

# BLM6191 Robotlar

## Ödev 1

Dr. Öğr. Üyesi Erkan Uslu  
euslu@yildiz.edu.tr

Yıldız Teknik Üniversitesi  
Bilgisayar Mühendisliği  
21/10/2024

### Giriş

Gazebo simülasyon ortamında iki farklı labirent dünyası (maze1 ve maze2) ve turtlebot3 oluşturulmuştur. Turtlebot3 /odom mesajı üzerinden dünyadaki **mutlak konumunu**, /scan mesajı üzerinde ise yaptığı lazer ölçümlerini sunmaktadır.

Turtlebot3'e /cmd\_vel mesajı üzerinden gövde çizgisel  $x$  hızı ve yine gövde açısız  $z$  hızı komutları iletilmektedir.

Turtlebot3, labirent dünyasında rastgele olarak labirentin bir köşesinde ( $(7.5m, 7.5m)$  veya  $(-7.5m, 7.5m)$  veya  $(-7.5m, -7.5m)$  veya  $(7.5m, -7.5m)$  koordinatlarında), dışı doğru bakacak şekilde oluşturulmaktadır. Labirentin boyutu  $16m \times 16m$  olarak verilmektedir. Labirentin merkezi  $x = 0, y = 0$  noktasıdır.

Labirenti başarılı bir şekilde tamamlayabilmek için merkezi  $x = 0, y = 0$  koordinatlarında olan bir kenarı  $2m$  olan kareye ulaşmak gerekmektedir.

Labirent dünyalarını ve turtlebot3'ü oluşturan paketi çalıştırmak için aşağıdaki adımları takip ediniz (micromouse\_maze paketinin README.md dosyasını inceleyin):

#### Command Line

```
$ cd ~/robotlar_ws/src
$ git clone https://gitlab.com/blm6191_2425b/blm6191/micromouse_maze.git
$ cd ~/robotlar_ws
$ catkin_make
$ source ~/.bashrc

$ echo "export TURTLEBOT3_MODEL=waffle" >> ~/.bashrc
$ source ~/.bashrc

$ echo "export GAZEBO_MODEL_PATH=${GAZEBO_MODEL_PATH}~/robotlar_ws/src/" >> ~/.bashrc
$ source ~/.bashrc

$ roslaunch micromouse_maze micromouse_maze4.launch
```

Gazeboda açılan dünya Şekil 1 ile verilmiştir.

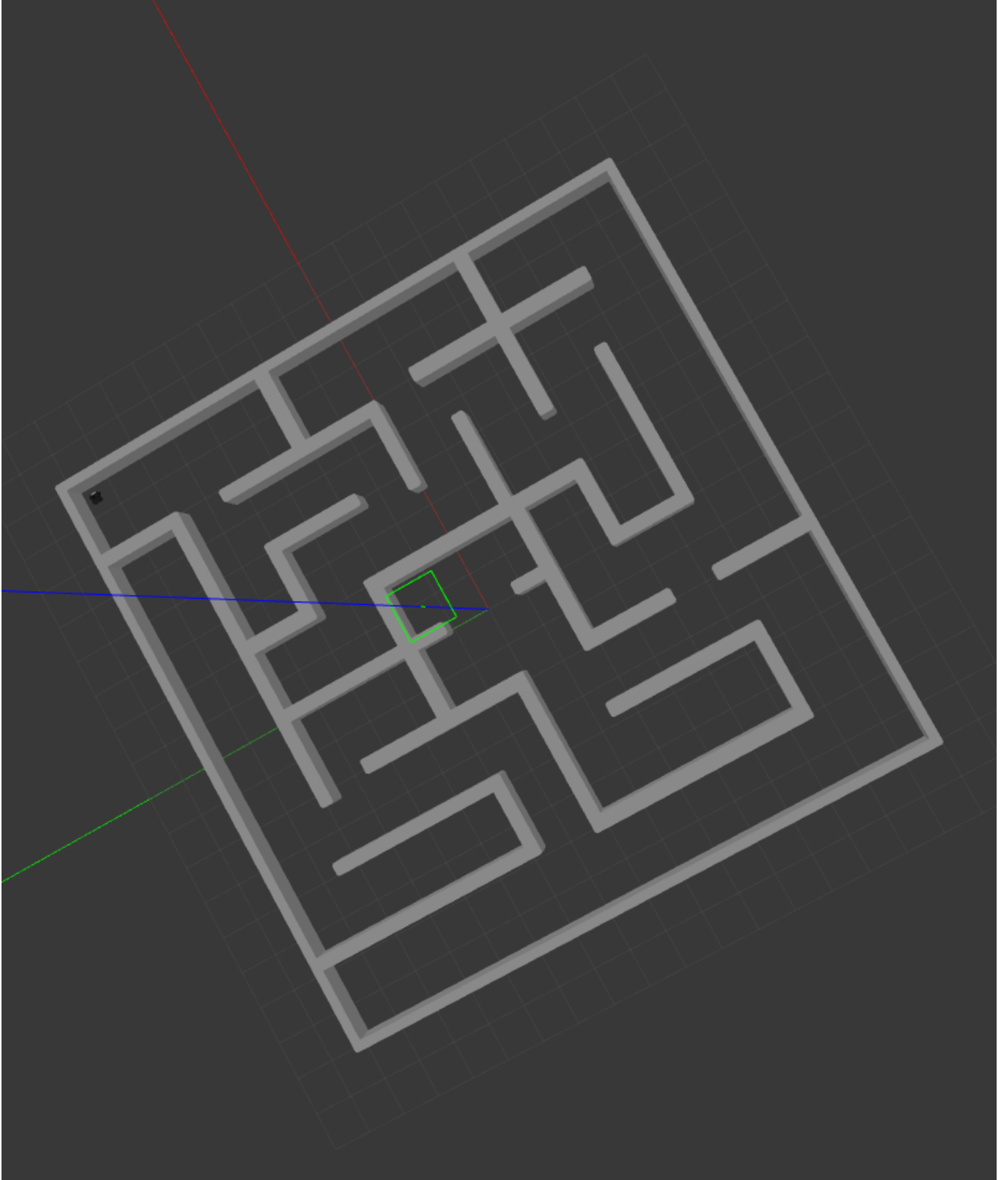
/odom mesajı Gazebo'nun referans koordinat sistemine (Gazebo'nun (0,0) noktasına) göre robotun /base\_footprint isimli çerçevesinin pozisyonunu vermektedir.

Sisteme ilişkin tf ağacı Şekil 2 ile verildiği gibidir.

Gazebo simülasyonu başlatıldıktan sonra aşağıdaki komut ile tf ağaç yapısı görüntülenebilir.

#### Command Line

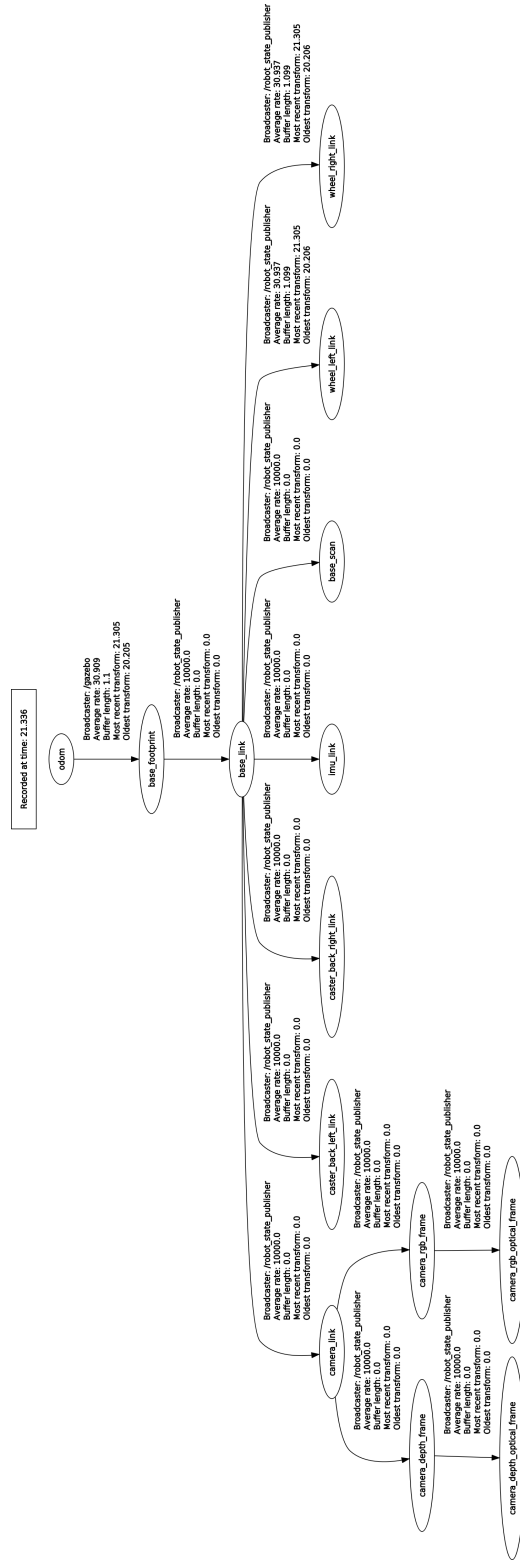
```
$ rosrn rqt_tf_tree rqt_tf_tree
```



Şekil 1: Gazebo dünyası ve robot

Bu tf ağacında base\_footprint çerçevesi ile base\_scan çerçevesi arasındaki sabit dönüşüm Şekil 3 ile verilmiştir.

Gazebo simülasyonu başlatıldıktan sonra base\_footprint çerçevesi ile base\_scan çerçevesi arasındaki sabit dönüşüm aşağıdaki komut ile görüntülenebilir.



Şekil 2: Sisteme ilişkin tf ağacı

#### Command Line

```
$ rosrn tf tf_echo base_footprint base_scan
```

/scan mesajı base\_scan isimli yerel referans çerçevesine göre alınmaktadır.

```
- Translation: [-0.064, 0.000, 0.132]
- Rotation: in Quaternion [0.000, 0.000, 0.000, 1.000]
           in RPY (radian) [0.000, -0.000, 0.000]
           in RPY (degree) [0.000, -0.000, 0.000]
```

Şekil 3: base\_footprint ile base\_scan çerçeveleri arası sabit dönüşüm

#### Command Line

```
$ rostopic echo /scan/header
seq: 0
stamp:
secs: 518
nsecs: 118000000
frame_id: "base_scan"
---
```

### Soru 1

Turtlebot3'ün labirenti tamamlayabilmesini sağlayan bir ROS paketi yazınız. Bunun için aşağıdaki adımları takip ediniz (50p):

- solve\_maze isimli bir ROS paketi oluşturun (roscpp ve gerekli bağımlılıkları komut satırında belirterek)
- solve\_maze paketi içinde solver.cpp isimli bir ROS nod dosyası oluşturun
- solver.cpp nod'unu /scan mesajına abone olacak ve /cmd\_vel mesajını yayınlayacak şekilde düzenleyin
- solver.cpp'de turtlebot3'ün labirenti tamamlayabilmesi için gerekli duvar takibi (wall following) algoritmasını gerçekleştirin
  - Duvara uzaklık parametrisi belirleyin ( $d$ )
  - Paralel bant genişliği parametresini ( $r$ ) belirleyin. (Duvara paralel kalmanın tanımı: robot merkezinin her an için bir ilgili duvardan ( $d - r, d + r$ ) uzaklıktaki bant içinde olmasıdır)
  - Başlangıçta  $d$  uzaklıkta duvar yoksa en yakın duvarı bulun
  - Bulunan duvara göre (birini seçerek) sağdan/soldan paralel duruma gelin
  - L1: Duvara paralel kalarak ilerleyin
  - Eğer  $90^\circ$  köşe varsa karşı duvara paralel olana kadar dönün
  - Eğer  $270^\circ$  köşe varsa açığı duvara paralel olana kadar dönün
  - Eğer  $(0, 0)$  koordinati merkezli  $2m \times 2m$  büyüklüğündeki kareye ulaşılmışsa (\odom isimli mesaj dinlenerek) sonlandırın
  - Tekrarlayın L1
- Paketinizi derleyebilmek için CMakeLists.txt dosyasında gerekli düzenlemeleri yapın, solver.cpp'den derlenecek nod'un ismini my\_solver olarak belirleyin
- Yazmış olduğunuz kodu micromouse\_maze4.launch ile test edin (otomatik olarak seçilen farklı başlangıç noktaları için tekrarlayın)

### Soru 1

Command Line

```
$ rosrunc solve_maze my_solver
```

- Oluşturduğunuz paket içerisindeki README dosyasında paketinizin derleme, çalıştırılma adımları ile elde ettiğiniz sonuçları içeren bir rapor hazırlayın
- Aşağıda verilen adreste boş bir repo oluşturarak kodunuzu yükleyin

### Soru 1

Command Line

```
https://gitlab.com/blm6191_2425b/members/<user>
```

## Soru 2

Turtlebot3'ün labirenti gezinimi (joystick, komut satırından cmd\_vel, rqt\_gui'den vb manuel kontrol ile) sırasında ortam haritasını çıkarmasını sağlayan bir ROS paketi yazınız. Bunun için aşağıdaki adımları takip ediniz (50p):

- solve\_maze isimli bir ROS paketi bu soru için de kullanılacaktır
- solve\_maze paketi içinde mapper.cpp isimli bir ROS nod dosyası oluşturun
- mapper.cpp nod'unu /odom, /scan mesajlarına abone olacak şekilde düzenleyin
- mapper.cpp'de turtlebot3'ün labirenti gezinimi sırasında aldığı her bir lazer sensör ölçümünü odom çerçevesine dönüştürün
  - Alınan ölçümü base\_scan çerçevesinde kartezyen koordinatlara çevirin
  - base\_scan çerçevesinde hesaplanan kartezyen koordinatları base\_footprint çerçevesine göre yeniden hesaplayın
  - base\_footprint çerçevesindeki koordinatları odom çerçevesine göre yeniden hesaplayın
- Elde ettiğiniz metrik koordinatlara, belirlediğiniz çözünürlüğe ve bir pencere genişliğine göre openCV penceresinde, önceki çizimleri koruyarak, bastırın
- Paketinizi derleyebilmek için CMakeLists.txt dosyasında gerekli düzenlemeleri yapın, mapper.cpp'den derlenecek nod'un ismini my\_mapper olarak berileyin
- Yazmış olduğunuz kodu micromouse\_maze4.launch ile test edin

## Soru 2

Command Line

```
$ rosrn solve_maze my_mapper
```

- Oluşturduğunuz paket içerisindeki README dosyasında paketinizin derleme, çalıştırılma adımları ile elde ettiğiniz sonuçları içeren bir rapor hazırlayın
- Aşağıda verilen adreste boş bir repo oluşturarak kodunuzu yükleyin

## Soru 2

Command Line

```
https://gitlab.com/blm6191_2425b/members/<user>
```



**Teslim Şekli ve Son Teslim Tarihi:** Soruya ilişkin cevabınız olan ROS paketini dersin gitlab hesabında [https://gitlab.com/blm6191\\_2425b/members](https://gitlab.com/blm6191_2425b/members) adresi altında kendi adınıza açılmış olan gruba yükleyiniz. Paketin README dosyası paketin yüklenmesi, çalıştırılması ve örnek çıktıların yer aldığı açıklamaları içermelidir.  
Son teslim tarihi **04/11/2024 08:59**