

TurtleBot3

Student: Luka Otović

Mentor: Kristijan Lenac

21.2.2020.

Zadatak:

- osposobiti TurtleBotov-e tipa burger koji se nalaze u APASLabu.

Instaliranje softvera

Radio sam s TurtleBot 3 tipa burger. Morao sam instalirati softver na robote kako bi se njima moglo upravljati, budući da nijedan od tri robota nije bio osposobljen za rad.

Bilo je ponuđeno više različitih operacijskih sustava koji se mogu instalirati na TurtleBot-u: Ubuntu MATE, Raspbian i webOS Robotics Platform.

Ja sam izabrao Raspbian radi jednostavnosti i zato što sam koristio raspberry pi.

Prvo je trebalo skinuti cijeli softver na računalo, postaviti ga na SD karticu koja se zatim ubaci u raspberry pi.

Nakon instalacije trebalo je podesiti nekoliko stvari:

- povećati datotečni sustav da koristi cijelu SD karticu

naredba: "sudo raspi-config (odabrati broj 7 Advanced Option > A1 Expand Filesystem)"

- podesiti vrijeme i datum

naredba: sudo apt-get install ntpdate

sudo ntpdate ntp.ubuntu.com

- podesiti IP robota i servera

Za podešavanje IP robota i servera trebalo je:

1. otvoriti datoteku bashrc

- nano ~/.bashrc

2. u nju zapisati sljedeće dvije linije

- export ROS_MASTER_URI=http://REMOTE_PC_IP:11311
- export ROS_HOSTNAME=RASPBERRY_PI_3_IP

Nakon što smo to dodali na kraju datoteke, morali smo upisati naredbu u terminal:

- source ~/.bashrc kako bi se te postavke primijenile.

Na taj način pripremili smo raspberry pi za rad.

Pripremanje računala

Također, bilo je potrebno pripremiti i računalo s kojim ćemo se spajati na TurtleBot i upravljati njime.

Preporučeno je da se na računalo instalira Ubuntu 16.04 jer je na njemu već sve isprobano da radi.

Nakon što instaliramo Ubuntu i nadogradimo ga, treba instalirati ROS.

To se može napraviti koristeći naredbu:

```
- wget https://raw.githubusercontent.com/ROBOTIS-GIT/  
robotis_tools/master/install_ros_kinetic.sh && chmod 755  
./install_ros_kinetic.sh && bash ./install_ros_kinetic.sh
```

Još je trebalo instalirati zavisne ROS pakete, a to se radi preko naredba:

```
- sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-  
teleop-twist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd-launch ros-  
kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-  
rosserial-python ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-  
kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server ros-kinetic-  
move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-compressed-image-  
transport ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-kinetic-  
navigation ros-kinetic-interactive-markers  
- cd ~/catkin_ws/src/  
- git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git  
- git clone https://github.com/ROBOTIS-GIT/turtlebot3.git  
- cd ~/catkin_ws && catkin_make
```

Isto kao i kod TurtleBot-a, trebalo je podesiti IP servera i IP računala.

Radi se na sličan način kao i kod TurtleBot-a.:

1. otvorimo bashrc

- nano ~/.bashrc

2. dodamo sljedeće dvije linije na kraju datoteke

- export ROS_MASTER_URI=http://192.168.0.100:11311

- export ROS_HOSTNAME=192.168.0.100

Na kraju potrebno je još u terminalu upisati naredbu: source ~/.bashrc.

S ovime smo pripremili naše računalo za rad.

Osposobljavanje OpenCR-a

Kako bismo mogli upravljati motorima i senzorima na TurtleBot-u, potrebno je podesiti pločicu OpenCR koja upravlja sa svim senzorima i motorima.

Preporučeno je da se koristi "Shell Script".

Za podešavanje rada pločice OpenCR-a postoje različiti načini, ovisno o tome koji model TurtleBot-a želimo koristiti.

Sljedeće naredbe su za TurtleBot modela burger:

```
- export OPENCR_PORT=/dev/ttyACM0
- export OPEN_MODEL=burger
- rm -rf ./opencr_update.tar.bz2
- wget https://github.com/ROBOTIS-GIT/OpenCR-Binaries/
raw/master/turtlebot3/ROS1/latest/opencr_update.tar.bz2 && tar -xvf
opencr_update.tar.bz2 && cd ./opencr_update && ./update.sh
$OPENCNCR_PORT $OPENCNCR_MODEL.opencr && cd ..
```

Ovaj postupak se isto može provesti preko Arduina IDE.

Nakon što se završi ovaj postupak, može se preko OpenCR-a izvršiti osnovne operacije, poput pomicanja robota unaprijed ako se pritisne PUSH SW 1 ili rotiranje robota ako se pritisne PUSH SW 2.

GPIO

1	3.3V	6	D53	11	D58	16	D63
2	GND	7	D54	12	D59	17	D64
3	D50	8	D55	13	D60	18	D65
4	D51	9	D56	14	D61	19	D66
5	D52	10	D57	15	D62	20	D67

1	3.3V	6	JTDO_SWO
2	JTMS_SWDIO	7	GND
3	GND	8	JTDI
4	JTCK_SWCLK	9	GND
5	GND	10	MCU_NRESET

JTAG

GPIO

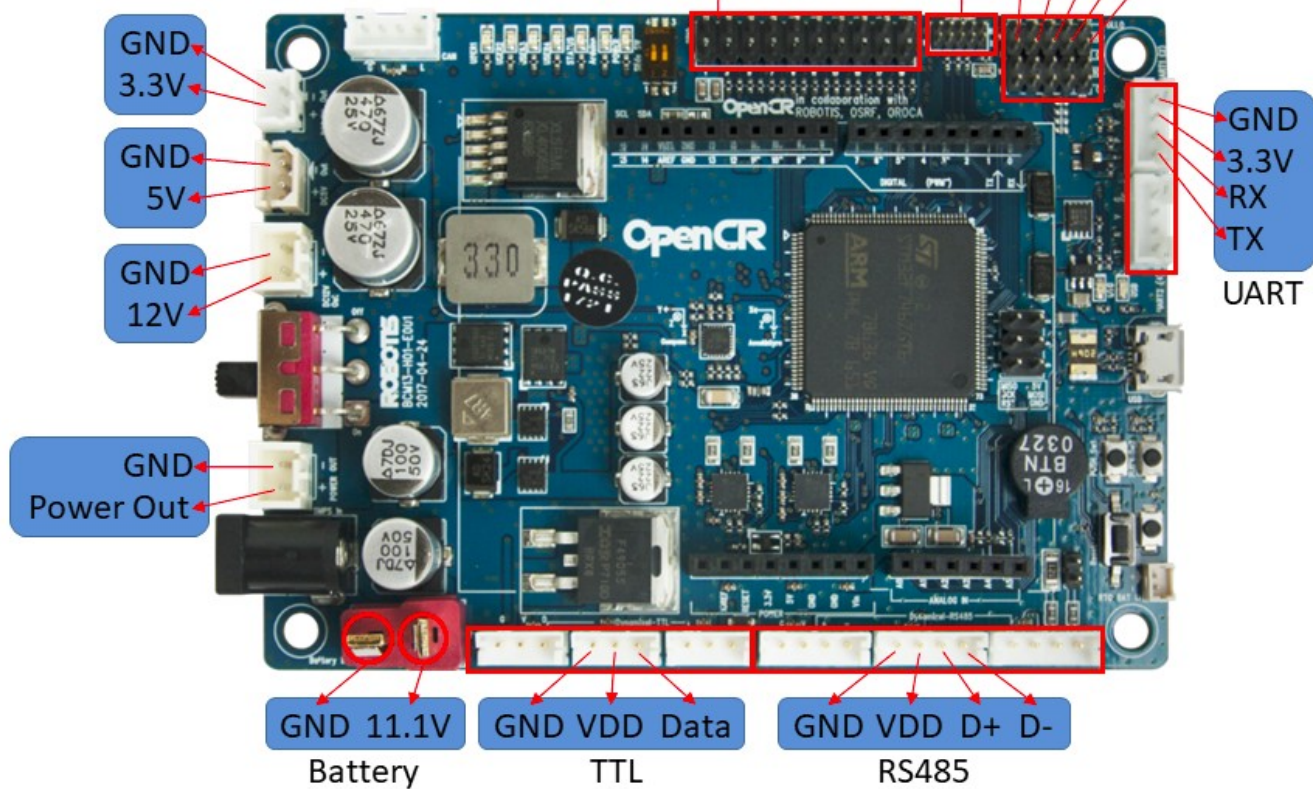
2	4	6	8	10	12	14	16	18	20
1	3	5	7	9	11	13	15	17	19

JTAG

9	7	5	3	1
10	8	6	4	2

OLLO

OUT2	GND	ADC	3.3V	OUT1
------	-----	-----	------	------



Slika 1: OpenCR

Funkcionalnosti TurtleBot-a

Prva funkcionalnost koju robot ima je očitavanje podataka s laserskog skenera. Ti podaci se mogu kroz RViz vizualizirati na ekranu računala.

Podaci koje senzor šalje je koliko je udaljena prepreka od nas i onda se u programu preko točkica prikazuju prepreke na različitim udaljenostima.

Kako bismo to testirali, prvo trebamo pokrenuti server na našem računalu. To se radi naredbom: `roscore`.

Nakon toga, moramo se povezati na TurtleBot preko naredbe `ssh` ili spojiti tipkovnicu, miša i monitor na robota.

Zatim treba upisati sljedeću naredbu:

- `roslaunch turtlebot3_bringup turtlebot3_robot.launch`.

Ova naredba će spojiti TurtleBot na naš server i pokrenuti robot (upaliti senzore, motore i sve ostale stvari koje su spojene na njega).

Ovaj postupak se još naziva i Bringup.

Kako bismo mogli pročitati podatke koje nam robot šalje, npr. od laserskog senzora trebamo pokrenuti RViz program.

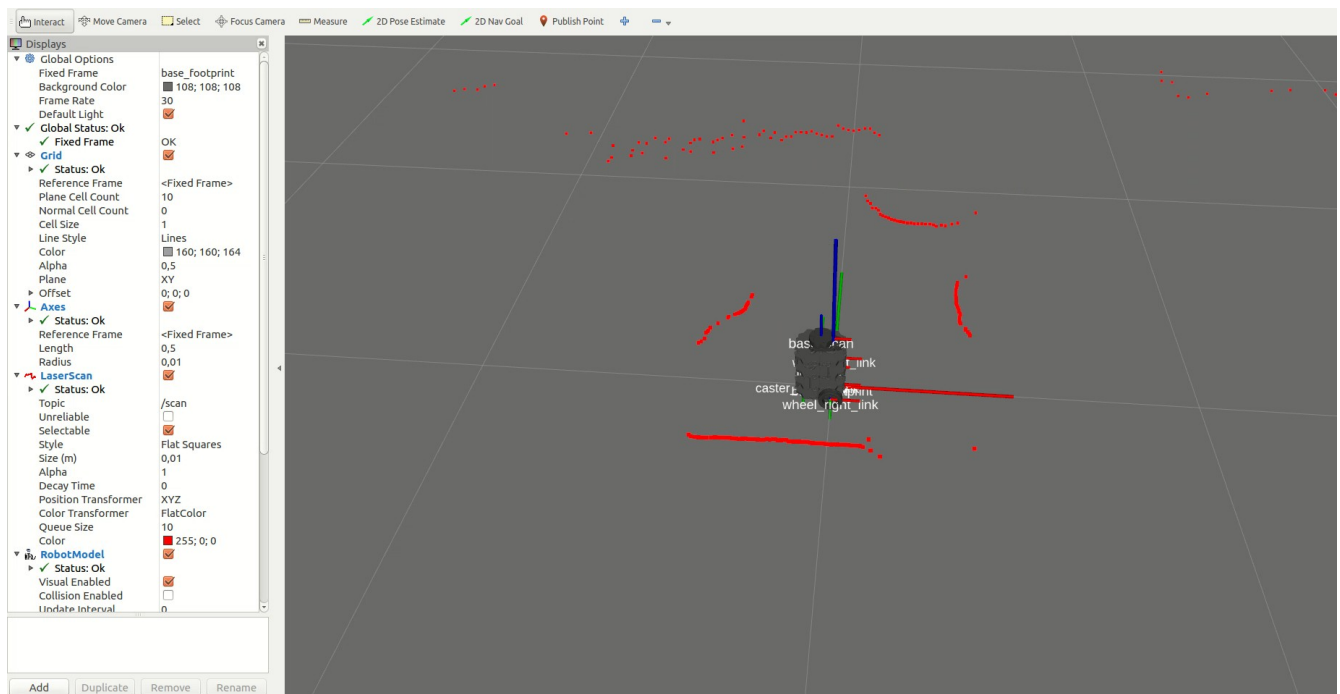
Da bismo to napravili, potrebno je upisati sljedeće naredbe u terminal:

- `export TURTLEBOT3_MODEL=burger`
- `roslaunch turtlebot3_bringup turtlebot3_remote.launch`

S ovim naredbama kažemo koji model TurtleBot-a koristimo i da želimo komunicirati s tim robotom (uspostava konekcije između računala i robota preko servera).

Sada još moramo preko drugog terminala upaliti RViz program:

- `roslaunch rviz rviz -d `rospack find turtlebot3_description`/rviz/model.rviz`



Slika 2: RViz

Druga funkcionalnost je upravljanje samim robotom, poput pomicanja naprijed - natrag i okretanja koja se mogu izvršiti na različite načine.

Ta funkcionalnost se može realizirati preko:

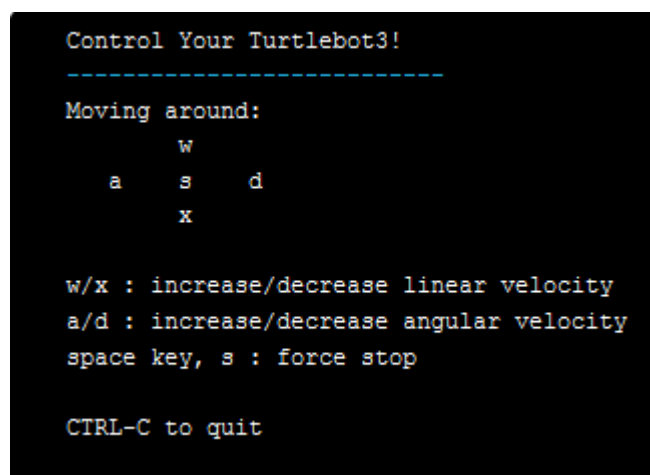
- terminala (korištenjem tipkovnice)
- RViz programa
- PS3 joysticka i na mnoge druge načine.

Ovo se još naziva "Teleoperation" ili "Basic Operation".

Prije nego što se započne raditi bilo koji od ovih funkcionalnosti, treba se izvršiti Bringup koji je opisan na početku ovog poglavlja.

Za upravljanje robotom preko terminala potrebno je upisati sljedeće dvije naredbe:

- `export TURTLEBOT3_MODEL=burger`
- `roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch`

A screenshot of a terminal window with a black background and light blue text. The text displays the 'Control Your Turtlebot3!' header, followed by a dashed line. Below this, it says 'Moving around:' and lists the keys 'w', 'a', 's', 'd', and 'x' in a staggered layout. Further down, it provides a legend: 'w/x : increase/decrease linear velocity', 'a/d : increase/decrease angular velocity', and 'space key, s : force stop'. At the bottom, it says 'CTRL-C to quit'.

Slika 3: Upravljanje TurtleBot-a preko terminala

Da bi se koristila ostala sredstva za upravljanje TurtleBot-om poput LEAP Motiona ili joysticka treba instalirati dodatne pakete prije nego što ih se koristi.

Također, robotom se može upravljati i preko RViz programa upisivanjem ovih naredbi:

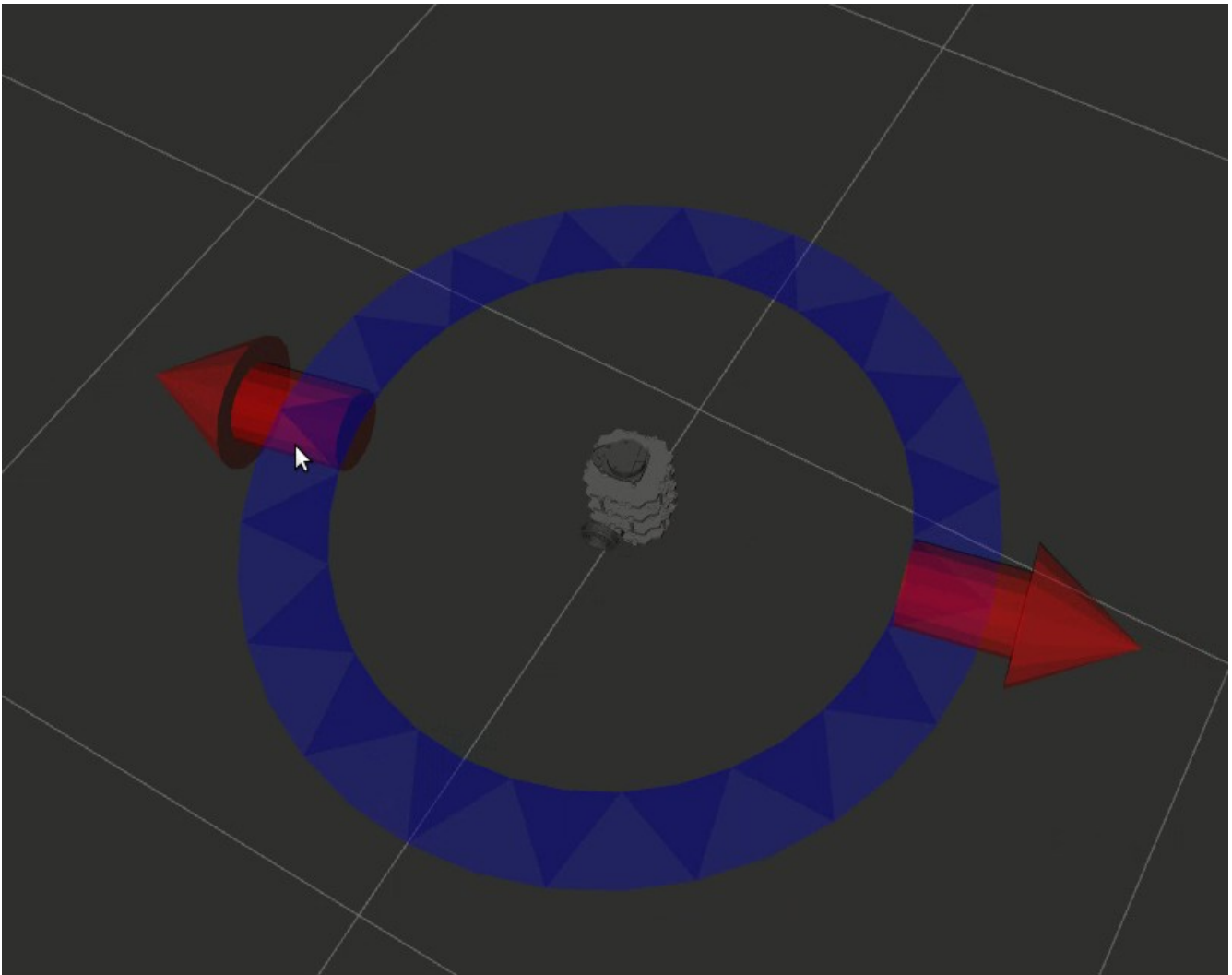
- export TURTLEBOT3_MODEL=burger
- roslaunch turtlebot3_bringup turtlebot3_remote.launch

Zatim u novom terminalu upisati naredbu:

- roslaunch turtlebot3_example interactive_markers.launch

Konačno, možemo pokrenuti rviz program naredbom:

- rosrn rviz rviz -d `rospack find turtlebot3_example`/rviz/
turtlebot3_interactive.rviz



Slika 4: Interaktivni RViz

Sljedeća mogućnost TurtleBot-a je pronalazak prepreke. Robot će se micati u jednom smjeru dok ne dođe do prepreke i tada će stati. Koristeći laserski senzor on može vidjeti prepreku i odrediti na kojoj udaljenosti se ona nalazi.

Robot nam šalje te informacije i ispisiuje ih u terminal. Kada se TurtleBot dovoljno približi prepreci, on će se zaustaviti i na ekranu će nam se početi ispisivati "Stop!".

To se pokreće sljedećom naredbom:

- `roslaunch turtlebot3_example turtlebot3_obstacle.launch`

Može mu se također zadati x, y koordinata u prostoru. On se tada pomakne na tu točku. Ako mu npr. pošaljemo (0.2, 0.5, 90) robot će se pomaknuti na točku koja je po x osi udaljena za 0.2 metara, po y osi za 0.3 metara i zarotirat će se za 90 stupnjeva.

To se pokreće naredbom:

- `roslaunch turtlebot3_example turtlebot3_pointop_key.launch`

Isto mu se može reći da se kreće po unaprijed definiranom putu (u obliku pravokutnika, trokuta ili kružnice).

To se pokreće naredbama:

- `roslaunch turtlebot3_example turtlebot3_server`

- `roslaunch turtlebot3_example turtlebot3_client.launch`

Naime, preko client-a mu šaljemo način rada. Za pravokutno i trokutasto pomicanje odredimo mu duljinu stranice i broj ponavljanja, a za kružno pomicanje odredimo mu radijus kružnice i također broj ponavljanja.

SLAM(Simultaneous Localization and Mapping)

SLAM je algoritam koji se može primijeniti na bilo kojoj vrsti robota. Kao što samo ime kaže, SLAM služi za crtanje mape i snalaženje robota na toj mapi. Preko laserskog senzora robot dobiva informacije gdje se nalaze prepreke i na kojoj udaljenosti se nalaze. TurtleBot pamti svoju poziciju i mjesto gdje se nalaze prepreke i zatim iz tih informacija se iscrtava mapa na ekranu i na samoj mapi se prikazuje robot. Ideja je da pomicemo TurtleBot-a po prostoriji (možemo koristiti terminal za upravljanje robotom) te da nam on iscrta preciznu mapu.

Najprije moramo izvršiti Bringup i zatim pokrenuti SLAM program na računalu.

To se pokreće naredbama:

- export TURTLEBOT3_MODEL=burger
- roslaunch turtlebot3_slam turtlebot3_slam.launch
slam_methods:=gmapping

Kada pokrenemo SLAM, moramo pokrenuti bilo koji program za pomicanje TurtleBot-a.

Ovdje je pokazano pomicanje robota kroz terminal.

Naredbe:

- export TURTLEBOT3_MODEL=burger
- roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch

Kada smo iscrtali mapu s kojom smo zadovoljni, treba ju spremiti na računalo.

Naredba:

- `roslaunch map_server map_server -f ~/map` (putanja di će se spremiti slika i njezino ime)



Slika 5: Mapa

Navigacija

Do sada smo vidjeli jako mnogo različitih načina za upravljanje/navigiranje TurtleBot-a. Prošle navigacije su se izvršavale iz terminala i bilo je jako teško za vidjeti gdje se u prostoru nalazimo i koliko smo udaljeni od prepreke.

Ovoj metodi dajemo mapu koju smo iscrtali u prošlom koraku i damo mu početnu poziciju i krajnju lokaciju. TurtleBot će pronaći najbrži put od početne točke do cilja, a pritom će zaobići sve prepreke.

Prvo pokrenemo Bringup, a zatim program za navigaciju.

Naredbe:

- `export TURTLEBOT3_MODEL=burger`
- `roslaunch turtlebot3_navigation turtlebot3_navigation.launch`
`map_file:=$HOME/map.yml`

Pomoću ovih naredbi definirali smo koji ćemo navigacijski program i mapu koristiti.

Na početku moramo zadati početnu lokaciju TurtleBot-a. Preko izbornika mu kažemo približno točnu poziciju i smjer gledanja, a zatim preko terminala ga pomičemo po mapi i prostorijski kako bi se odredila precizna lokacija robota.

Najjednostavniji način za to je iskoristiti terminal za upravljanje TurtleBot-a.

Jednom kada je određena precizna početna lokacija, možemo mu dati krajnju točku/cilj. TurtleBot će nam pokazati na mapi put do nje i početak će se micati po tom putu. Kada dođe na cilj, stat će i zatim mu možemo zadati novu lokaciju na koju želimo da se robot pomakne.

Isto se može izvesti simulacija u kojoj će se virtualni TurtleBot pomicati od početne točke do cilja.

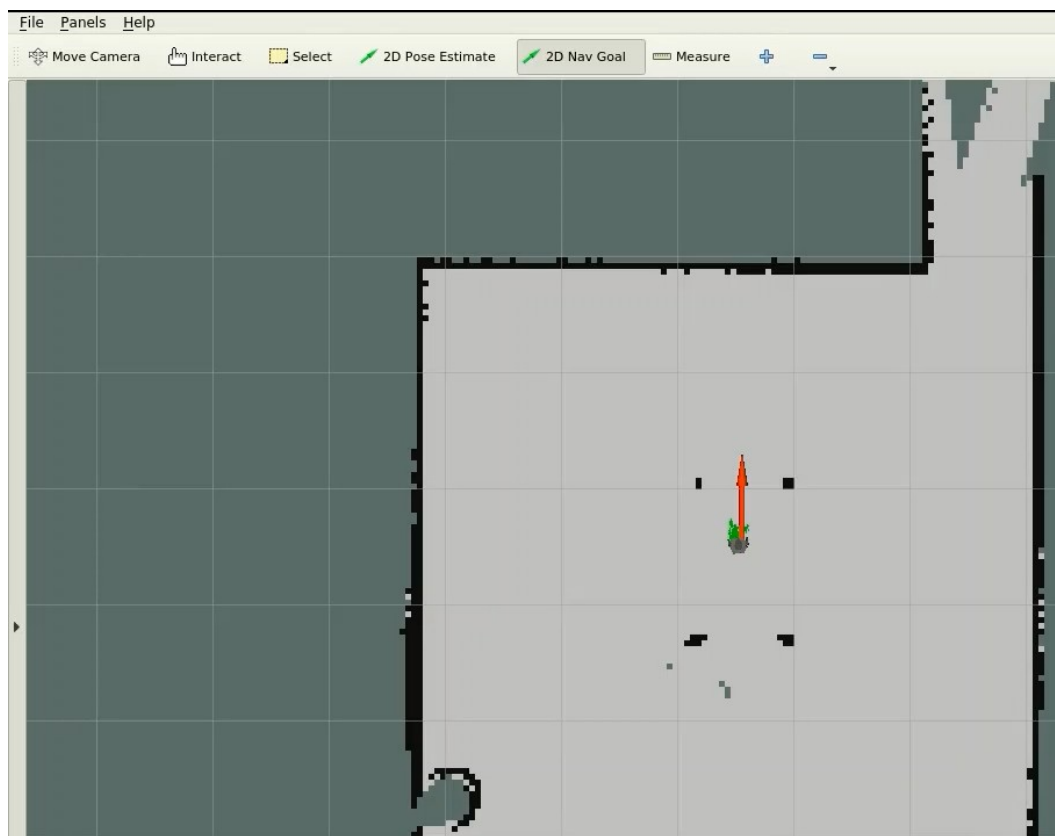
Prvo treba pokrenuti server preko naredbe: roscore.

Potom treba pokrenuti virtualno okruženje i dati mu mapu po kojoj će se kretati.

Naredbe:

- roslaunch turtlebot3_gazebo turtlebot3_house.launch (pokreće virtualno okruženje)
- roslaunch turtlebot3_navigation turtlebot3_navigation.launch
map_file:=/home/luka/novaMapa.yaml

Nakon što mu odredimo početnu poziciju, možemo mu zadati točku u prostoru gdje želimo da se on pomakne.



Slika 6: Postavljanje početne točke u gazebu



Slika 7: Pomikanje TurtleBot-a

Problemi

Tijekom izvođenja projekta desilo se nekoliko problema. Neki od njih su se uspješno riješili, dok se neki od njih nisu mogli u cijelosti riješiti zbog manjka opreme.

Uvijek prije rada s TurtleBot-om napunite njegovu bateriju, jer ako padne ispod određenog naponskog nivoa, određeni dijelovi robota će prestati raditi. Tako npr. nećete ga moći pomicati, ali će se i dalje moći očitavati informacije sa senzora.

Prvi problem je bio kod konfiguriranja IP adresa. Naime, ako ste spojeni na hotspot, vaš laptop i TurtleBot ne moraju imati svaki put istu adresu pa to može izazvati sitne probleme.

Drugi problem je bio što jedan od TurtleBot-ova ima dodatnu treću nogu koja nema zglobove pa se kotač nije mogao rotirati. To je izazvalo probleme u navigacijskom programu.

Treći problem je bila komunikacija između OpenCR-a i motora. Ovaj problem još nije popravljen radi manjka opreme. Naime, kupnjom novih dynamixel motora, OpenCR nije mogao komunicirati s njima.

Problem je bio u tome, kada se kupe novi motori, njihov ID na mreži je uvijek 1. U trenutku kada se oba spoje na isti OpenCR izazivaju smetnje u komunikaciji. Da bi se to riješilo, potrebno je kupiti DYNAMIXEL U2D2 pomoću kojeg se može promijeniti ID motora.