometry" servisi çağrılarak robotun başlangıç konumu ve çekirdek modülü içerisindeki darbe sayıları sıfırlanmaktadır.

3.3.1.2 Evarobot_controller

Bu düğüm, robotun hızını kontrol eden düğümdür. Düğüm içerisinde referans hız ile ölçülen hız arasındaki farkı sıfırlamak için tasarlanan iki adet PID denetleyici

çalışmaktadır. PID denetleyiciler sağ ve sol tekerlerin hızını kontrol etmektedir. Standart PID denetleyicilerde bulunan D bileşeni ani referans değişikliklerinde aşırı tepkilere sebep olmaktadır. Bu yüzden D bileşeni tasarlanan denetleyici geri besleme tarafında kullanılmıştır. Ayrıca gerçek ortamda ani çıkışlarda gerilim uygulamak mümkün olmadığından her iki denetleyicinin sonuna alçak geçirgen filtre kullanılmıştır.



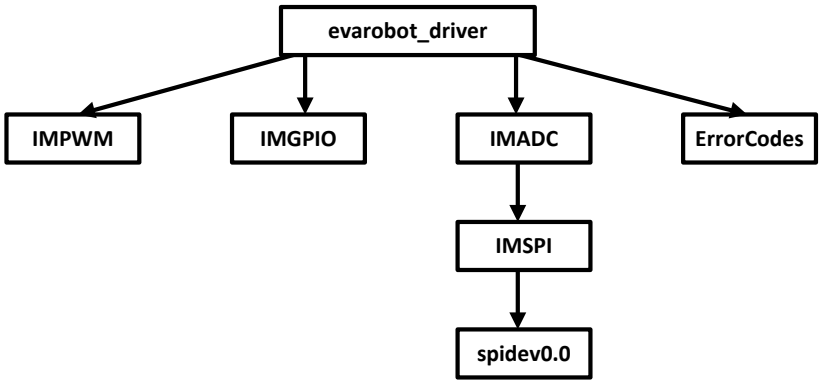
Şekil 3.12 evarobot_controller düğüm yapısı

Evarobot içerisinde açısal/doğrusal hızların kontrolü ya da sağ/sol teker hızları yapılabilmektedir. Bu iki tür denetleyiciden hangisinin kullanılacağına bir parametrede değişiklik yapılarak karar verilebilmektedir. Bu sebepten düğüm ölçülen açısal/doğrusal hızları için "/odom" ve sağ/sol teker hızları için "/wheel_vel" topiklerine abone olmaktadır. Denetleyicideki referans hız için "/cmd_vel" topiği dinlenmektedir. Denetleyicilerin çıktısı olan sağ ve sol tekere uygulanacak hızlar "/cntr_wheel_vel" isimli topikten yayınlanmaktadır. Ayrıca düğüm içerisinde denetleyicilerin hataları sıfırlamak için "/reset_controller" adında bir servis çalışmaktadır. Düğümün girdi ve çıktıları Şekil 3.12'de görselleştirilmiştir.

3.3.1.3 Evarobot_driver

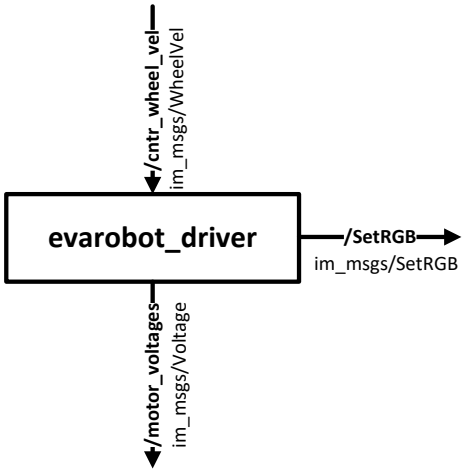
Bu düğüm denetleyiciler tarafından üretilen sağ ve sol teker hızlarını darbe genişlik modülasyonu (PWM) kullanarak motor sürücüleri üzerinden motorlara uygulamaktadır.

Düğümün kullandığı kütüphaneler Şekil 3.13'te verilmiştir. "IMPWM" ve "IMGPIO" kütüphaneleri motorları sürmek için kullanılmaktadır. "IMPWM" kütüphanesi ile motorlara uygulanacak sinyaller üretilmekte ve "IMGPIO" kütüphanesi ile tekerlerin dönüş yönleri ayarlanmaktadır. Ayrıca bu düğüm motorların çektiği akım bilgisini analog-dijital çevirici ile SPI üzerinden haberleşerek elde etmektedir. Düğüm çekilen akım motorlar için tanımlanan maksimum akım bilgisine ulaştığında motorları durdurarak motorların yanmasını engellemektedir. Düğüm çalışırken akım hatası ya da kod içerisinde hata oluştuğunda kullanıcıyı bilgilendirmek için "ErrorCodes" kütüphanesi kullanılarak hata kodu üretilmekte ve kaydedilmektedir.



Şekil 3.13 evarobot_driver düğümünün diğer katmanlarla olan ilişkisi

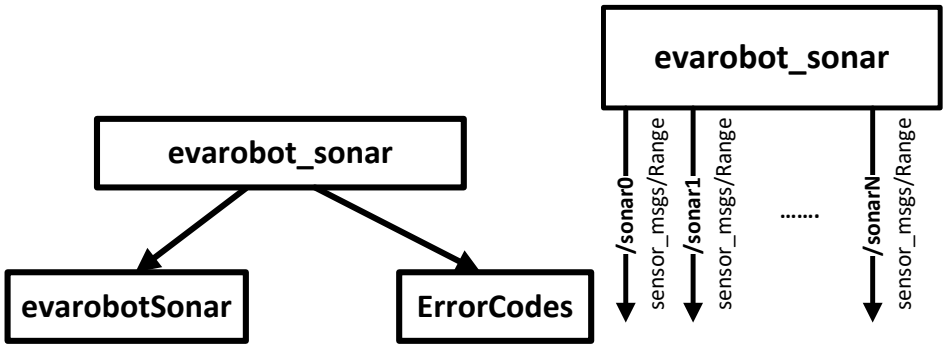
Düğümün topik ve servis girdi çıktıları Şekil 3.14'te verilmiştir. "evarobot_driver" düğümü robota uygulanacak hızlar için "evarobot_controller" düğümünün yaydığı "/cntr_wheel_vel" topiğini dinlemektedir. Eğer bu topikten gelen veriler tanımlanmış frekans aralığında veri yayınlamıyorsa, "evarobot_driver" düğümü hata kodu üreterek robotun hareketini sonlandırmaktadır. Düğüm içerisinde oluşan hataların kaydedilmesinin yanı sıra IM-RGB10 led modülü üzerinden kullanıcının bilgilendirilmesi de sağlanmaktadır. Bunun için "/SetRGB" servisi kullanılmaktadır. Ayrıca "evarobot_driver" düğümü motorların çektiği akım bilgisini "/motor_voltages" topiğinden yayınlamaktadır.



Şekil 3.14 evarobot_driver düğüm yapısı

3.3.1.4 Evarobot_sonar

Bu düğüm robot üzerindeki sonar mesafe algılayıcılarını sürmektedir. Düğüm "evarobotSonar" ve "ErrorCodes" kütüphanelerini kullanmaktadır (Şekil 3.15). Düğüm, algılayıcıları sürmek için "evarobotSonar" isimli Linux çekirdek modülünü kullanmaktadır. IMGPIO kütüphanesini yerine Linux çekirdek modülünün geliştirilmesinin sebebi daha yüksek frekanslarda çalışmak ve daha modüler bir yapı oluşturmaktır. Evarobot üzerinde kullanılan IMSMO20 isimli sonar mesafe algılayıcıları tek bir GPIO üzerinden haberleşmektedir. Algılayıcılar tetikleneceği zaman çıkış moduna alınmakta, tetiklendikten sonra ise giriş moduna alınarak algılayıcının çıktısı beklenmektedir. "evarobot_düğümü" algılayıcıların okuduğu mesafe bilgisini okumanın yanında belirli bir frekansta algılayıcıların canlılıklarını kontrol etmektedir. Bir hata oluşması durumunda "ErrorCodes" kütüphanesi üzerinden hata kodu üretmekte ve bu hatayı kaydetmektedir.

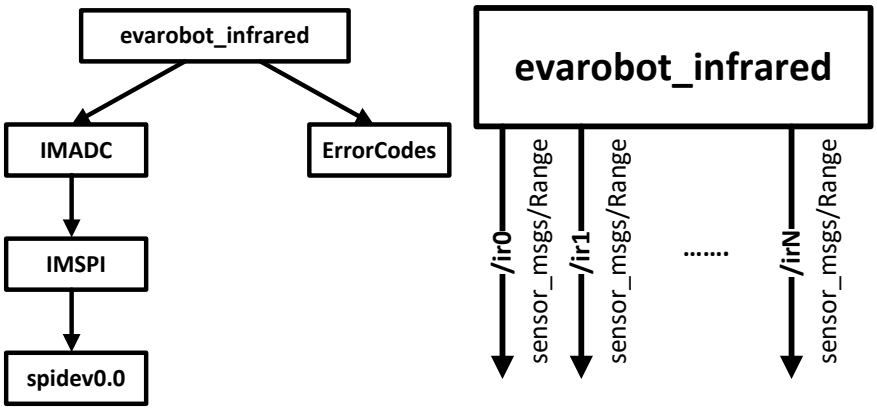


Şekil 3.15 evarobot_sonar düğümü bağımlılıkları ve düğüm yapısı

"evarobot_sonar" düğümü okunacak algılayıcı bilgisini parametre olarak almaktadır. Ayrıca düğüm, algılayıcıların gruplanarak daha yüksek frekansta sürülmesine olanak sağlamaktadır. Düğüm, sadece okunacak olarak ayarlanan algılayıcıların derinlik bilgilerini ayrı topiklerden yayınlamaktadır. Donanım kısıtından dolayı maksimum 7 adet sonar mesafe algılayıcısı sürülebilmektedir. Bu algılayıcı bilgileri ROS'un standart mesaj tipinde yayınlanarak hazır araçlarla görselleştirilmesinde kolaylık sağlanması amaçlanmıştır.

3.3.1.5 Evarobot_infrared

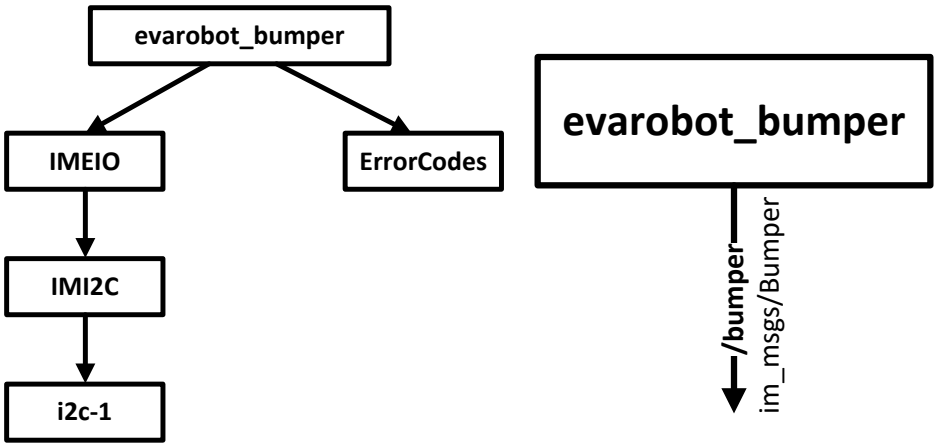
Bu düğüm, robot üzerinde takılı olan kızılötesi algılayıcıları sürmektedir. Bu düğümün bağlı olduğu kütüphaneler Şekil 3.16'da verilmiştir. Kızılötesi algılayıcılar ölçülen derinlik bilgisine göre gerilim değeri üretmektedir. Bu analog sinyal "IMADC" kütüphanesi ile SPI üzerinden alınmaktadır. Ölçülen gerilim değeri aşağıdaki formül kullanılarak mesafe bilgisine dönüştürülmektedir. Denklemde kullanılan değerler deneysel olarak elde edilmiştir. Düğüm içerisinde algılayıcıların canlılık kontrolü yapılmaktadır ve bir problem ile karşılaşıldığında "ErrorCodes" kütüphanesi kullanılarak hata mesajı üretilmekte ve kaydedilmektedir. "evarobot_infrared" düğümü tanımlanan kızılötesi algılayıcıların her biri için ayrı bir topik oluşturmakta ve okunan değerleri buradan yayınlamaktadır.



Şekil 3.16 evarobot_infrared düğümü bağımlılıkları ve düğüm yapısı

3.3.1.6 Evarobot_bumper

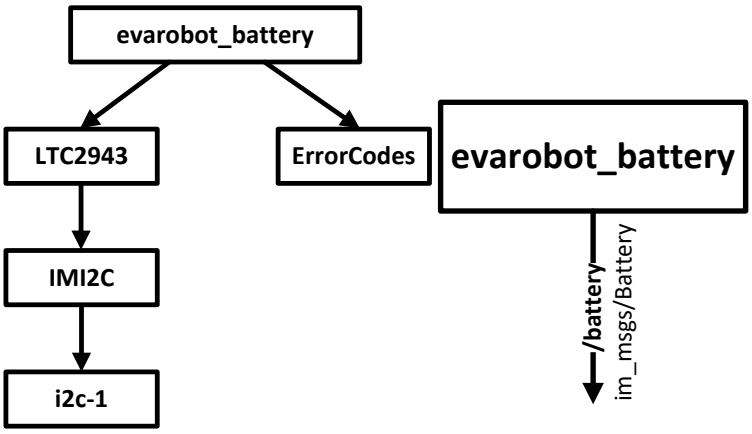
Bu, robot üzerindeki bumper algılayıcılarının durumlarını okuyan düğümdür. Düğümün kullandığı diğer kütüphaneler Şekil 3.17'te verilmiştir. Bumper sensörleri genişletilmiş GPIO entegresi üzerinden kontrol edilmektedir. Bu sebepten dolayı entegre ile haberleşmek için "IMEIO" ve "IMI2C" kütüphaneleri kullanılmıştır. Diğer düğümlerde olduğu gibi hata kodları ve mesajları üretmek için "evarobot_bumper" düğümü "ErrorCodes" kütüphanesini kullanmaktadır. Düğüm okunan bumper sensör bilgilerini tanımlanmış özel mesaj tipi ile tek bir düğüm üzerinden yayınlamaktadır.



Şekil 3.17 evarobot_bumper düğümü bağımlılıkları ve düğüm yapısı

3.3.1.7 Evarobot_battery

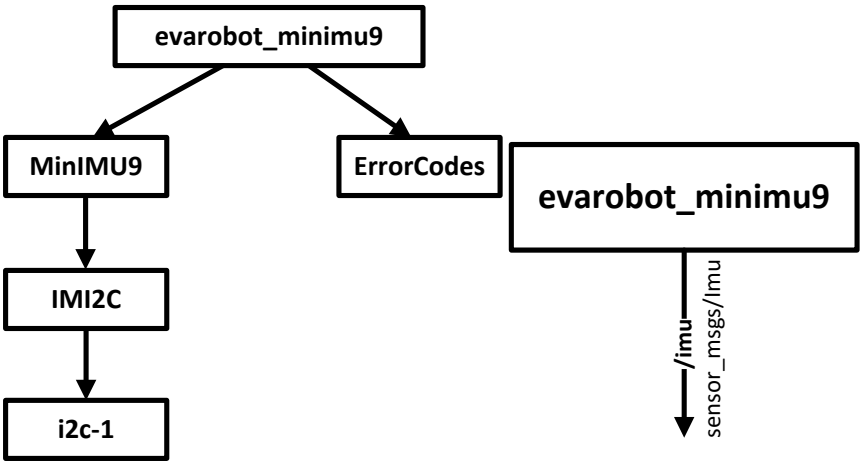
Bu, batarya sensörü ile I2C ile üzerinden haberleşerek robot üzerindeki batarya bilgilerini okuyan ve yayınlayan düğümdür. Batarya algılayıcısı üzerinden bataryaların toplam gerilimi, çekilen akım, sıcaklık ve kapasite bilgilerini edinmektedir. Bu kapsamda "LTC2943" ve "ErrorCodes" kütüphanelerini kullanmaktadır (Şekil 3.18). Okunan algılayıcı bilgilerini özel mesaj tipli "/battery" isimli topikten yayınlamaktadır.



Şekil 3.18 evarobot_battery düğümü bağımlılıkları ve düğüm yapısı

3.3.1.8 Evarobot_minimu9

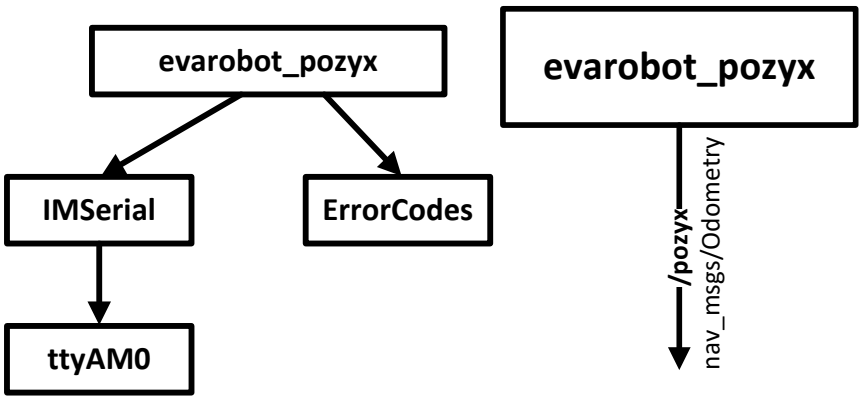
Bu düğüm, IMU algılayıcısının okuduğu manyetik alan ve jiroskop verilerini Kalman Filtre ile birleştirerek robotun oryantasyon bilgisini üretmektedir. Algılayıcılar ile haberleşme "IMI2C" kütüphanesi kullanılarak I2C üzerinden gerçekleşmektedir. "MinIMU9" kütüphanesi ise Kalman Filtrenin çalıştığı kütüphanedir. Ayrıca düğüm içerinde hata kontrolü yapılmakta ve herhangi bir hata durumunda "ErrorCodes" kütüphanesi ile hata mesajı üretilmekte ve kaydedilmektedir. Düğüm elde edilen robotun oryantasyon bilgisini standart ROS mesaj tipi kullanılarak "Imu" isimli topikten yayınlanmaktadır (Şekil 3.19).



Şekil 3.19 evarobot_minimu9 düğümü bağımlılıkları ve düğüm yapısı

3.3.1.9 Evarobot_pozyx

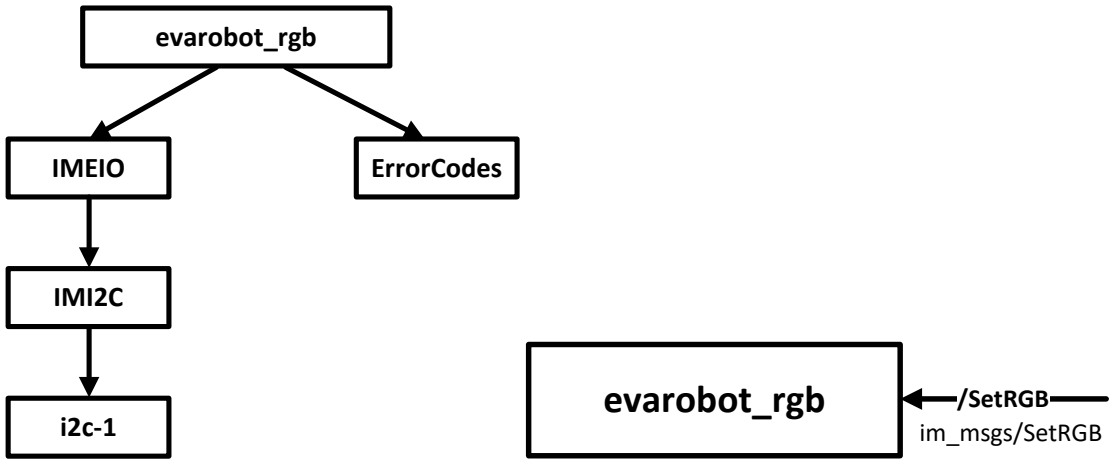
Bu düğüm, robotun daha hassas konumunun elde edilebilmesi için Ultra Geniş Band (UWB) kullanılarak konum bilgisi üreten pozyx isimli algılayıcının ROS sürücüsüdür. Düğümün "IMSerial" ve "ErrorCodes" kütüphanelerine bağımlılıkları bulunmaktadır (Şekil 3.20). Algılayıcı üzerinde bulunan Arduino kit ile "IMSerial" kütüphanesi kullanılarak seri port haberleşmesi yapılmaktadır. Bu haberleşme için seri port protokolü geliştirilmiştir. "evarobot_pozyx" düğümü robotun hesaplanan global konumunu "/pozyx" isimli topikten yayınlanmaktadır.



Şekil 3.20 evarobot_pozyx düğümü bağımlılıkları ve düğüm yapısı

3.3.1.10 Evarobot_rgb

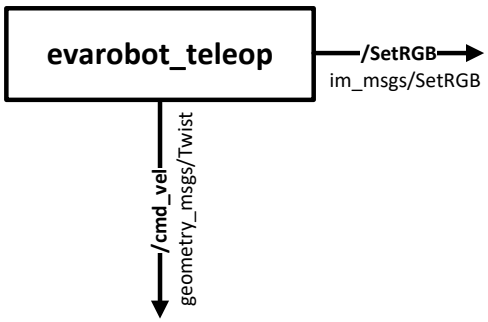
Bu, robot ile kullanıcı arasındaki etkileşimi sağlamak için geliştirilen IM-RGB10 modülünü kontrol eden düğümdür. IM-RGB10 modülü genişletilmiş GPIO entegresi ile sürüldüğünden I2C üzerinden gerçekleşen haberleşmenin kontrolü "IMEIO" kütüphanesi ile gerçekleştirilmiştir (Şekil 3.21). "ErrorCodes" kütüphanesi ile düğüm içerindeki oluşabilecek hata mesajları üretilmekte ve kaydedilmektedir. "evarobot_rgb" düğümü herhangi bir topik yayınlamamakta ve üye olmamaktadır. İçerisinde sadece "/SetRGB" isimli servis bulundurmaktadır. Bu servis sayesinde tanımlı modlar kullanılabilmekte veya yeni mod oluşturulabilmektedir. Diğer düğümler görsellik sağlamak için bu servis ile haberleşmektedir. Ayrıca geliştirilecek yeni düğümlerle entegre edilebilmektedir.



Şekil 3.21 evarobot_rgb düğümü bağımlılıkları ve düğüm yapısı

3.3.1.11 Evarobot_teleop

Bu düğüm, robotun joystick ile kontrol edilmesini sağlamaktadır. Bu düğüm Bluetooth üzerinden joystick ile haberleşmektedir. Joystickte tanımlanan butonlara basıldığında robota uygulanması gereken hız bilgilerini üretmektedir. Bu hız bilgileri "/cmd_vel" isimli topikten yayınlanmaktadır (Şekil 3.22). Ayrıca robotun joystick ile sürülme modunda olduğunun bilgisi "/SetRGB" servisi kullanılarak IM-RGB10 üzerinde görselleştirilmektedir.

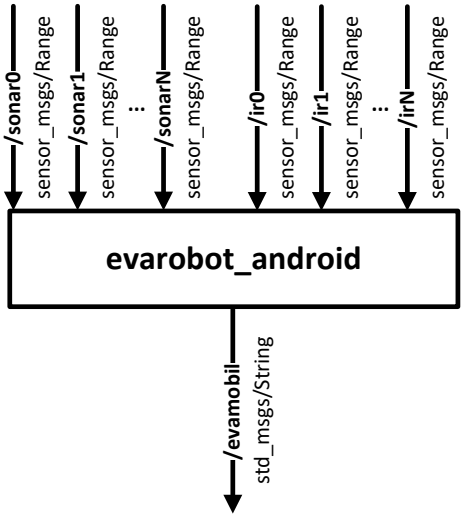


Şekil 3.22 evarobot_teleop düğüm yapısı

3.3.1.12 Evarobot_android

Bu düğüm, robotun Android işletim sistemine sahip mobil cihazlar üzerinden kontrolünün sağlayan ROS sürücüsüdür. Bu düğüm mobil cihaz ile sonar ve kızılötesi algılayıcı bilgilerini paylaşmaktadır. Ayrıca mobil cihaz üzerinde çalışan yazılımdan elde

edilen hız bilgilerini alarak robot uygulanmasını sağlamaktadır. Düğüm ile mobil cihaz arasındaki haberleşme wi-fi üzerinden JSON formatındaki bilgi paketleri ile gerçekleştirilmektedir. Düğümün abone olduğu ve yayınladığı topikler Şekil 3.23'te verilmiştir.



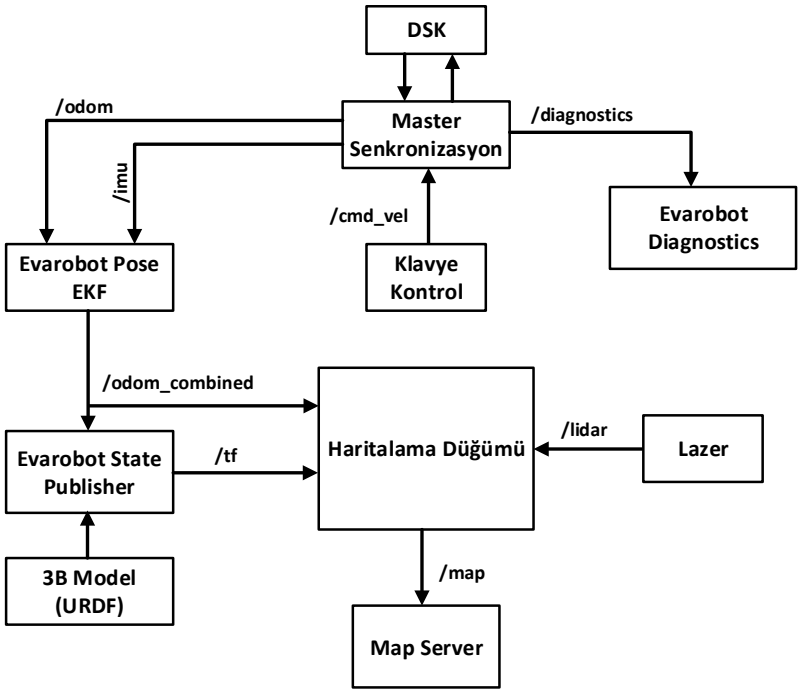
Şekil 3.23 evarobot_android düğüm yapısı

3.3.2 Yüksek seviyeli kontrol

Düşük seviyeli kontrolde robot üzerindeki donanımların sürülmesi ve alınan bilgilerin ROS ortamına aktarılması gerçekleştirilirken, yüksek seviyeli kontrolde bu bilgilerin anlamlandırılması ve otonom kontrol için ilgili komutların oluşturulması işlemi gerçekleştirilmektir. Yüksek seviyeli kontrolde temel amaç robotun bir noktadan diğerine otonom hareket ederken ortamdaki dinamik ve statik nesnelere çarpmadan görevini tamamlamasını sağlamaktır. Bu amaç doğrultusunda robot harita üzerinde kendisini konumlandırması gerekmektedir. Algılayıcılardan aldığı bilgiler ile kendisini ortam haritasında konumlandırırken, derinlik algılayıcılarından aldığı bilgilerle bir sonraki adımlarda çarpışma olup olmayacağını hesaplayarak buna karşı önlem almalıdır. Bu bölümde robottan beklenen bu görevleri üstlenen yüksek seviyeli kontrolün mimarisinin ve içerisindeki düğümlerin anlatımı yapılacaktır. Yüksek seviyeli kontrolden sorumlu düğümlerin anlatımı birim bazlı yerine ortamın haritasının çıkartılması, konumlandırma ve otonom navigasyon başlıkları altında düğümlerin üstlendiği görevler üzerinden yapılacaktır.

3.3.2.1 Haritalama

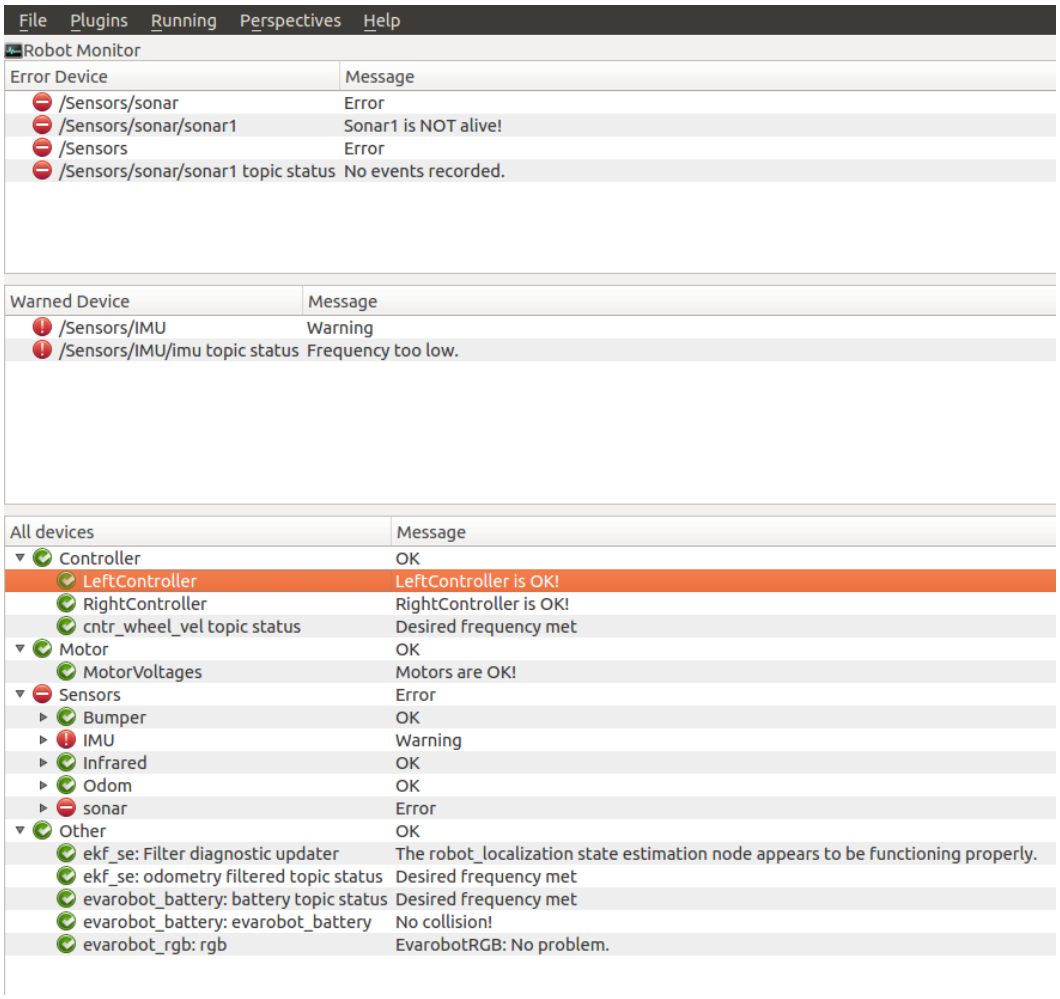
Bazı uygulamalarda Evarobot'un kendisini konumlandırabilmesi için ortam haritasının daha önceden oluşturulmuş olması gerekmektedir. Çalışmada SLAM (Simultaneous Localization and Mapping) kullanılarak robot eş zamanlı konumlandırılmakta ve ortamın haritası çıkartılmaktadır. Evarobot ile ortam haritası çıkartılırken yüksek seviyeli kontrol tarafında çalışan düğümler ve bunların arasındaki ilişki Şekil 3.24'te verilmiştir.



Şekil 3.24 Ortam haritası çıkarılırken kullanılan düğüm yapısı

Evarobot için tasarlanan yazılım kontrol mimarisinde DSK ve YSK içerisindeki işlemlerinin birbirini etkilememesi için birbirinden bağımsız iki adet ROS-Master çalışmaktadır. Bu iki master arasında veri iletişiminin sağlanabilmesi için düğüm ve topiklerin senkronizasyonu yapan "/master_sync" düğümü kullanılmaktadır. DSK tarafından üretilen "/odom" ve "/imu" bilgileri "evarobot_pose_ekf" isimli düğüm içerisinde EKF (Genişletilmiş Kalman Filtre) ile tümleştirelerek robotun oryantasyon bilgisi iyileştirilmiş bağıl konumu elde edilmektedir ve "/odom_combined" isimli topik üzerinden yayınlanmaktadır. İyileştirilmiş bağıl konum ve robotun 3B sanal modeli,

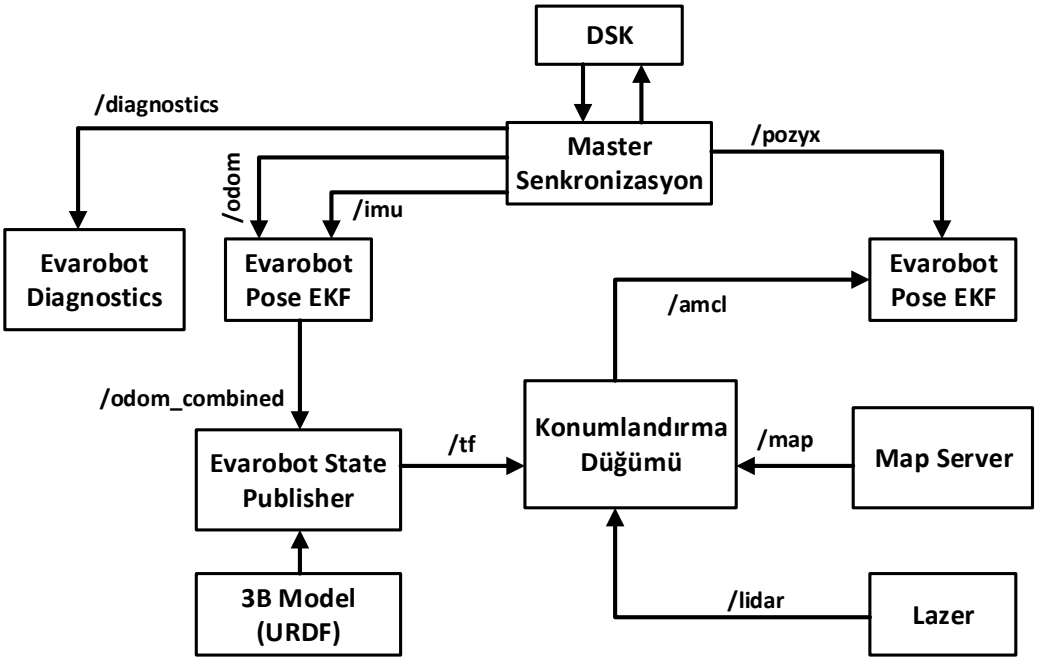
“evarobot_state_publisher” isimli düğüm tarafından kullanılarak eklemler ve bağıl konum arasındaki koordinat dönüşüm matrisleri “/tf” isimli topikten yayınlanmaktadır. İçerisinde SLAM algoritması çalışan “evarobot_mapping” düğümü, “odom_combined”, “tf” ve lazer mesafe algılayıcısının okuduğu derinlik bilgisini içeren “lidar” topiklerini kullanarak ortamın haritası çıkarmakta ve “/map” isimli topikten yayınlamaktadır. Ortamın haritası çıkartılırken robot “teleop_twist_keyboard” düğümü üzerinden klavye ile sürülmektedir. Bütün ortamın haritası çıkartılacak şekilde robot gezdirildikten sonra “map_saver” düğümü kullanılarak harita konumlandırma ve otonom navigasyonda kullanılmak amacıyla kaydedilmektedir. Ayrıca kullanıcı tarafında robot çalışırken üzerindeki donanımların hataların görselleştirilmesi yapılması için “evarobot_diagnostics” düğümü kullanılmaktadır (Şekil 3.25).



Şekil 3.25 Evarobot diyagnostik ekranı

3.3.2.2 Konumlandırma

İç ortamlarda konumlandırmada, dış ortamlardaki gibi standartlaşmış bir yapı bulunmamaktadır. İç ortam için GPS benzeri geliştirilmiş sistemler olup, ortama vericiler ve robot üstüne de ek alıcılar yerleştirilmesi gerekmektedir. Çok hassas konuma ihtiyaç duyulmayan robotik çalışmalarında lazer ve enkoder sensörleri kullanılarak yapılan konumlandırma tercih edilmektedir. Çünkü bu konumlandırma yönteminde ortama sabit alıcı ya da vericiler yerleştirmeye gerek kalmamaktadır. Evarobot ile bu iki yöntemle de konumlandırma yapılabilmesi için gerekli mimari oluşturulmuştur. Bu bölümde öncelikle SLAM kullanılarak oluşturulan harita kullanılarak robotun gerçek ortamda algılayıcılardan elde ettiği bilgiler ile harita bilgisi karşılaştırılarak robotun konumlandırılması yapılmaktadır. Konumu iyileştirmek için Pozyx algılayıcısı kullanılmaktadır. Evarobot'un konumlandırılması yapılırken kullanılan düğümlerin aralarındaki ilişki Şekil 3.26'da verilmiştir.



Şekil 3.26 Konumlandırma yapılırken kullanılan düğüm yapısı

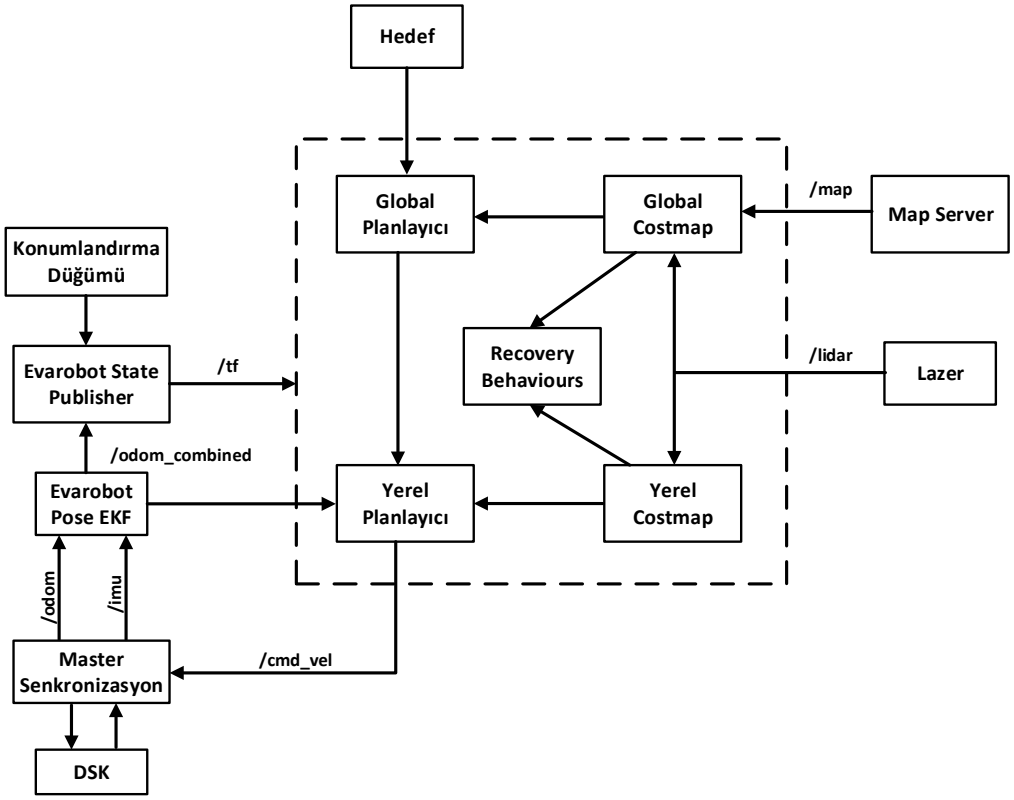
Haritalama bölümünde de bahsedildiği gibi burada da DSK ile haberleşme ROS-Master'ın senkronizasyonu yapılarak gerçekleşmektedir ve "evarobot_pose_ekf" ve "evarobot_state_publisher" düğümleri kullanılarak iyileştirilmiş bağıl konum ve transformasyon matrisleri elde edilmektedir. "map_server" düğümü kullanılarak

oluşturulmuş ortam haritası yayınlanmaktadır. Anlık olarak ortamdan lazer ile alınan ortam bilgisi ile iyileştirilmiş bağıl konum ve harita, AMCL (Adaptive Monte Carlo Localization) kullanılarak tümleştirilmekte ve robotun global konumu elde edilmektedir. Bu yöntem kullanılarak genel robotik çalışmaları için yeterli bir hassasiyette konum üretilmektedir. Fakat daha hassas bir konum bilgisine ihtiyaç duyulduğu zamanlarda Pozyx algılayıcısı kullanılabilmektedir. Pozyx ile AMCL'de elde edilen konum bilgisi "evarobot_pose_ekf" düğümü kullanılarak EKF ile tümleştirildiğinde daha hassas bir konum elde edilmektedir.

3.3.2.3 Otonom navigasyon

Robotun otonom navigasyon yapabilmesi için kurulan yapı Şekil 3.27'de verilmiştir. Buradaki DSK ile haberleşme ve iyileştirilmiş bağıl konum önceki bölümlerde bahsedildiğinden bir farkı bulunmamaktadır. Şeklin kolay anlaşılabilmesi için global konumun üretilme kısmı detaylandırılmamış ve konumlandırma düğümü olarak görselleştirilmiştir. Robota hedef nokta "goal" isimli topik üzerinden tanımlanmaktadır. Bu hedef nokta topiğe herhangi bir düğüm üzerinden verilebileceği gibi rviz kullanılarak harita üzerinden bir hedef nokta işaretlenerek de yapılabilmektedir. Evarobot'un otonom olarak verilen hedef noktasına ulaşması için iki tane yol planlayıcısı bulunmaktadır. Bunlardan ilki global planlayıcıdır. SLAM yapılarak çıkartılmış ortam haritası, harita sunucusu tarafından yayınlanmaktadır ve global planlayıcının girdisidir. Global planlayıcı robotun şu anki konumdan verilen hedef noktasına ulaşması için A* algoritmasını kullanarak yol planı çıkarmaktadır. Bu plan çıkartılırken ortamda bulunan dinamik nesneler dahil edilmemektedir. Sadece harita üzerindeki nesneler dahil edilmektedir. Robot tek bir noktadan oluşmadığı için harita üzerinde bulunan engellerin en az robotun yarı çapı kadar şişirilmesi gerekmektedir ve bu işlem global costmap isimli düğüm tarafından gerçekleştirilmektedir. Bunun yüzden global planlayıcı haritayı global costmap'den geçirildikten sonra almaktadır. Otonom navigasyonda çalışan diğer planlayıcı ise yerel planlayıcıdır. Bu planlayıcının görevi, global planlayıcıdan alınan planı ortamdaki dinamik ve haritada bulunmayan diğer nesnelerden sakınarak uygulamaktadır. Evarobot'da çalışan yerel planlayıcı olarak DWA (Dynamic Windows Approach) yöntemi kullanılmaktadır. Global planlayıcıda olduğu gibi yerel planlayıcıda da nesne büyültülme işlemi gerçekleştirilmektedir. Yerel planlayıcının çıktısı robota uygulanacak açısal ve doğrusal

hız bilgileridir. Bu bilgiler "cmd_vel" topiği üzerinden DSK'ya iletilerek robota uygulanması sağlanmaktadır.



Şekil 3.27 Otonom navigasyon yapılırken kullanılan düğüm yapısı

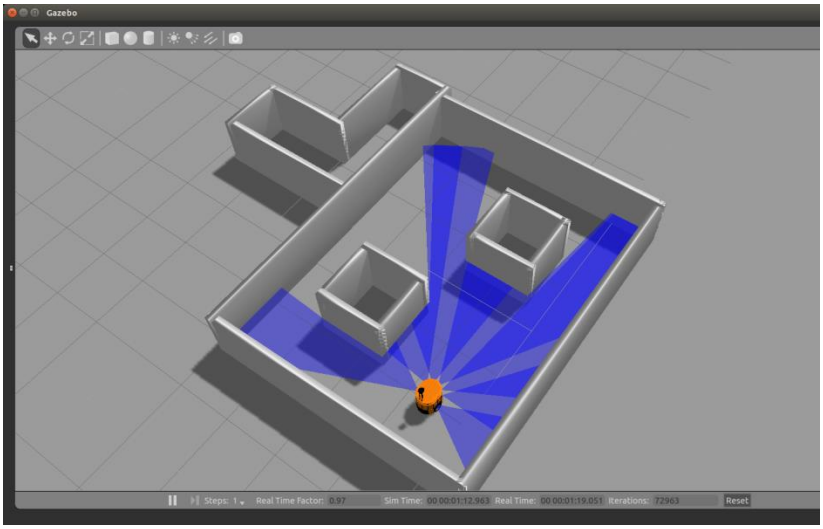
4 BULGULAR VE TARTIŞMA

Geliştirilen Evarobot gezgin robot platformu Gazebo benzetim ortamında ve gerçek ortamda test edilmiştir. Test ortamına ait görüntü Şekil 4.1'de verilmektedir. Test ortamı yaklaşık 30 m2'lik bir ortam üzerine kurulmuştur. Bu ortamda gezgin robotlar için temel davranışlar olan uzaktan kumandalı kontrol, rastgele dolaşma davranışı ve eş zamanlı konumlandırma ve haritalandırma test edilmiştir. Aynı ortam Gazebo benzetim ortamına aktarılarak aynı davranışlar burada da test edilmiştir.

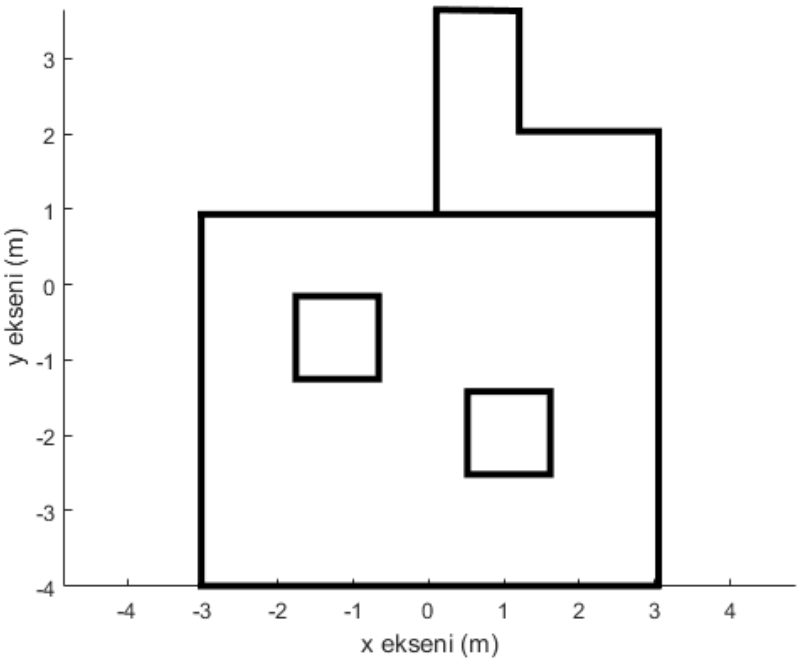
Şekil 4.1 Test ortamı

4.1 Benzetim Ortamı Testleri

Evarobot'un benzetim ortamı testleri, Şekil 4.1 ile verilen gerçek ortamın Şekil 4.3'deki gibi Gazebo'da oluşturulan modeli ile gerçekleştirilmiştir. Gazebo benzetim ortamında oluşturulan test ortamının krokisi Şekil 4.3'te verilmiştir. Bu bölümde uzaktan kumandalı kontrol, rastgele dolaşma davranışı ve eş zamanlı konumlandırma ve haritalandırma olmak üzere üç davranış üzerinden robotun işlevsellik testleri yapılmıştır.



Şekil 4.2 Gazebo benzetim test ortamı

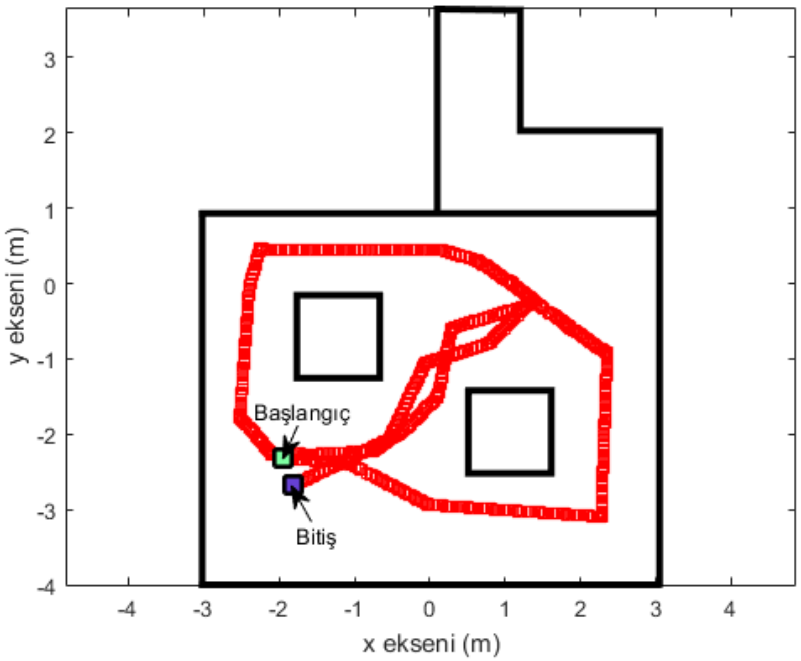


Şekil 4.3 Gazebo benzetim test ortamı krokisi

4.1.1 Uzaktan kumandalı kontrol

Evarobot'un Gazebo ortamındaki ilk testinde, platform uzaktan kumanda ile kontrol edilerek robotun benzetim ortamında hareketi sağlanmıştır. Test sırasında robotun izlediği yörünge Şekil 4.4'te verilmiştir. Robotun uzaktan kumandalı kontrolü için gereken referans hız değerleri "teleop_twist_keyboard" isimli düğüm tarafından klavyeden girilen

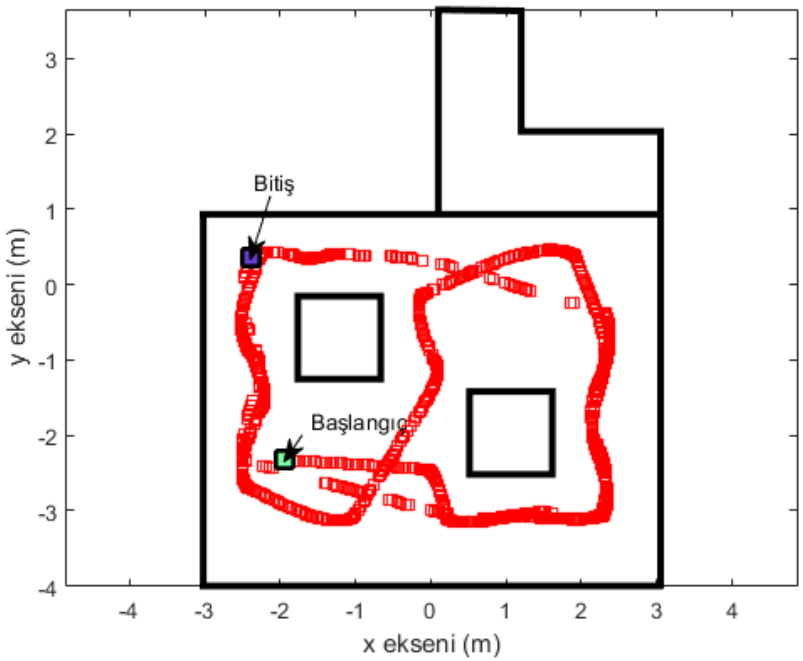
komutlara göre üretilmiştir. Referans hızların takibi için motorlar üzerindeki enkoderlerden okunan dönüş bilgilerinden elde edilen doğrusal hız değerleri sisteme geri besleme olarak verilmiştir. Bu test ile benzetim ortamında oluşturulan robot modelinin hareket kabiliyetinin kontrolü amaçlanmıştır. Test sonucunda benzetim ortamında oluşturulan robot modelinin hareketinde bir problem olmadığı ve referans hızlarına göre hareket ettiği gözlemlenmiştir.



Şekil 4.4 Benzetim ortamı uzaktan kumandalı kontrol izlenen yörünge

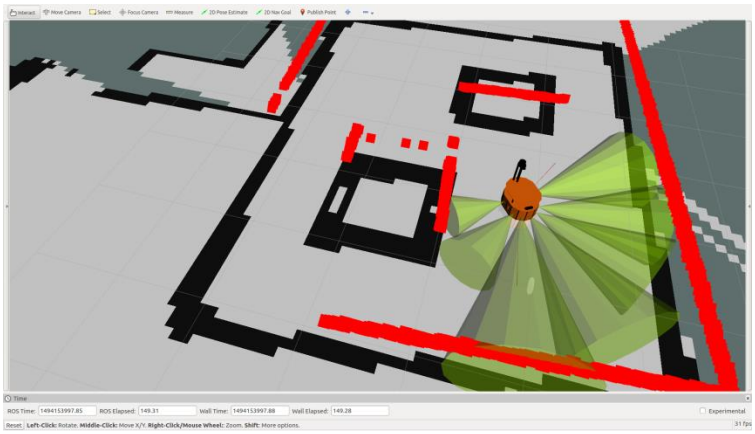
4.1.2 Rastgele dolaşma davranışı

Gazebo benzetim ortamında oluşturulan dünya modelinde Evarobot'un rastgele dolaşma davranışı test edilmiştir. Bu davranış potansiyel alanlar yöntemi kullanılarak geliştirilmiştir. Davranış sırasında robot üzerinde yedi adet sonar mesafe algılayıcılarından alınan mesafe bilgisi kullanılmıştır. Ayrıca enkoderler kullanılarak elde edilen hız bilgisi ile düşük seviyeli denetleyici üzerinden robota verilen referans hızlarının takibi sağlanmıştır.



Şekil 4.5 Benzetim ortamı rastgele dolaşma davranışı izlenen yol

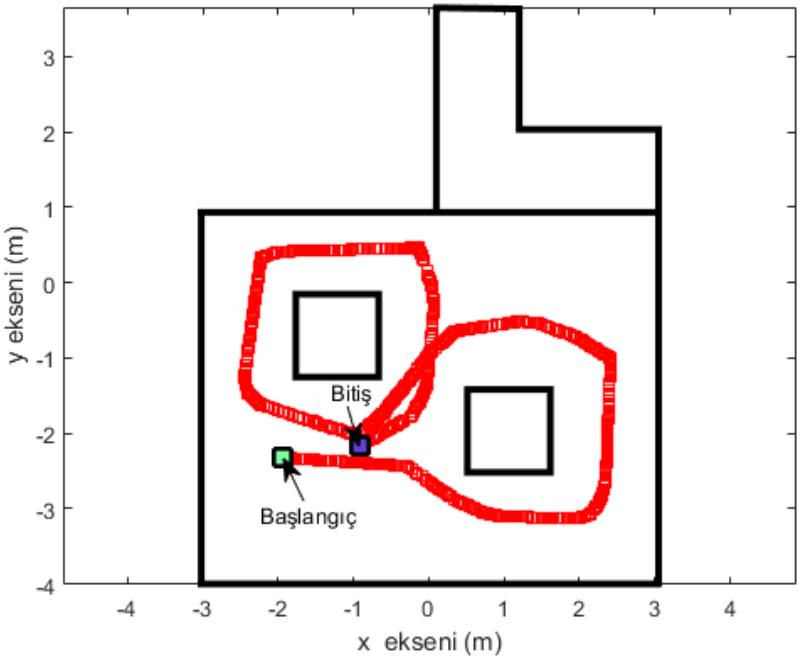
Bu davranış testi yapılırken robotun yüksek seviyeli kontrolcüsü kullanılmamaktadır. Bu sayede robotun sadece düşük seviyeli kontrol kısmı ile otonom hareket edebilirliğinin testi gerçekleştirilmiştir. Test sırasında robotun harita üzerinde izlediği yol Şekil 4.5'te verilmiştir. Ayrıca test sırasında robot üzerindeki algılayıcılardan elde edilen rviz görüntüsü Şekil 4.6'teki gibidir. Bu testin çıktısı olarak benzetim ortamı robot modeli ile düşük seviyeli kontrol kısmının bütünleşmiş bir şekilde çalıştığı doğrulanmıştır. Ayrıca benzetim ortamında kullanılan algılayıcılardan sonar ve enkoderin ürettiği değerlerin beklenen değerler olduğu gözlemlenmiştir.



Şekil 4.6 Benzetim ortamı rastgele dolaşma davranışı rviz görüntüsü

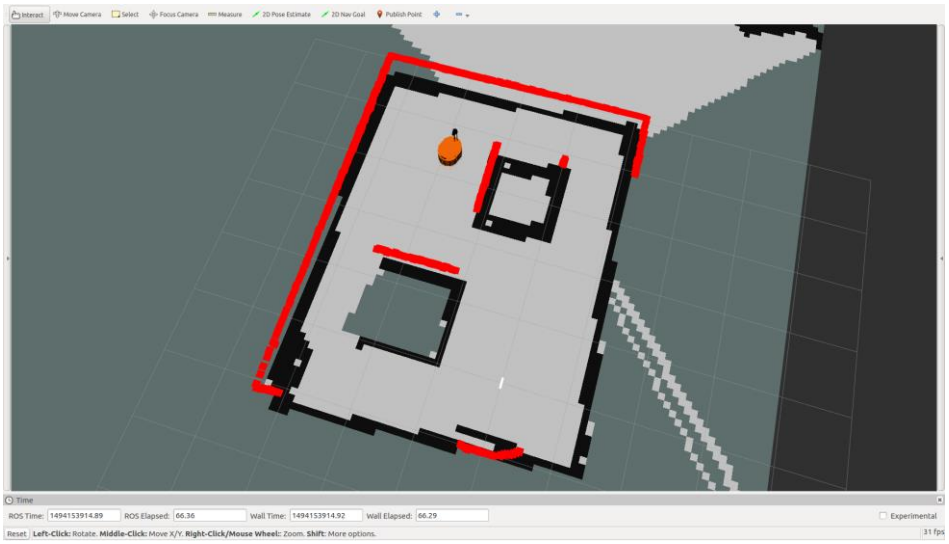
4.1.3 Eş zamanlı konumlandırma ve haritalama

Bu bölümde, Evarobot benzetim ortamı modeli kullanılarak yüksek seviyeli kontrol kısmında bahsedilen haritalama düğüm yapısının testi yapılmıştır. Önceki bölümlerde detaylı bir şekilde bahsedildiği üzere Evarobot'un kontrolü düşük ve yüksek seviyeli olmak üzere iki kısımdan oluşmaktadır. Robotun eş zamanlı konumlandırılması ve ortam haritasının çıkarıldığı bu testte bu iki kontrol kısmının birbirleri ile uyumlu çalışmaları kontrol edilmiştir. Deney ortamı üzerinde robotun izlediği yörünge Şekil 4.7'te verilmiştir.



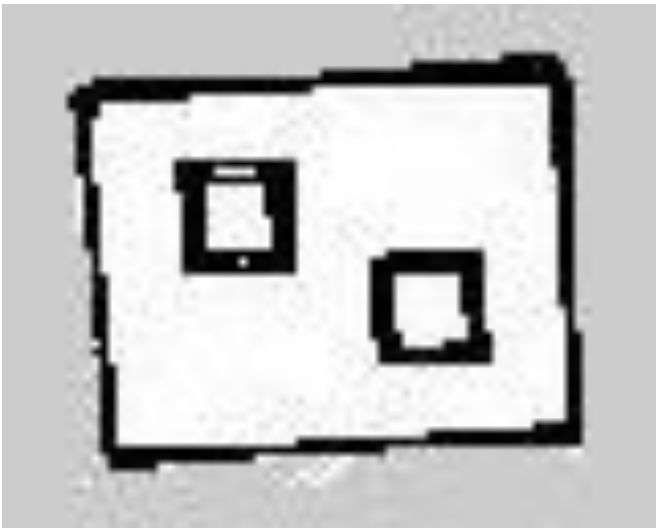
Şekil 4.7 Benzetim ortamı eş zamanlı konumlandırma ve haritalama izlenen yörünge

Test kapsamında ortam haritasının çıkartılması için lazer mesafe algılayıcısı kullanılmıştır. Algılayıcıdan elde edilen mesafe bilgileri, enkoder ve imu algılayıcılarından elde edilen konum bilgisi ile haritalama düğümünde birleştirilerek ortamın haritası çıkartılmıştır. Haritalama düğümü içerisinde "gmapping" isimli konumlandırma düğümü kullanılmıştır. Test sırasında anlık olarak çevre birimlerinden elde edilen verilen görselleştirilmesi işlemini gerçekleştiren rviz görüntüsü Şekil 4.8'dedir. Ayrıca elde edilen ortamın haritası Şekil 4.9'te verilmiştir.



Şekil 4.8 Benzetim ortamı eş zamanlı konumlandırma ve haritalama rviz görüntüsü

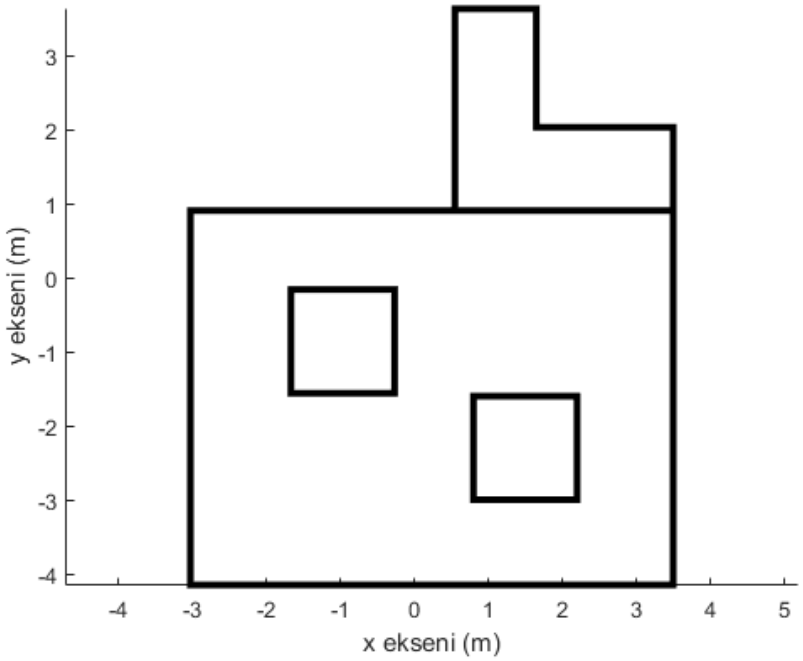
Gerçekleştirilen test sonucunda benzetim ortamındaki Evarobot modeli bütün bir sistem olarak test edilmiş ve işlevselliği kontrol edilmiştir. Bu test ile düşük ve yüksek seviye kontrollerin bütünleşmiş bir şekilde çalıştığı sonucuna varılmıştır. Test sırasında çıkartılan harita ile robot üzerindeki çevre birimleri arasındaki koordinat dönüşümlerinin başarılı bir şekilde gerçekleştirildiği gözlemlenmiştir.



Şekil 4.9 Benzetim ortamı çıkartılan ortam haritası

4.2 Gerçek Ortam Testleri

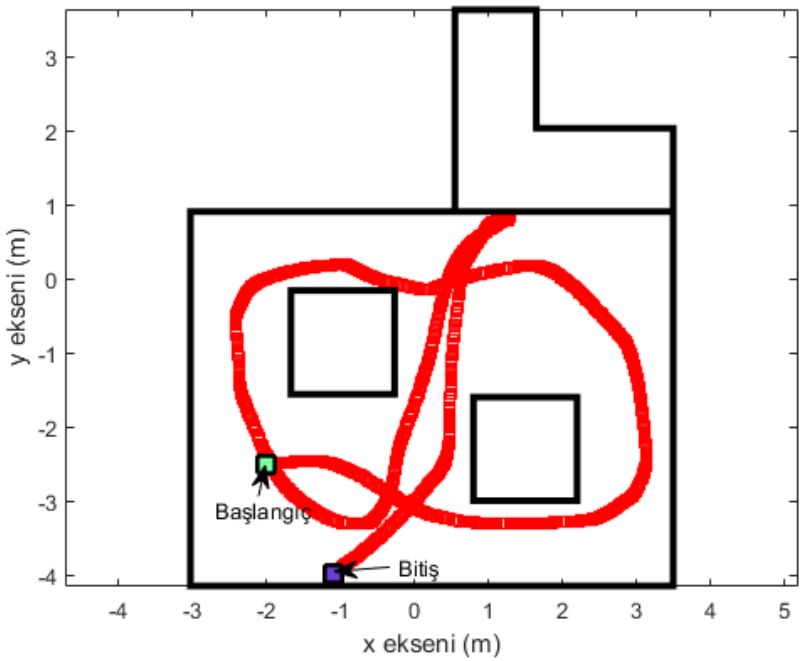
Gerçek ortam testleri Eskişehir Osmangazi Üniversitesi robotik laboratuvarında gerçekleştirilmiştir. Gerçek ortam testleri kapsamında kurulan deney ortamının krokisi Şekil 4.10'da verilmiştir. Benzetim ortamında üç davranış üzerinden yapılan işlevsellik testlerinin aynısı gerçek ortam üzerinde de uygulanmıştır.



Şekil 4.10 Gerçek test ortamı krokisi

4.2.1 Uzaktan kumandalı kontrol

Evarobot'un gerçek ortam testlerinden ilkinde robot uzaktan kumanda ile kontrol edilmiştir. Robotun uzaktan kontrolü için android işletim sistemli mobil cihazlar için geliştirilen yazılım kullanılmıştır. Robot üzerinde mobil cihazdan alınan komutları işleyerek robota uygulanacak referans hızlar haline getiren "evarobot_android" düğümü çalışmaktadır. Robotun test sırasında izlediği yörünge Şekil 4.11'da verilmiştir.



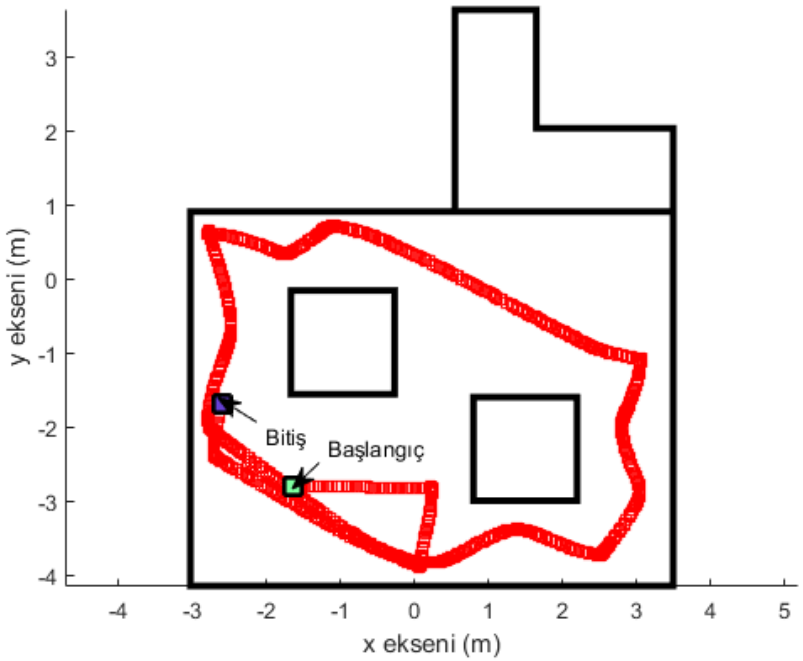
Şekil 4.11 Gerçek ortam uzaktan kumandalı kontrol izlenen yörünge

Test kapsamında robota kablosuz ağ üzerinden uygulanan referans hızlara göre başarılı bir şekilde hareket ettiği görülmüştür. Ayrıca robotun hareket etmesi için mekanik ve elektro-mekanik olarak hiçbir probleminin olmadığı tespit edilmiştir. Ayrıca yapılan test ile düşük seviyeli kontrol biriminin işlevselliği, bu kontrol birimi ile kablosuz olarak haberleşilebildiği ve enkoder algılayıcıların ürettiği değerler kontrol edilmiştir. Test sırasında robotun örnek bir görüntüsü Şekil 4.12'te verilmiştir.

Şekil 4.12 Gerçek ortam uzaktan kumandalı kontrol test ortamından görüntü

4.2.2 Rastgele dolaşma davranışı

Bu bölümde gerçek ortam üzerinde Evarobot'un rastgele dolaşması davranışı test edilmiştir. Potansiyel alanlar yöntemi kullanılarak yapılan bu davranış için benzetim ortamında yapılan testteki düğümün aynısı kullanılmıştır. Gerçek ortam testleri sırasında da yedi adet sonar algılayıcı kullanılmıştır. Ayrıca düşük seviyeli kontrolde geri besleme olarak alınan hız bilgileri için iki adet enkoder algılayıcıdan elde edilmiştir.



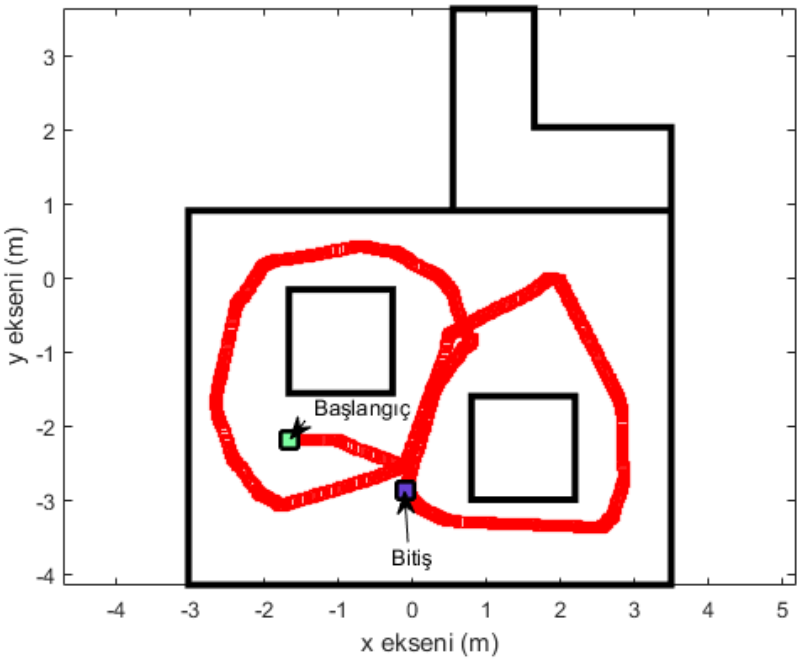
Şekil 4.13 Gerçek ortam rastgele dolaşma davranışı izlenen yörünge

Test sonunda geliştirilen sistemdeki düşük seviyeli kontrolün otonom davranış gerçeklenerek başarılı bir şekilde çalıştığı gözlemlenmiştir (Şekil 4.13). Düşük seviyeli kontrol düğümlerinin birbirleri arasındaki veri akışında ve çalışma durumlarında herhangi bir problem tespit edilmemiştir. Test sırasında kullanılan sonar ve enkoder algılayıcılardan okunan değerlerin doğruluğu kontrol edilmiştir. Evarobot'un rastgele dolaşma davranışı yaparken alınan örnek bir görüntüsü Şekil 4.14'te verilmiştir.

Şekil 4.14 Gerçek ortam rastgele dolaşma davranışı test ortamından görüntü

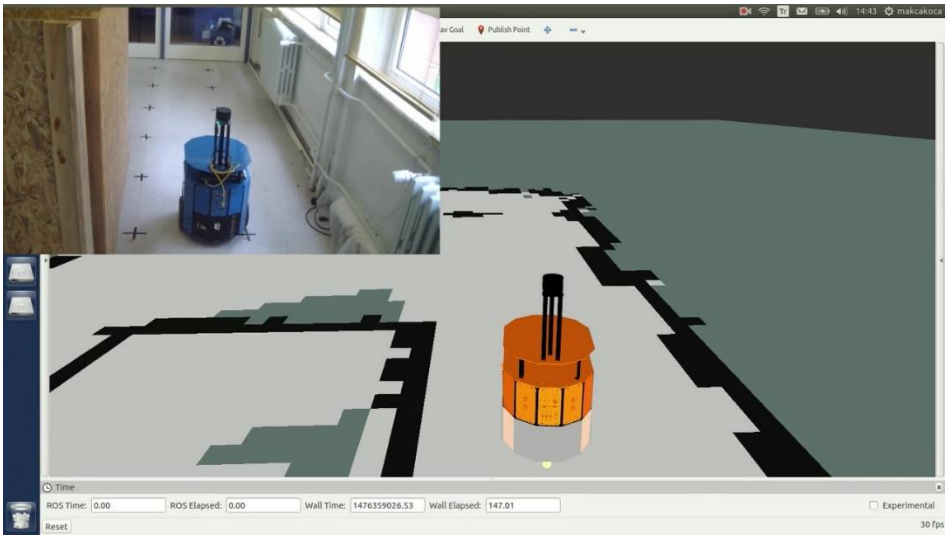
4.2.3 Eş zamanlı konumlandırma ve haritalama

Test kapsamında gerçek ortamda kurulan deney ortamının haritasının çıkarılması ve robotun eş zamanlı olarak harita üzerinde kendisini konumlandırması gerçekleştirilmiştir. Test süresi boyunca robotun izlediği yörünge Şekil 4.15'te kroki üzerinde belirtilmiştir.



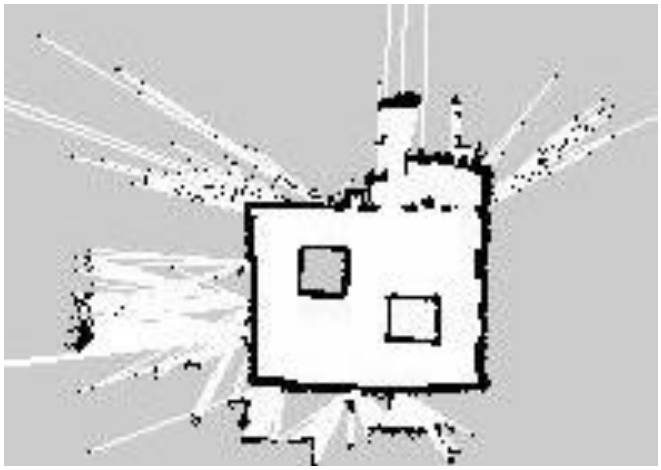
Şekil 4.15 Gerçek ortam eş zamanlı konumlandırma ve haritalama izlenen yörünge

Robot üzerindeki Rplidar 360 isimli lazer algılayıcı kullanılarak elde edilen derinlik bilgileri odometri bilgisi ile tümleştirilerek robotun anlık konumu tespit edilmiş ve ortamın haritası çıkartılmıştır. Ortam haritasının çıkarılması için haritalama düğümü olarak "gmapping" düğümü kullanılmıştır. Odometri bilgisi ise enkoder algılayıcılarından elde edilen konum bilgisinin imu algılayıcısından elde edilen yönelim bilgisi ile düzeltilmesinden sonra elde edilmektedir. Bu bilginin elde edilmesi için içerisinde genişletilmiş Kalman filtre çalışan "evarobot_pose_ekf" düğümü kullanılmıştır. Gerçek ortamda eş zamanlı konumlandırma ve haritalama yapılırken robotun rviz ve ortamdaki görüntüsü Şekil 4.16'te verilmiştir.



Şekil 4.16 Gerçek ortam eş zamanlı konumlandırma test ortamından görüntü

Gerçek ortamda eş zamanlı konumlandırma ve haritalama yapılarak yüksek ve düşük seviyeli kontrol birimlerinin tamamının bütünleşik bir şekilde testi bu bölümde başarı bir şekilde gerçekleştirilmiştir. Şekil 4.17'teki çıkartılmış ortam haritası göstermektedir ki algılayıcılardan alınan verilerin koordinat dönüşümleri ve bunların tümleştirilmesi işlemi doğru bir şekilde yapılmaktadır.



Şekil 4.17 Gerçek ortam çıkartılan ortam haritası

5 SONUÇ VE ÖNERİLER

Çalışma kapsamında uluslararası standartlarda eğitim ve araştırma amaçlı çalışmalarda kullanılmak üzere iç ortamlarda çalışan gezgin robot platformu geliştirilmiştir. Modüler ve tekrardan yapılandırılabilir bir şekilde tasarlanan ve gerçeklenen Evarobot için 3B benzetim ortamı Gazebo için de robotun kinematik ve dinamik modeli hazırlanmıştır. Kullanıcılar gerçek platformu temin etmeden de internetten benzetim modelini indirip bazı geliştirmeler yapabilirler.

Yazılım kontrol mimarisi düşük seviye ve yüksek seviyeli olmak üzere iki bileşenden oluşturulmuştur. Bu bileşenler ROS ara katmanı kullanılarak geliştirilmiştir. Ayrıca robotun MATLAB, Android vb. farklı yazılım ortamları üzerinden kontrolüne yönelik ilgili kütüphaneler geliştirilmiştir. Eğitim ve araştırmalarda kullanımı kolaylaştırmak için geniş çaplı kullanıcı materyalleri ve deney dokümanları hazırlanmıştır. Ayrıca bu materyaller ROS'un sitesinde açılan web sayfası üzerinden paylaşılmaktadır.

Geliştirilen Evarobot sistemi donanım ve yazılım olarak birçok farklı birimlerden oluşmaktadır. Bu birimlerin bir araya gelmesinin ardından tüm sistemin bütünleşik şekilde çalışırlığının kontrolü için Gazebo benzetim ortamında ve gerçek ortamda testler gerçekleştirilmiştir. Testlerde önce uzaktan kumandalı kontrol yapılarak benzetim ortamında Evarobot modelinin ve gerçek ortamda mekanik ve elektro-mekanik sistemlerin hareket sırasındaki testleri gerçekleştirilmiştir. Ardından rastgele dolaşma davranışı yapılarak Evarobot'un sadece düşük seviyeli kontrol kısmı test edilmiştir. Son olarak ise bütün sistemin bir arada testinin yapılması için eş zamanlı konumlandırma ve haritalama yapılmıştır. Yapılan testler ile ayrı ayrı ya da tüm sistemin bütünleşik bir şekilde çalıştığı gözlemlenmiştir.

KAYNAKLAR DİZİNİ

Abdessemed, F., Faisal, M., Emmadeddine, M., Hedjar, R., Al-Mutib, K., vd, 2014, A hierarchical fuzzy control design for indoor mobile robot, International Journal of Advanced Robotic Systems, 11(33), 1-16.

Ackerman, E., 2015, Bosch's giant robot can punch weeds to death, http://spectrum.ieee.org/automaton/robotics/industrial-robots/bosch-deepfield-robotics-weed-control , erişim tarihi: 21.11.2016.

Adept Technology, 2011, Pioneer 3-DX, http://www.mobilerobots.com/Libraries/Downloads/Pioneer3DX-P3DX-RevA.sflb.ashx, erişim tarihi: 01.10.2016, c.

Adept Technology, 2011, Autonomous Research PatrolBot, http://www.mobilerobots.com/Libraries/Downloads/PatrolBot-PTLB-RevA.sflb.ashx, erişim tarihi: 01.11.2016,a.

Adept Technology, 2011, Laser PowerBot, http://www.mobilerobots.com/Libraries/Downloads/PowerBot-PWRB-RevA.sflb.ashx, erişim tarihi: 01.11.2016, b.

Ando, N., Suehiro, T., Kotoku, T., 2008, A software platform for component based RT-system development: OpenRTM-Aist, Simulation, Modeling, and Programming for Autonomous Robots, 5325 LNAI, p.87–98.

Barber, R., Rodriguez-Conejo, M. A., Melendez, J., Garrido, S., 2015, Design of an infrared imaging system for robotic inspection of gas leaks in industrial environments. International Journal of Advanced Robotic Systems, 1-12.

Bischoff, R., Huggenberger, U., Prassler, E., 2011, KUKA youBot - A mobile manipulator for research and education, IEEE International Conference on Robotics and Automation, 3-6.

Bruyninckx, H., 2001, Open robot control software: the OROCOS project, IEEE International Conference on Robotics and Automation, 3, 2523–2528.

Bruyninckx, H., Soetens, S., 2007, https://people.mech.kuleuven.be/~orocos/pub/stable/documentation/rtt/current/doc-xml/orocos-overview.pdf, erişim tarihi: 21.11.2016.

Chafkin, M, 2016, Uber's first self-driving fleet arrives in Pittsburgh this month, http://www.bloomberg.com/news/features/2016-08-18/uber-s-first-self-driving-fleet-arrives-in-pittsburgh-this-month-is06r7on, erişim tarihi: 21.11.2016.

KAYNAKLAR DİZİNİ (devam)

Chang, W. C., Nguyen, V. T., Chu, P. R., 2012, Reconstruction of 3D contour with an active laser-vision robotic system, Asian Journal of Control, 14(2), 400–412.

Chang, H. J., Lee, C. S. G., Lu, Y. H., Hu, Y. C., 2007, P-SLAM: Simultaneous localization and mapping with environmental-structure prediction, IEEE Transactions on Robotics, 23(2), 281–293.

Cianci, C. M., Raemy, X., Pugh, J., Martinoli, A., 2007, Communication in a Swarm of Miniature Robots: The e-Puck as an Educational Tool for Swarm Robotics, Proceedings of the 2nd international conference on Swarm robotics (SAB'06), Berlin, Heidelberg, 103-115.

Clearpath Robotics, 2014, Turtlebot technical specifications, http://bit.ly/1L2FIzG, erişim tarihi: 01.10.2016.

Collett, T. H. J., MacDonald, B. A, Gerkey, B. P., 2005, Player 2.0: Toward a Practical Robot Programming Framework, In Proceedings of the Australasian Conference on Robotics and Automation.

Côté, C., Brosseau, Y., Létourneau, D., Raïevsky, C., Michaud, F., 2006, Robotic software integration using MARIE, International Journal of Advanced Robotic Systems, 3(1), 55-60.

Davies, A., 2016, Uber's self-driving truck makes its first delivery: 50,000 beers, https://www.wired.com/2016/10/ubers-self-driving-truck-makes-first-delivery-50000-beers/ , erişim tarihi: 21.11.2016.

Dillet, R, 2016, BMW, Mobileye and Intel are building a full self-driving car for 2021, https://techcrunch.com/2016/07/01/bmw-mobileye-and-intel-are-building-a-full-self-driving-car-for-2021/, erişim tarihi: 21.11.2016.

Dr Robot, 2013, X80SV Quick Start Guide, http://www.drrobot.com/products/item_downloads/X80SV_1.pdf, erişim tarihi: 01.11.2016.

Duc, T. N., Terracina, A., Mecella, M., Iocchi, L., 2012, Robotic Teaching Assistant for " Tower of Hanoi " Problem, European Conference on Games Based Learning 2, 723-732.

EasyMile, 2016, Shared Driverless Vehicles, http://easymile.com/, erişim tarihi: 21.11.2016.

KAYNAKLAR DİZİNİ (devam)

Egemin Automation, 2016, Automatic Guided Vehicles, http://www.egemin-automation.com/en/automation/material-handling-automation_ha-solutions/agv-systems, erişim tarihi: 22.11.2016.

Festo, 2016, Robotino - For research and education: Premium Edition and Basic Edition, http://www.festo-didactic.com/int-en/learning-systems/education-and-research-robots-robotino/robotino-for-research-and-education-premium-edition-and-basic-edition.htm?fbid=aW50LmVuLjU1Ny4xNy4xOC44NTguODAyNQ, erişim tarihi: 01.10.2016.

Fetch Robotics, 2016, Fetch Robotics User Manual, http://docs.fetchrobotics.com/FetchRobotics.pdf, erişim tarihi: 01.10.2016.

Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., Birattari, M., 2014, AutoMoDe: A novel approach to the automatic design of control software for robot swarms, Swarm Intelligence, 8(2), 89-112.

Gazebo, 2016, Plugins 101, http://gazebosim.org/tutorials/?tut=plugins_hello_world, erişim tarihi: 21.11.2016.

Google, 2016, Google self-driving car project, https://www.google.com/selfdrivingcar/, erişim tarihi: 21.11.2016.

Gritti, A. P., Tarabini, O., Guzzi, J., Di Caro, G. A., Caglioti, V., vd., 2014, Kinect-based people detection and tracking from small-footprint ground robots, IEEE International Conference on Intelligent Robots and Systems, (Iros), 4096–4103.

Guzel, M. S., Bicker, R, 2012, A behaviour-based architecture for mapless navigation using vision, International Journal of Advanced Robotic Systems, 9, 1–13.

Karasalo, M., Piccolo, G., Kragic, D., Hu, X., 2009, Contour reconstruction using recursive smoothing splines - Algorithms and experimental validation, Robotics and Autonomous Systems, 57(6-7), 617–628.

Kranz, M., Rusu, R. B., Maldonado, A., Beetz, M., Schmidt, A., 2006, A Player / Stage System for Context-Aware Intelligent Environments, System Support for Ubiquitous Computing Workshop (UbiSys 2006), 1–6.

Kuka Robotics, 2016, KUKA youbot Data Sheet, http://www.kuka-robotics.com/res/sps/9cb8e311-bfd7-44b4-b0ea-b25ca883f6d3_youBot_data_sheet.pdf, erişim tarihi: 01.11.2016.

KAYNAKLAR DİZİNİ (devam)

Lopez-Franco, M., Sanchez, E. N., Alanis, A. Y., Lopez-Franco, C., Arana-Daniel, N., 2014, Discrete-time decentralized inverse optimal neural control combined with sliding mode for mobile robots, World Automation Congress Proceedings, 496–501.

Mondada, F., 2015, EPFL education robot, http://www.e-puck.org/, erişim tarihi: 22.11.2016.

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., vd., 2006, The e-puck , a Robot Designed for Education in Engineering. Robotics, Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions, 59-65.

Nesnas, I., Wright, A., Bajracharya, M., Simmons, R., Estlin, T., vd., 2003, CLARAty: An architecture for reusable robotic software, Proc. SPIE 5083, Unmanned Ground Vehicle Technology V, 253-264.

Prieto, A., Becerra, J. A., Bellas, F., Duro, R. J., 2010, Open-ended evolution as a means to self-organize heterogeneous multi-robot systems in real time. Robotics and Autonomous Systems, 58(12), 1282–1291.

Quigley, M., Conley, K., Gerkey, B., FAust, J., Foote, T., vd., 2009, ROS: an open-source Robot Operating System, ICRA Workshop pn Open Source Software.

Ray, A. K., Behera, L., Jamshidi, M., 2008, Sonar-based rover navigation for single or multiple platforms: Forward safe path and target switching approach, IEEE Systems Journal, 2(2), 258–272.

Robotics Technology Consortium, 2013, A roadmap for U.S. robotics: from internet to robotics, 129 p.

Ross, P. E., 2016, A ride in Ford self-driving car, http://spectrum.ieee.org/cars-that-think/transportation/self-driving/a-ride-in-fords-selfdriving-car, erişim tarihi: 21.11.2016.

Straßberger, D., Mercorelli, P., Georgiadis, A., 2014, On the Functional Controllability Using a Geometric Approach together with a Decoupled MPC for Motion Control in Robotino, 21st International Symposium on Mathematical Theory of Networks and Systems, Groningen, The Netherlands,1080–1087.

Swisslog, 2016, Automated guided vehicle systems, http://www.swisslog.com/en/Products/WDS/Automated-Guided-Vehicles, erişim tarihi: 22.11.2016.

KAYNAKLAR DİZİNİ (devam)

Systems, E., 2014, A Virtual Laboratory of a Manufacturing Plant operated with Mobile Robots, IEEE Engineering. Emerging Technology and Factory Automation, 1–4.

Tesla Motors, 2016, Autopilot Tesla, https://www.tesla.com/autopilot, erişim tarihi: 21.11.2016.

The Orocos Project, 2016, http://www.orocos.org/, erişim tarihi: 01.10.2016.

Thomas, D., 2014, ROS Introduction, http://wiki.ros.org/ROS/Introduction, erişim tarihi: 21.11.2016.

Toyota Forklifts, 2016, Material handling and warehouse lift equipment, https://www.toyotaforklift.com/, erişim tarihi: 22.11.2016.

Utz, H., Sablatnög, S., Enderle, S., Kraetzschmar, G., 2002, Miro - Middleware for mobile robot applications, IEEE Transactions on Robotics and Automation, 18(4), 493-497.

Volvo Car, 2016, Self driving Volvos by 2017, http://www.volvocars.com/au/about/innovations/intellisafe/autopilot, erişim tarihi: 21.11.2016.

Weinert, H., Pensky, D., 2011, Mobile robotics in education and student engineering competitions, IEEE AFRICON Conference, 13–15.

Wu, K., Ranasinghe, R., Dissanayake, G., 2015, Active recognition and pose estimation of household objects in clutter, IEEE International Conference on Robotics and Automation (ICRA), 4230–4237.

Yazici, A., Kirlik, G., Parlaktuna, O., Sipahioglu, A., 2014, A dynamic path planning approach for multirobot sensor-based coverage considering energy constraints, IEEE Transactions on Cybernetics, 44(3), 305–314.