

Robot Guidance with Head Movements

Versiyon : v3.0

Tarih : 18/10/2020

Hazırlayan : Elif GENÇ

DÖKÜMAN REVİZYON SAYFASI

| VERSIYON | TARİH | SAYFA | AÇIKLAMA |
|----------|------------|-------|---|
| v1.0 | 31/08/2020 | 10 | Ubuntu Bionic Beaver ve Ros Melodic Kurulumu, Ros Beginner Tutorials |
| v2.0 | 06/09/2020 | 14 | Publisher/Subscriber ve Service/Client yapılarının Turtlesim üzerinde örnekleri |
| v3.0 | 18/10/2020 | 11 | Turtlebot3 Waffle Pi modelinin kurulması, Gazebo’da bir simülasyon ortamı tasarlanması,Turtlebot3 Waffle Pi modelinin bu ortamda çağrılmasını sağlayacak başlatma dosyalarının hazırlanması ve robotun bu ortamda klavye ile kontrolünün sağlanması |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

İÇİNDEKİLER

| | |
|---|-----------|
| 1. TANITIM VE KAPSAM | 4 |
| 2. KULLANILAN MATERYALLER | 4 |
| 3. ÖNERİLEN YÖNTEM / PROBLEM ÇÖZÜM AŞAMALARI | 4 |
| 4. GERÇEKLENEN GÖREVİN AŞAMALARI | 5 |
| 5. SONUÇLAR | 10 |
| 6. KAYNAKÇA | 11 |

1. TANITIM VE KAPSAM

TurtleBot 3, eğitim, araştırma, hobi ve ürün prototip oluşturmada kullanım için küçük, uygun fiyatlı, programlanabilir, ROS tabanlı bir mobil robottur. TurtleBot 3'ün amacı, platformun boyutunu önemli ölçüde küçültmek ve işlevselliği ve kalitesinden ödün vermek zorunda kalmadan fiyatı düşürmek ve aynı zamanda genişletilebilirlik sunmaktır. TurtleBot3, mekanik parçaları nasıl yeniden yapılandırdığınıza ve bilgisayar ve sensör gibi isteğe bağlı parçaları nasıl kullandığınıza bağlı olarak çeşitli şekillerde özelleştirilebilir. Ek olarak, TurtleBot 3, sağlam gömülü sistem, 360 derece mesafe sensörü ve 3D baskı teknolojisi için uygun, uygun maliyetli ve küçük boyutlu SBC ile geliştirilmiştir. TurtleBot3'ün temel teknolojisi SLAM, Navigasyon ve Manipülasyondur, bu da onu ev hizmeti robotları için uygun hale getirir. TurtleBot, bir harita oluşturmak için SLAM (eşzamanlı yerelleştirme ve haritalama) algoritmalarını çalıştırabilir ve odanızda dolaşabilir. Ayrıca bir dizüstü bilgisayar, joypad veya Android tabanlı akıllı telefonla uzaktan kontrol edilebilir. TurtleBot, bir kişinin bir odada yürürken bacaklarını da takip edebilir. Ayrıca TurtleBot 3, Open MANIPULATOR gibi bir manipülatör ekleyerek bir nesneyi manipüle edebilen bir mobil manipülatör olarak kullanılabilir. [1]

Gazebo, ücretsiz ve açık kaynak kodlu robot simülasyon ortamıdır. Bir ve birden fazla robotun 3 boyutlu ortamda simüle edilebildiği bir programdır. V-REP gibi DART, Simbody, ODE, Bullet kütüphanelerinin kullanılması ile fiziksel ortamı işlemesi kolaylaştırılmıştır. Gazebo genelinde server-client mantığı ile çalışmaktadır. Gzserver (sunucu) somut işlevleri yaparken (robot ortamda dolaşırken), Gzclient (istemci) kullanıcının isteklerinin yerine getirilmesi ve simülasyonun görsele aktarılmasını sağlar. Veritabanındaki robotlarla sınırlı kalmayıp, kendi robot ve dünyamızı yaratma imkânı bize tanınmıştır. Eğer istenilirse model.config ve model.sdf dosyalarının içindeki XML kodlarını değiştirerek fiziksel robotlara istenilen sensör ve kamera ekleyebilme şansımız vardır. [2]

Bu çalışma kapsamında istenilen TurtleBot 3 Waffle Pi modelinin kurulması, Gazebo' da 5x5 metrelik bir labirent simülasyonunun hazırlanması, TurtleBot 3 Waffle Pi robotunun Gazebo' da hazırlanan labirent üzerinde başlatılması ve klavye yardımıyla robotun yönlendirilmesinin sağlanması gerekmektedir.

2. KULLANILAN MATERYALLER

Bu görev kapsamında simülatör olarak Gazebo ve Rviz uygulamaları, robot simülasyonu için TurtleBot 3 Waffle Pi modeli kullanılmıştır. Labirent çiziminde mapping olarak kullanılan .png uzantılı bir resim dosyası kullanılmıştır. [3] Ağ yapılandırma işlemleri için net-tools paketi indirilmiş ve bu paketin sağladığı “\$ ifconfig” komutu ile ağ bilgilerine ulaşılmıştır.

Bu simülasyon için gerekli TurtleBot 3 dosyalarına github üzerinden ulaşılmıştır ve bu dosyalar klonlanmıştır.

3. ÖNERİLEN YÖNTEM / PROBLEM ÇÖZÜM AŞAMALARI

Bu aşamada 2 farklı sorunla karşılaşılmıştır. Gerçeklenen görev içerisinde ilk karşılaşılan problem “\$ sudo apt-get install ros-melodic-joy” komutunun çalıştırılamamasıdır. Bu problem, yapılan araştırmalar sonucunda ROS paketlerinin güncel olmadığından kaynaklanan bir problem olduğu anlaşıldı. Daha sonra bu problemi nasıl çözeceğimiz ele alındı ve ROS paketlerini güncelleyebilmek için “\$ sudo apt-get update” ve “\$ sudo apt-get upgrade” komutları kullanılarak sistem ve ROS paketleri güncellenmiştir ve problem ortadan kalkmıştır.

Daha sonra yapılan görev kapsamında ağ yapılandırma işlemi sırasında bilgisayara ait ağ bilgilerine erişebilmek için “\$ ifconfig” komutunun çalıştırılmamasıyla karşılaşmıştır. Bu problemin çözümü için gerekli araştırmalar sonucunda “net-tools” paketinin eksikliğinden kaynaklandığı anlaşıldı, bu paketin kurulması için “\$ sudo apt install net-tools” komutu kullanıldı ve problem ortadan kaldırıldı.

4. GERÇEKLENEN GÖREVİN AŞAMALARI

- İlk olarak ROS’un güncellenmesi gerekmektedir. Güncelleme işleminin yapılmasını sağlayan komutlar: [4]

```
$ sudo apt-get update
```

```
$ sudo apt-get upgrade
```

- TurtleBot 3’ ü bilgisayar üzerinde kontrol edebilmek için bazı paketlerin kurulması gerekmektedir. Bu işlem için kullanılan komut: [4]

```
$ sudo apt-get install ros-melodic-joy
```

```
$ cd ~/catkin_ws/src/
```

```
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git
```

```
$ git clone -b melodic-devel https://github.com/ROBOTIS-GIT/turtlebot3.git
```

Yukarıdaki komutlar çalıştırdıktan sonra TurtleBot 3’ ün hazırlanmasını tamamlamak için “catkin_ws” klasörü aşağıdaki komut ile derlenir.

```
$ cd ~/catkin_ws && catkin_make
```

- TurtleBot ile bilgisayar arasında uzaktan iletişim kurulması için ağ ayarlarının yapılandırılması gerekmektedir. Bu ayarlar yapılandırdıktan sonra TurtleBot ile bilgisayar arasında iletişim kurulabilmesi için aynı wifi yönlendiricisine bağlanması gerekmektedir. Bu yüzden ağ bilgilerine ulaşabilmemiz için çalıştırılan komut: [4]

```
$ ifconfig
```

- Ardından bilgisayarın ip adresi öğrenildikten sonra bashrc isimli dosyayı açtıktan sonra gerekli eklemeler yapıldı ve dosya kaydedildi. [4]

```
$ gedit ~/.bashrc
```

```
export ROS_MASTER_URI=http://192.168.1.2:11311
```

```
export ROS_HOSTNAME=192.168.1.2
```

```
$ source ~/.bashrc
```

- “turtlebot3_fake_node” isimli düğümü kullanabilmek için “turtlebot3_simulation” isimli meta pakete ihtiyacımız vardır. Bu paketin kurulumu için gerekli komutlar: [5]

```
$ cd ~/catkin_ws/src/
```

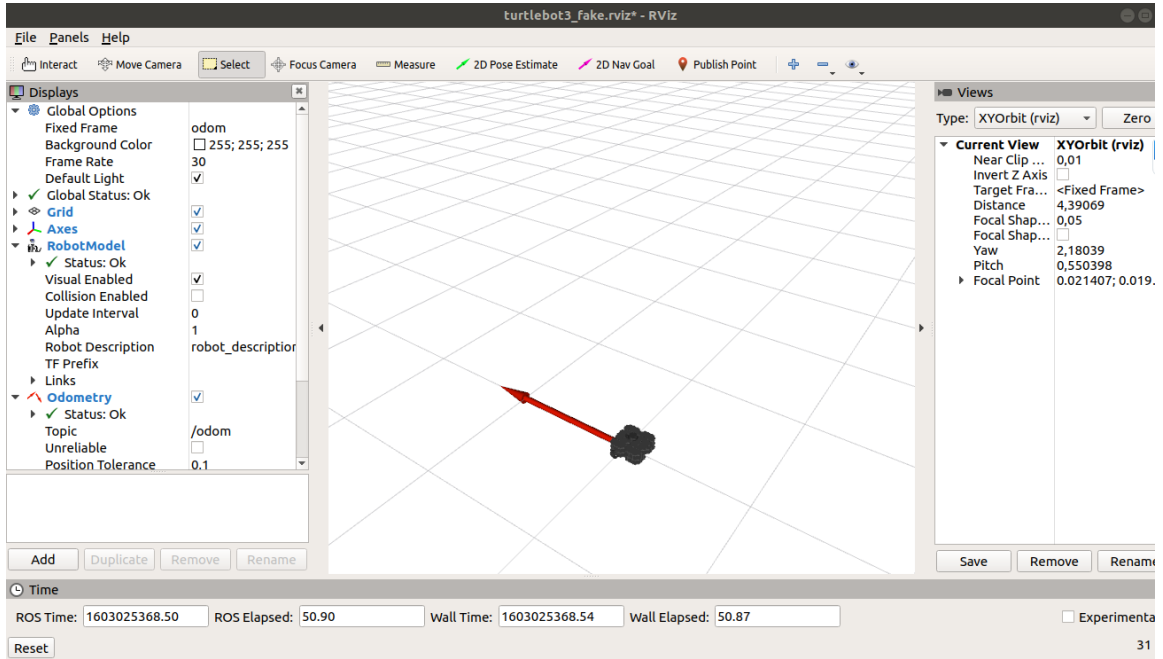
```
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations.git
```

```
$ cd ~/catkin_ws && catkin_make
```

- “turtlebot3_fake” isimli paket gerçek robot olmadan çalışan çok basit bir simülasyon düğümüdür. [5]

```
$ export TURTLEBOT3_MODEL= waffle_pi
```

```
$ roslaunch turtlebot3_fake turtlebot3_fake.launch
```

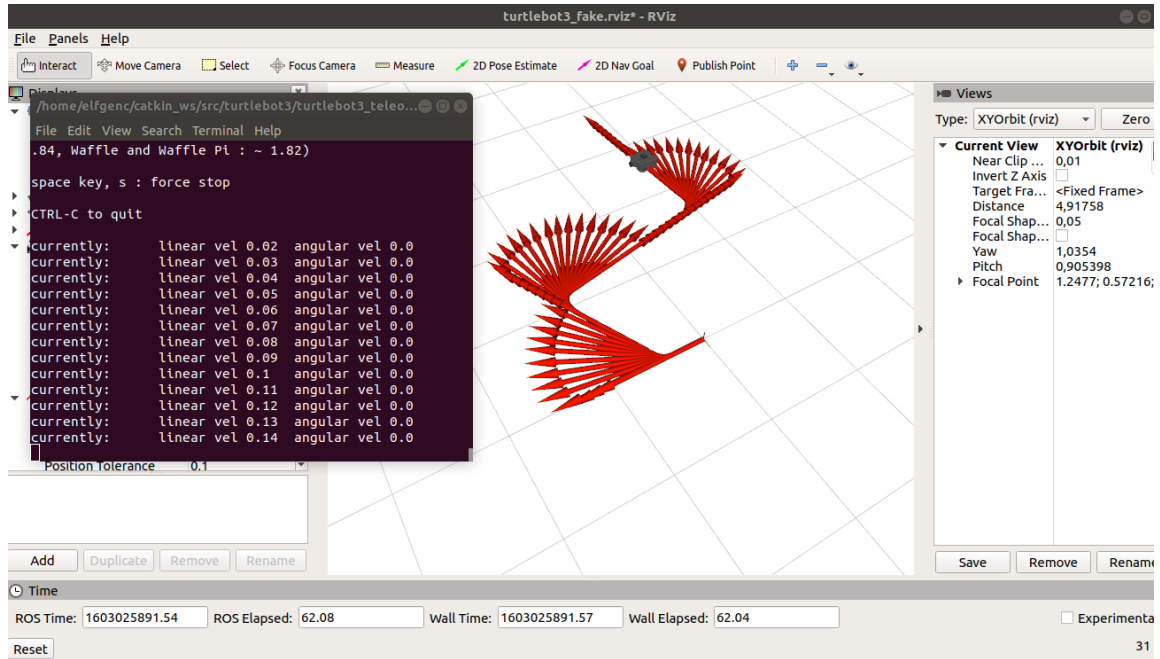


Şekil 1: Rviz Uygulamasında TurtleBot 3

Ardından robotun klavye ile kontrolünün sağlanması için gerekli komutlar: [5]

```
$ export TURTLEBOT3_MODEL= waffle_pi
```

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```



Şekil 2: Rviz Uygulamasında Gösterilen TurtleBot 3' ün Hareketleri

- Öncelikle catkin_ws/src klasörü altında “my_simulations” isimli bir simülasyon klasörü ve dosyalarını oluşturmak için gerekli komutu yazdıktan sonra dosyalar derlendi.

```
$ cd catkin_ws
```

```
$ cd src
```

```
$ creat catkin_create_pkg my_simulations
```

```
$ cd ..
```

```
$ catkin_make
```

- “my_simulations” isimli kalsör içerisine “launch” ve “worlds” isimli klasörler oluşturuldu. Daha sonra “launch” isimli klasör içerisine “my_world.launch” dosyası ve “worlds” isimli klasör içerisine “empty_world.world” dosyası oluşturuldu.
- “catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch” isimli klasör içerisindeki turtlebot3_empty_world.launch dosya içeriği kopyalanıp “catkin_ws/src/my_simulation/launch/my_world.launch” dosyasına kaydedildi. Ardından dosya içerisinde bulunan “<arg name=“world_name” value=“\$(find turtlebot3_gazebo)/worlds/empty.world”/>” kod, “<arg name=“world_name” value=“\$(find my_simulations)/worlds/empty_world.world”/>” kodu ile değiştirildi. Dosya içeriği XML olarak kaydedildi.
- Ardından “catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/worlds” klasörü içerisinde bulunan empty.world dosya içeriği kopyalandı ve “catkin_ws/src/my_simulation/worlds/empty_world.world” dosyasına kaydedildi. Daha sonra dosya içeriği XML olarak kaydedildi.
- Daha sonra terminal ekranı açılarak catkin_ws klasörüne gidilir ve bu klasör derlenir.

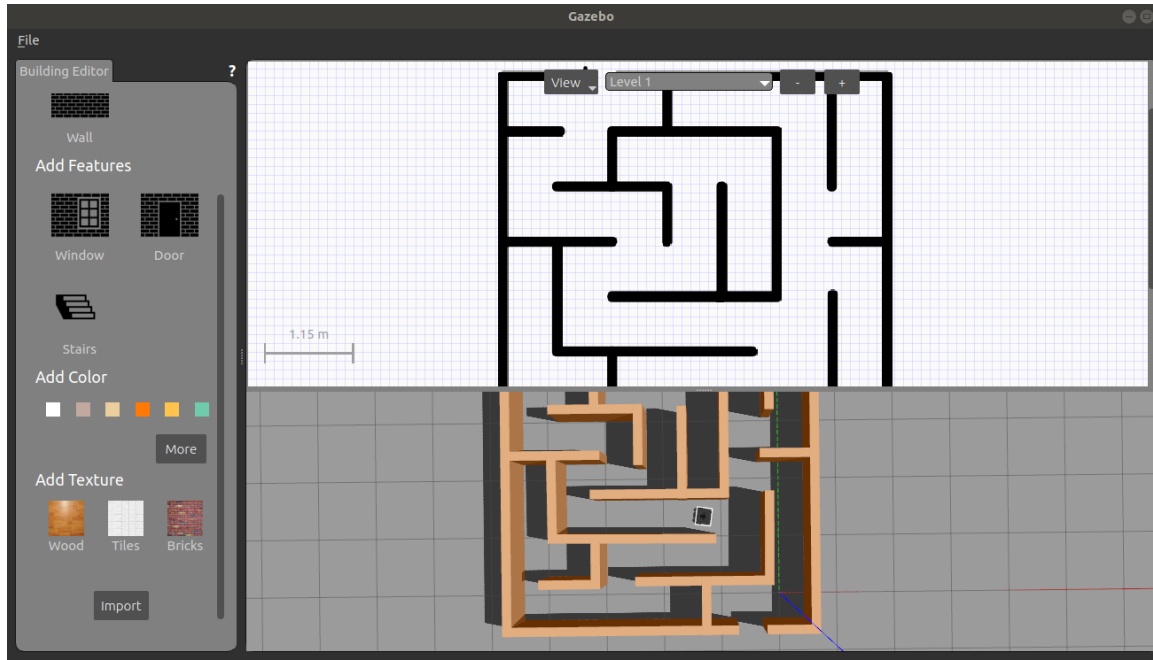
```
$ cd catkin_ws
```

```
$ catkin_make
```

- Derleme işleminden sonra “roscore” çalıştırılıp yeni bir terminal açılır. Bu terminal üzerinde kendi oluşturduğumuz simülasyon dosyaları çalıştırılır. [5]

```
$ roslaunch my_simulations my_world.launch
```

- Gazebo uygulaması açıldıktan sonra kendi simülasyonumuzu özelleştirmek için uygulama içerisinde bulunan “edit>building editor” seçilerek labirentin oluşturulmasına başlanır. Labirenti bir resim üzerinden oluşturmak için “import” seçeneği ile bilgisayarda bulunan .png uzantılı bir labirent resmi seçilir. [5]



Şekil 3: Labirent Çizimi

- Resim üzerindeki çizgilerden faydalanarak 5x5 metrelik bir labirent platformu oluşturuldu. Labirentin özellikleri istenilen şekilde ayarlandı. Oluşturulan labirent modelini kaydetmek için `file>save` as seçeneği seçilerek “catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/models” pathine “my_maze” isimli klasör olarak kaydedildi. Bu klasör içerisinde model.sdf ve model.config dosyaları yer almaktadır. Bu dosyalar bizim labirentimizi temsil etmektedir. [5]
- Daha sonra empty_world.world isimli oluşturduğum dosya içeriği açılarak oluşturduğum labirentin dosya ismi, dosya içerisine şu şekilde eklenmiştir ve robotun ilk bulunması gereken koordinat “pose” ile dosya içeriğine eklenmiştir. [5]

```
<!-- my_maze -->
```

```
<include>
```

```
<pose>-2 2 0 0 0 0</pose>
```

```
<uri>model://my_maze</uri>
```

```
</include>
```

- Daha sonra catkin_ws isimli klasöre giderek “catkin_make” komutu ile klasör derlenir. Ardından “roscore” komutu çalıştırılır ve yeni terminal açılır. Labirent simülasyonun başlatılması için çalıştırılan komutlar: [5]

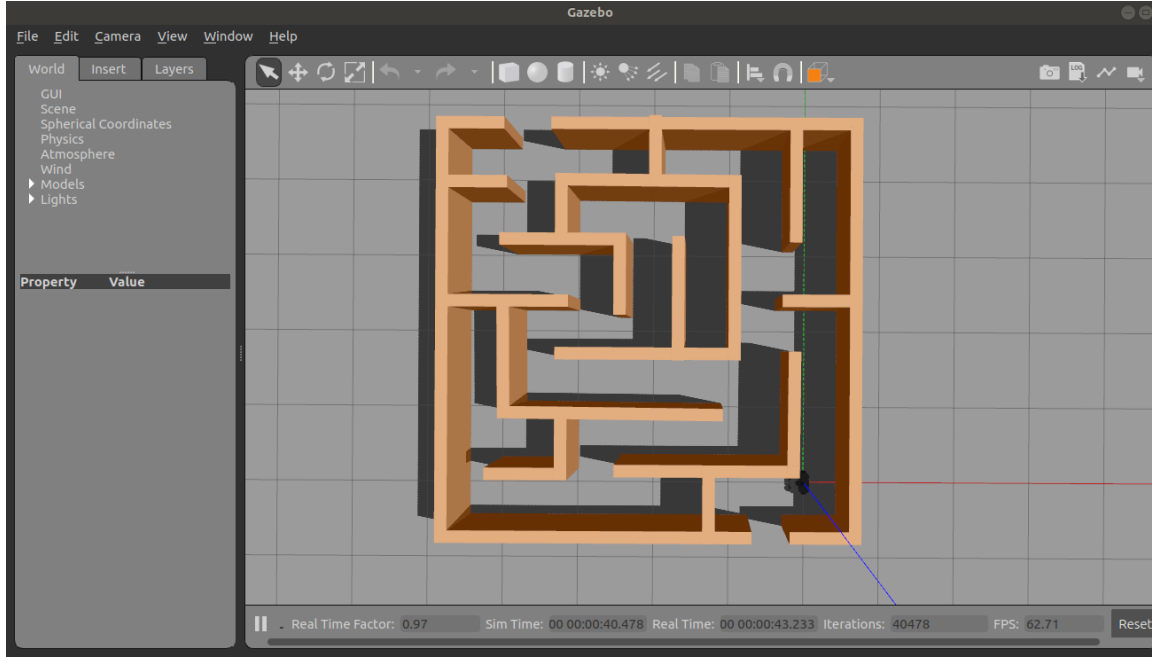
```
$ export TURTLEBOT3_MODEL=waffle_pi
```

```
$ roslaunch my_simulations my_world.launch
```

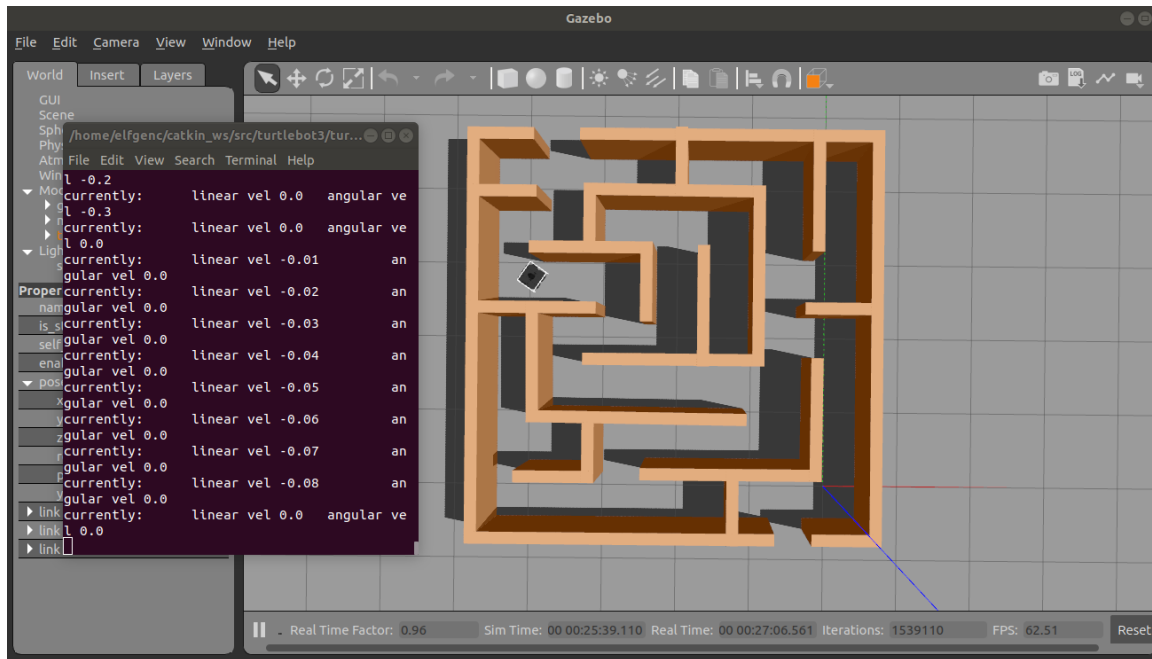

- Bu labirent içerisinde robotun hareket edebilmesi için yeni bir terminal ekranını açılır. Bu terminal ekranında robotun hareket etmesini sağlayan komutlar çalıştırılır. [5]

```
$ export TURTLEBOT3_MODEL=waffle_pi
```

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```



Şekil 4: TurtleBot 3' ün Başlangıç Noktası



Şekil 5: TurtleBot 3' ün Yönlendirildikten Sonra Bulunduğu Konum

5. SONUÇLAR

Yapılan çalışmalar kapsamında sistem ve Ros paketlerinin güncel kalmasının yapılan işlemler için gerekli olduğu, bazı komutların çalışmamasının gerekli olan paketlerin eksikliğinden kaynaklanabileceğini, ağ yapılandırma işlemlerinin nasıl yapıldığı ve ağ bilgilerine nasıl erişilmesi gerektiğini, TurtleBot 3 Waffle Pi modelinin kurulumunu, TurtleBot 3 ile basit bir Rviz simülasyonun nasıl açıldığı ve bu uygulama üzerinde robotu klavye ile kontrol edebilmeyi, Gazebo uygulaması üzerinde 5x5 metrelik bir labirent ortamının tasarlanması ve bu tasarım içerisinde uygulama arayüzünün incelenmesi, TurtleBot 3 Waffle Pi modelinin kendi oluşturduğumuz labirent ortamı içerisinde kullanmayı sağlayacak launch dosyasının hazırlanması, ve TurtleBot 3 Waffle Pi modelinin labirent ortamında klavye ile kontrol edilerek labirentin içerisinde gezebilmesi sağlanmıştır. Bu kapsamda yapılan her aşama ayrıntılı olarak anlatılmıştır ve öğrenilmiştir.

6. KAYNAKÇA

- [1] <https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/> [Erişim tarihi: Ekim, 2020]
- [2] <https://medium.com/@cennttceylnn/gazebo-nedir-ros-nedir-19983c017818> [Erişim tarihi: Ekim, 2020]
- [3] <https://pngio.com/images/png-a619904.html> [Erişim tarihi: Ekim, 2020]
- [4] https://emanual.robotis.com/docs/en/platform/turtlebot3/pc_setup/#install-ubuntu-on-remote-pc [Erişim tarihi: Ekim, 2020]
- [5] <https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/#turtlebot3-simulation-using-fake-node> [Erişim tarihi: Ekim, 2020]