

Simulink Kullanılarak ROS Tabanlı Gezgin Robot Kontrol Benzetimi

¹Yavuz Selim Fatihoglu

¹Kocaeli Üniversitesi Teknoloji Fakültesi Bilişim Sistemleri Mühendisliği ,
Umuttepe Kampüsü, Kocaeli, Türkiye.

Özet

Gezgin robot sistemleri alanında araştırmalar gün geçtikçe artmaktadır. Gerçek robotlarla, bu geliştirme ortamını kurmak zaman kaybına ve zamanla robotların yıpranması ve maliyet kaybına sebep olmaktadır. Bu çalışmada turtlebot gezgin robotun matlab Simulink ile oluşturulmuş bir kontroller model kullanılarak ROS üzerinden kontrolü gerçekleştirilmiştir.

Anahtar kelimeler: Gezgin robot, ROS-Robot Operating System, Simulink, Matlab.

1. Giriş

Gezgin robot sistemleri alanında araştırmalar gün geçtikçe artmaktadır. Gezgin robotların günümüzde, gelişmiş üretim sistemleri, sağlık sektörü, dış ortamlarda otonom araçlar, uzay keşifleri, maden arama, tarım, su altı ve kurtarma sistemleri, güvenlik uygulamaları, biyomedikal uygulamalar gibi geniş bir alanda uygulamalarını görmekteyiz. Gezgin robotlarda, yöntemler geliştirilirken uygulama ve uygulama sonuçlarına göre yöntemin geliştirilmesi durumu söz konusudur. Gerçek robotlarla, bu geliştirme ortamını kurmak zaman kaybına ve zamanla robotların yıpranması ve maliyet kaybına sebep olmaktadır. Bu çalışmada turtlebot gezgin robotun matlab Simulink ile oluşturulmuş bir kontroller model kullanılarak ROS üzerinden kontrolü gerçekleştirilmiştir.

2. ROS (Robot Operating System)

ROS(Robot Operating System) yani robot işletim sistemi, içerisinde barındırdığı kütüphaneler ve araçlarla robot uygulamaları geliştirmeyi sağlayan bir yazılım sistemidir. ROS bir robot için meta-işletim sistemidir [1]. ROS, açık kaynak kodlu bir yazılımdır. Linux tabanlı bir işletim sistemi üzerinde çalışmaktadır.

ROS bir işletim sistemi olmasının ötesinde robot donanımı ile yazılım arasındaki bir katman ya da çalışma çerçevesi (framework) şeklinde düşünülebilir. ROS endüstriyel dünya ve akademik dünyanın çalışma beraberliğini arttırdığı, profesyonel yazılım geliştirme ve araştırma sonuçlarının beraber harmanlandığı yeni bir yaklaşımdır [2].

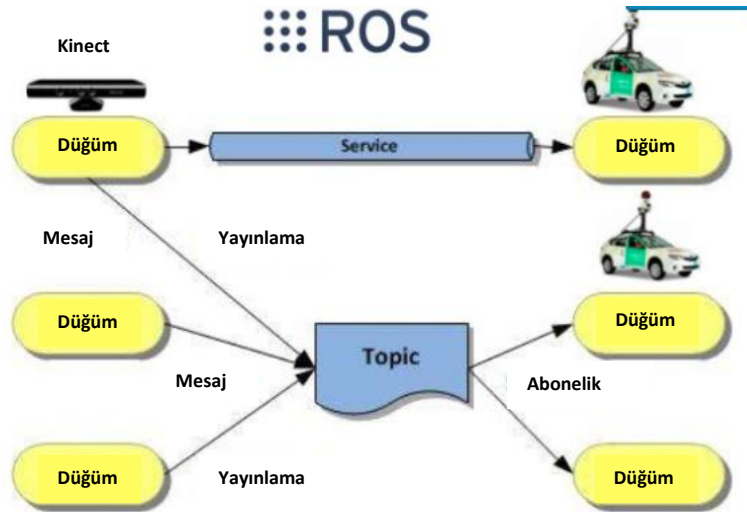
Bu framework içinde tanımlı kütüphaneler, servisler aracılığı ile farklı robot platformlarının kontrolü sağlanabilir ayrıca algoritma geliştirme, kod yazma (C++, Python vb.) imkanları da sağlamaktadır. ROS'un farklı dağıtımları mevcuttur [3] ve farklı robot platformlarını

¹E-posta: yselim@kocaeli.edu.tr, telefon: +902623032295

desteklemektedir [4]. Bu çalışmada Linux işletim sistemi tabanlı Ubuntu 14.04 Trusty LTS(Long Term Support-Uzun Vade Destek) dağıtımını üzerine kurulu ROS Indigo dağıtımını kullanılmıştır.

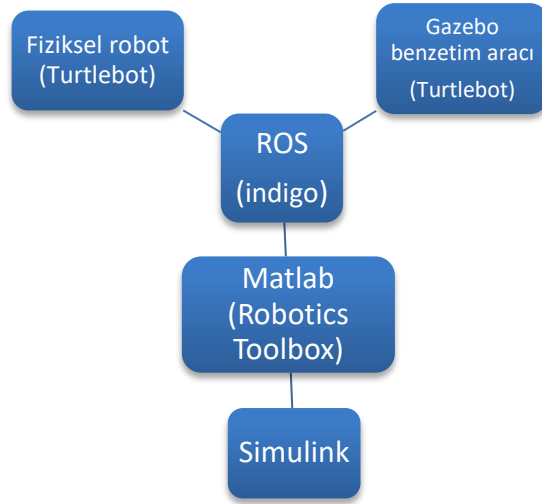
ROS, yayın gönderme (publish) ve bu yayınlara ilgili düğümlerin bu yayına abone olması (subscribe) temelli bir paradigma üzerinde kurulmuş bir sistemdir. Hazırlanan robot davranışları node(düğüm) adı verilen program parçaları üzerinden simülasyon programına yada robota gönderilir (publish), hareket için gerekli veriler çevreden alınır.

Publish ve subscribe ilişkisi şu şekildedir ; bir uygulama(publishers) belli bir konu üzerindeki mesajı yayınlıyor , diğer uygulamalar(subscribers) da yayınlanan mesajlara, konu ya da içerik tabanlı kuralları kullanarak abone olur. Bu form birden-çokta(one-to-many) iletişim şeklindedir ve özünde eşzamanlı değildir çünkü mesajı yayınlayan bir uygulama diğer alıcı uygulamalar tarafından geri dönüş olmasını beklemeyebilir. ROS yapısı Şekil 1’de gösterildiği gibi çalışmaktadır.



Şekil 1. ROS çalışma şeması

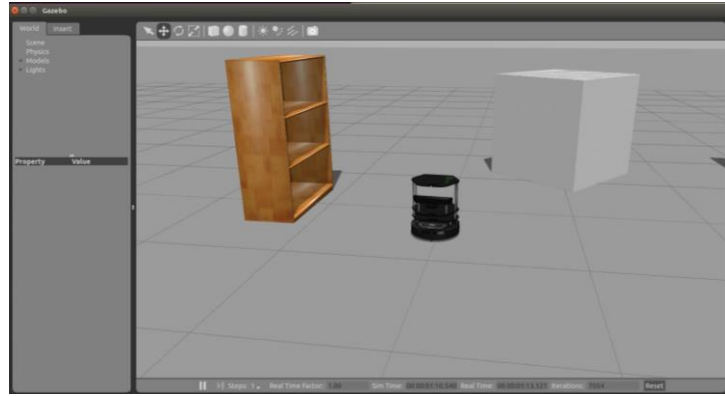
Publisher(Yayınlayıcı) ve subscriber(Abone), ROS' da kullanılan bir modeldir. Oluşturulan sınıflar üzerinden veri simülasyon programına gönderilir(publish), robotun aksesuarlarından sensör verileri program içerisine alınabilir(subscribe).



Şekil 2. Matlab – ROS – Turtlebot/Gazebo bağlantı şeması

3. ROS ve Gazebo Benzetim Aracının Yapılandırılması

Gazebo benzetim aracı Player projesi içinde yer alan 3-boyutlu dinamiği içeren çoklu robot benzetim ortamıdır. Stage yazılımına benzer şekilde, robotları ve nesnelere benzetimini gerçekleştirmektedir. Ancak, 3D bir ortamda, nesnelere arası fiziksel etkileşimleri gösterebilen bir benzetim yazılımıdır [5]. Bu çalışmada Linux sistem üzerine ROS Indigo dağıtımını ile kullanılmak üzere Gazebo 2.2 sürümü kurulmuş ve yapılandırılmıştır.



Şekil 3. Benzetim yapılan modelin Gazebo örnek ekran görüntüsü

ROS kurulu sistemde ROS katmanının robot donanımı ile iletişime geçmesi ve komutları gönderebilmesi için çevre değişkenlerinin ayarlanması gereklidir. ROS_MASTER_URI ve ROS_HOSTNAME çevre değişkenleri aşağıdaki komutlar kullanılarak *.bashrc* dosyası içinde set edilmelidir.

```

> echo export ROS_MASTER_URI=http://ip_numarasi:11311 >> ~/.bashrc
> echo export ROS_HOSTNAME=ip_numarasi >> ~/.bashrc
  
```

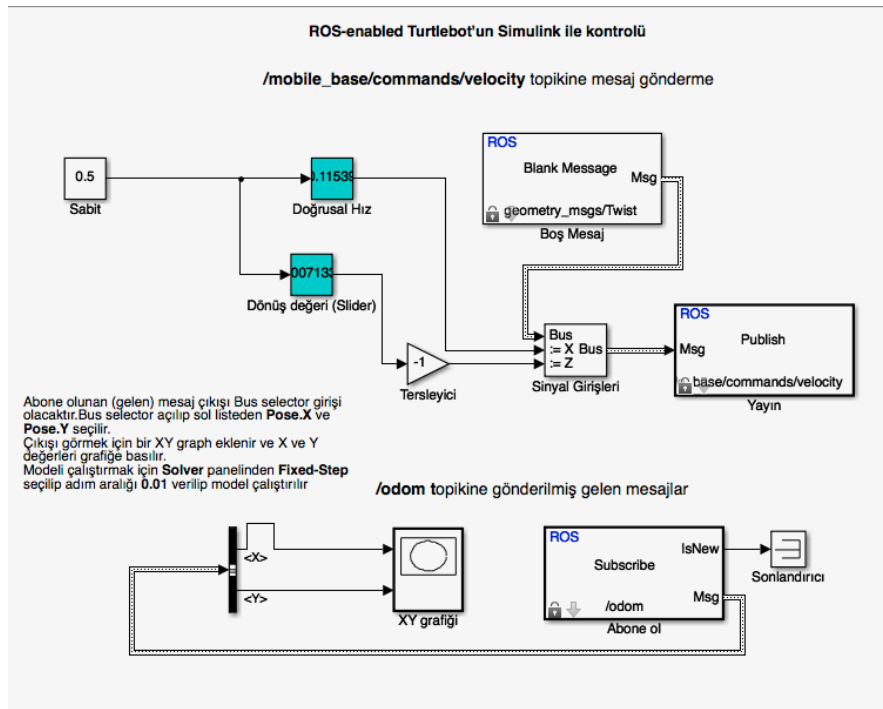
ROS temel yapılandırılması tamamlandıktan sonra yeni terminal penceresinde Gazebo benzetim aracı aşağıdaki komut çalıştırılarak gezgin robot kontrol için hazır hale getirilir.

```
> roslaunch turtlebot_gazebo turtlebot_world.launch
```

4. Matlab ROS bağlantısı ve Simulink kontrol modeli

MATLAB, Robotic System Toolbox ile ROS katmanı için destek sağlamaktadır. MATLAB kullanılarak oluşturulacak prototip algoritmalar ROS desteği ile Gazebo vb. benzetim araçlarında test edilebilir ya da doğrudan fiziksel robot platformu üzerinde çalıştırılabilir. Matlab Robotik Sistem kütüphanesindeki *rosinit* komutu ile ROS tabanlı gezgin robotun IP numarası verilerek global bir düğüm oluşturulur. Simulink modeli yayınlanan mesajları, abone olunan mesajları ve servisleri bu IP numarası üzerinden gönderir yada alır.

Simulink kullanılarak oluşturulan modeller ile fiziksel yada simulator bazlı uygulamalar geliştirilebilir. Bu makalede turtlebot gezgin robot platform kontrolü için Şekil 4'te gösterilen Matlab Simulink üzerinde tasarlanan model kullanarak ROS tabanlı bir kontrol uygulaması gerçekleştirilmiştir.



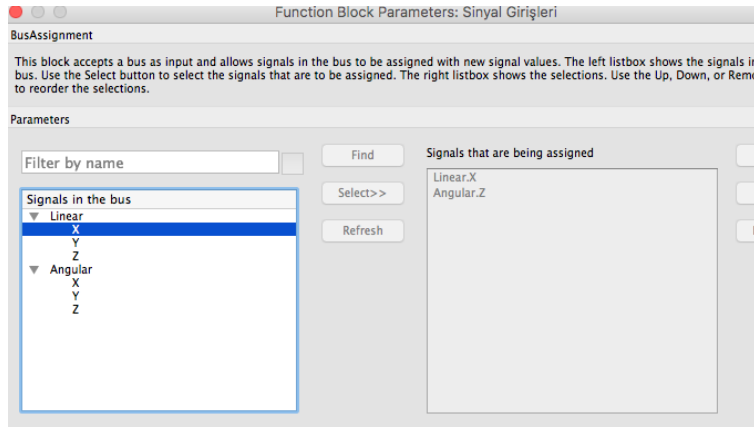
Şekil 4. Simulink kontrol modeli

5. Simulink Modelinin Uygulanması

Model üç temel kısımdan oluşmaktadır. İlk kısımda bir ROS mesajı üretilmektedir. ROS mesajı robotun seyir esnasında izlediği yolun x ve y koordinat değerlerini tutar. İkinci kısımda üretilen

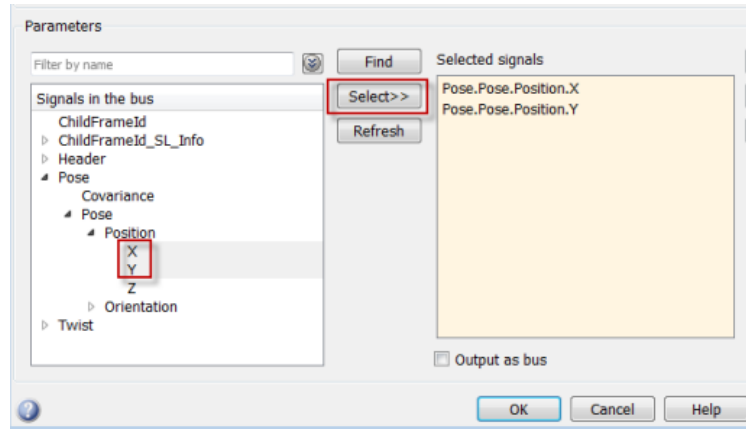
mesajlara abone (subscriber) olacak olan kısım tanımlıdır. Bu değerler ROS ağına yayımlanacak değerlerdir. Bu kısımların çalışma prensipleri sırasıyla aşağıda verilmiştir.

ROS mesajını oluşturmak için simulink modelinde bir sinyal tanımlamak gereklidir. Dönüş değeri ve doğrusal hız değerleri sinyal atama bloğuna girer. Ayrıca boş mesaj bloğu da sinyal atama bloğuna girer, atama bloğu çıkışı ROS yayın(publish) bloğuna atanır. Şekil 5'te gösterildiği gibi doğrusal X ve açısal Z değeri yeni sinyal değeri olarak atanırlar. Doğrusal hız değeri ve açısal dönüş değerleri mesaj türü "geometry/Twist" olacak şekilde ROS ağındaki "mobile_base/commands/velocity" başlığa(topic) gönderilir. Gezgin robotun yayımlanan mesaj içindeki açısal ve doğrusal hız değerleri aracılığı ile kontrolü sağlanır.



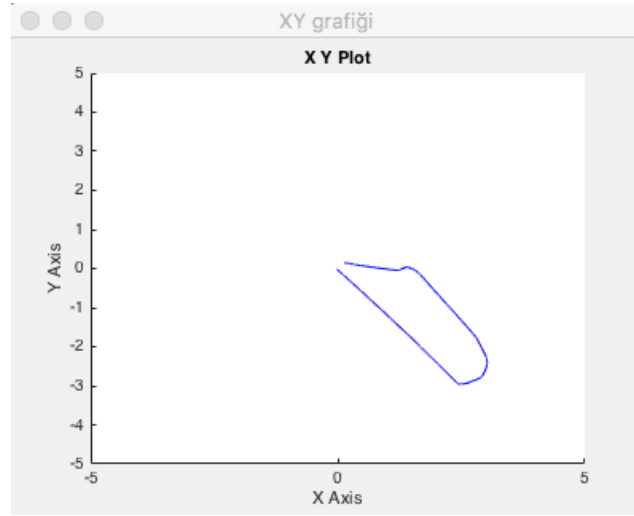
Şekil 5. Doğrusal hız ve Dönüş eksen değerleri atama bloğu

Robot seyri esnasında robotun yol durumunu takip edip XZ düzlemi üzerindeki konumunu grafik olarak göstermek için robotun "/odom" başlığına abone olunur. Bu başlığa abone olmak için Şekil 4'te gösterildiği gibi bir abone bloğu eklenir.



Şekil 6. Robot konum değerleri atama ekranı

ROS ağı üzerinden "/odom" başlığına gelen değerler Şekil 6'da gösterildiği gibi POS.X ve POS.Y sinyallerine atanırlar. Abone bloğu çıkışındaki mesaj grafik bloğuna giriş olarak gönderilir.



Şekil 7. Gezgin Robot seyir grafiği

Şekil 7’de gösterildiği gibi robotun seyir esnasındaki konum koordinatları ve konum grafiği XY düzleminde çizdirilir.

6. Sonuç

Bu çalışmada ROS ağına gönderilen yada alınan mesajlar ve başlıklar üzerinden Gazebo aracı kullanılarak gezgin robot için kontrol benzetimi gerçekleştirilmiştir. ROS robot donanımı ile geliştirilecek yol planlama ve kontrol algoritması arasında köprü vazifesi görecektir. Matlab ve Simulink kullanılarak fiziksel robot üzerinde yada benzetim araçlarında yeni kontrol ve yol planlama algoritmaları gerçekleştirilebilir.

7. Referanslar

- [1] Pajaziti A., Avdullahu P., SLAM – Map Building and Navigation via ROS, International Journal of Intelligent Systems and Applications in Engineering, 2014, p. 71-75.
- [2] S. Cousins, Welcome to ROS Topics, IEEE Robotics & Automation Magazine, March 2010.
- [3] <http://wiki.ros.org/Distributions>, en son ziyaret tarihi Eylül 2016.
- [4] <http://wiki.ros.org/Robots>, en son ziyaret tarihi Eylül 2016.
- [5] Özkan M., Yazıcı A., Parlaktuna O., Otonom Robot Sistemleri için Eğitim Araçları, 2009.