

---

# Industria Elektronikaren eta Automatikaren Ingeniaritzako Gradua

## **3. ERANSKINA: ROS 2ren OINARRIZKO GIDA III**

**ACKERMANN DIREKZIOA DUEN LAU  
GURPILEKO ROBOT MUGIKOR BATEN  
NABIGAZIO AUTONOMOA ROS2N**

---

**Ikaslearen izen eta abizenak:** Iñaki Azkarragaurizar Jauregi

---

**Irakaslearen izen eta abizenak:** Mikel Larrea Sukia

---

**Data:** Donostia, 2023ko uztailaren 21a



Universidad  
del País Vasco  
Euskal Herriko  
Unibertsitatea

GIPUZKOAKO  
INGENIARITZA  
ESKOLA  
ESCUELA  
DE INGENIERÍA  
DE GIPUZKOA

# Aurkibidea:

<b>Argazkien Zerrenda</b>	<b>5</b>
<b>Komandoen Zerrenda</b>	<b>6</b>
<b>Programazio-Kodeen Zerrenda</b>	<b>7</b>
<b>Akronimoen Zerrenda</b>	<b>8</b>
<b>1. SARRERA</b>	<b>9</b>
1.1 Aurrebaldintzak	9
1.2 Robota	10
<b>2. LAN-ESPAZIOA ETA PAKETEA SORTU</b>	<b>11</b>
<b>3. ROBOT MUGIKORRA SORTU</b>	<b>12</b>
3.1 URDF	12
3.2 RViz	14
3.3 Launch file:	14
3.4 Dependentziak (package.xml)	16
3.5 CMakeLists.txt	16
3.6 Paketea konpilatu	16
3.6 Jaurti robota RViz-en	18
3.7 rqt_graph	20
<b>4. GAZEBO</b>	<b>21</b>
4.1 Ackermann Plugina	21
4.2 URDF	23
4.3 Sortu mundua	23
4.4 Launch file	24
4.5 Martxan jarri	26
4.6 Nodoen zerrenda:	27
4.7 Nodoen informazioa	28
4.8 Gaien zerrenda	29
4.9 Gaien informazioa	30
4.10 Robota kontrolatu	31
<b>5. NABEGAZIOA</b>	<b>35</b>
5.1 Lokalizazioa	35
5.2 Mapaketa	40
<b>6. BIBLIOGRAFIA</b>	<b>47</b>
<b>7. Eranskina: Programazio-kodeak</b>	<b>48</b>

# Argazkien Zerrenda

1. Irudia: Marti_bot robota.	10
2. Irudia: proiektuaren hasierako egitura.	11
3. Irudia: proiektuaren egitura orain arte.	17
4. Irudia: lau gurpildun robota RViz-en.	18
5. Irudia: robotaren koordenatu-markoaren zuhaitza.	19
6. Irudia: nodoen komunikazioa grafikoki.	20
7. Irudia: Gazebo simulazio-ingurunea.	21
8. Irudia: marti_bot-a Gazebo simulazio-ingurunean.	27
9. Irudia: marti_bot-aren posizio berria.	32
10. Irudia: nodoen konexioen irudikapena.	34
11. Irudia: Marti_bot-a LIDAR sentsorearekin.	37
12. Irudia: LaserScan aktibatu RVizen.	39
13. Irudia: oztopoen detekzioa RVizen, lidarrarari esker.	39
14. Irudia: marti_bot-a SLAMarekin.	44

# Komandoen Zerrenda

1. Komandoa: karpeta bat sortzeko komandoa.	10
2. Komandoa: src karpetara joateko komandoa	10
3. Komandoa: C++-eko paketea sortzeko komandoa.	10
4. Komandoa: fitxategi bat sortzeko komandoa.	11
5. Komandoa: konpilatzeko komandoa.	15
6. Komandoa: lan-espazioa kargatzeko komandoa.	16
7. Komandoa: “launch” fitxategi bat abiarazteko komandoa.	17
8. Komandoa: exekutagarri bat abiarazteko komandoa.	18
9. Komandoa: nodoen konexioak era grafiko batean ikusteko komandoa..	19
10. Komandoa: gazebo abiarazteko komandoa.	20
11. Komandoa: GitHub-eko gordailu bat kopiatzeko komandoa.	21
12. Komandoa: aktibo dauden nodoen zerrenda bat ikusteko komandoa.	26
13. Komandoa: nodo baten informazioa ikusteko komandoa.	27
14. Komandoa: aktibo dauden gaien zerrenda bat ikusteko komandoa.	28
15. Komandoa: gai baten informazioa ikusteko komandoa.	29

# Programazio-Kodeen Zerrenda

1. Programazio-kodea: deskribapena.xacro	49
2. Programazio-kodea: marti_bot.urdf.xacro	50
3. Programazio-kodea: urdf_config.rviz	51
4. Programazio-kodea: rviz_robota.launch.py	52
5. Programazio-kodea: package.xml	53
6. Programazio-kodea: CMakeLists.txt	53
7. Programazio-kodea: gazebo_control.xacro	54
8. Programazio-kodea: etxea.world	70
9. Programazio-kodea: spawn_car.launch.py	71
10. Programazio-kodea: marti_gazebo.launch.py	71
11. Programazio-kodea: lidar.xacro	73
12. Programazio-kodea: CMakeLists.txt	73
13. Programazio-kodea: nav2_params.yaml	79
14. Programazio-kodea: nav2_default_view.rviz	88
15. Programazio-kodea: rviz_launch.py	90
16. Programazio-kodea: nabegazioa.launch.py	92

# Akronimoen Zerrenda

**GIE** Gipuzkoako Ingeniaritza Eskola

**GrAL** Gradu Amaierako Lana

**ROS** Robot Operating System

**XML** eXtensible Markup Language

**EOL** End of Life

**URDF** Unified Robot Description Format

**SDF** Simulation Description Format

**SLAM** Simultaneous Localization And Mapping

**AMCL** Adaptive Monte Carlo Localization

**LiDAR** Light Detection and Ranging

# 1. SARRERA

## 1.1 Aurrebaldintzak

- Robot Operating System 2ren Oinarrizko Gida I, osorik osatua.
- Robot Operating System 2ren Oinarrizko Gida II, osorik osatua.
- Gida honek The Construct [1] plataforma erabiltzen du, robotentzako simulazio-ingurune bat eskaintzen baitu. Horri esker, gida honetan erabiliko diren pakete asko deskargatuta egongo dira. Bertan, soilik, eskuragarri dauden paketeen zerrenda egunerautu eta sisteman instalatutako paketeak egunerautu beharko dira. Horretarako:

Terminal bat ireki:

```
sudo apt update
```

Komando hori exekutatzean, sistemak instalatutako paketeetarako egunerautze erabilgarriak bilatuko ditu, eta tokiko biltegien informazioa egunerautuko du.

Ondoren, terminal berdinean:

```
sudo apt-get upgrade
```

Sisteman instalatutako paketeak egunerautzeko erabiltzen da komando hau. Horri esker, plataforma honetan nabegazioko liburutegiak, beste zenbaiten artean erabiltza ahalbidetuko du.

- ROS 2 Humble Hawksbill [2] bertsioa erabiliko da. The Construct plataforma erabiltzea gomendagarria da, baina horrela ez bada, ROS 2n [2] eta Navigation 2ren [3] webgune ofizialean behar den guzta deskarga daiteke.

The Construct plataforma buruzko informazio gehiago, bere web orrian aurki daiteke [1].

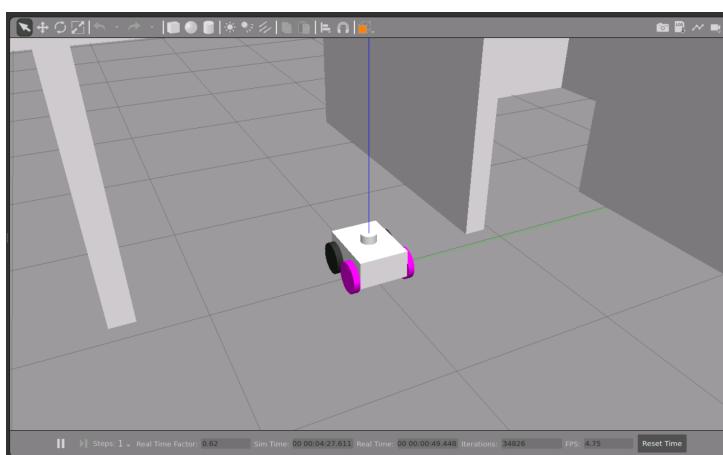
## 1.2 Robota

Gida honetako robotak lau gurpileko trakzio-sistema bat du, eta Ackermann direkzio-sistema bat erabiltzen du, non aurreko bi gurpilek direkzionalak diren. Gida honen helburu nagusia robotaren nabigazio autonomoa lortzea da, ROS2k eskainitako nabigazio-liburutegi baten bitartez (Nav2). Hau guztia, hurrengo urratsetan egingo da:

1. Robota zerotik sortu Xacro fitxategien bidez eta behar diren konfigurazio guztiak konfiguratu, RVizen bistaratzeko.
2. Robota Gazebo simulazio-eremuan abiatu, honen mugimendua kontrolatzen duen nodoarekin. Robotak erabiltzen duen direkzio sistema kontrolatzen duen plugin gehitu eta Gazebo simulazio-eremuan abiatu.
3. Robotari nabegazio-liburutegia gehitu, horretarako, beharrezko sentsore, pakete eta parametroak konfiguratu.

Sortuko den robota, Marti\_bot izena izango du eta ondorengo osagai fisikoak izango ditu:

1. Txasisa (zuria).
2. Lau trakzio-gurpil.
  - a. Aurrekoak direkzionalak (moreak)
  - b. Atzekoak finkoak (beltzak)
3. LIDAR sentsorea (zuria).



1. irudia: Marti\_bot robota.

## 2. LAN-ESPAZIOA ETA PAKETEA SORTU

Robotarekin lanean hasi baino lehen, lehenik eta behin lan-espazio bat sortuko da, hau da, proiektu honen karpeta izango dena. Horretarako terminal bat ireki:

```
mkdir -p marti_ws/src
```

**1. Komandoa:** karpeta bat sortzeko komandoa.

Ondoren, Marti\_bot izeneko C++ motako paketea eraikiko da aurretik sortutako lan-espazioaren barruko src karpetan, direkzio horretara joateko, terminal berdinean.

```
cd marti_ws/src/
```

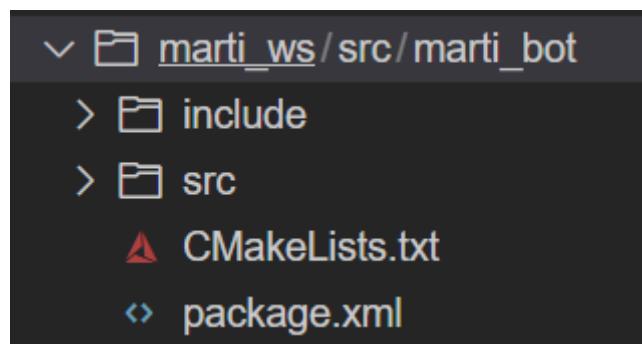
**2. Komandoa:** src karpetara joateko komandoa

Behin karpeta horren barruan kokatuta, C++eko paketea eraikitzeko komandoa, hurrengoa da:

```
ros2 pkg create --build-type ament_cmake marti_bot
```

**3. Komandoa:** C++-eko paketea sortzeko komandoa.

Code Editor-ean sartuta, 2. irudian ikus daitekeen egitura berdina eduki beharko litzateke.



**2. Irudia:** proiektuaren hasierako egitura.

## 3. ROBOT MUGIKORRA SORTU

### 3.1 URDF

Robot mugikor simulatu bat eraikitzeko, ROS 2k, Universal Robot Description Format (URDF) erabiltzen du. Bertan, robotaren propietate fisiko guztiak deskribatuko dira. URDF fitxategiei buruzko informazio zabalagoa eta zehatzagoa izateko, dokumentazio ofizialean aurki daiteke [4].

Robotaren kode garbiago bat eskeintzeko, ROSEk Xacro izeneko pakete bat dauka. Pakete honek robot ereduak edo osagai robotikoak modu modularragoan deskribatzeko aukera ematen du. Fitxategi honi buruzko informazio gehiago izateko, dokumentazio ofizialean aurki daiteke [5].

Lehenik eta behin, “deskribapena” izeneko karpeta bat sortuko da, robota deskribatzen duten fitxategi guztiak biltzeko. Horretarako, terminal bat ireki:

```
cd ~/marti_ws/src/marti_bot/  
mkdir deskribapena
```

Oraintxe sortutako karpetaren barruan, lehenengo fitxategia sortuko da. Bertan, Marti\_bot robotaren egitura fisikoak deskribatuko dira, XML formatua erabiliz. Terminal batean:

```
cd deskribapena/  
touch deskribapena.xacro
```

**4. Komandoa:** fitxategi bat sortzeko komandoa.

Egin klik [deskribapena.xacro](#) atalean kode osoa ikusteko.

**Kodearen azalpena:**

1. Balio konstanteak: Hasieran zenbait propietate definitzen dira, hala nola robotaren onarriarekin eta gurpilekin lotutako dimensio era parametroak.

2. Propietateak: Kutxa eta zilindro baten propietate inertzialak zehazten dira, aurrerago robotaren txasia eta gurpilak sortzeko erabiliko dira.
3. Base link: Lotura hau, robotaren oinarria izango da. Kutxa baten itxurakoa da eta bere propietate inertzialak ditu.
4. Base footprint: Lotura honek, robotak lurrean duen proiekzioa irudikatzen du. Ez du inolako geometriarik, erreferentzia puntu gisa erabiltzen da beste lotura batzuekiko.
5. Base joint: “Fixed” motatako artikulazio honek, base\_footprint eta base\_link loturak konektatzen ditu. Hierarkia mota baten, base\_footprint bestearren gainetik jarritz.
6. Makro Atzeko gurpilak: Makro honen bitartez atzoko ezker eta eskuin gurpilen loturak eta artikulazioak definitzen dira. Gurpil bakoitza, “continuous” artikulazio mota du, etengabe biratu ahal izateko. Eta base\_link-erako lotura dute.
7. Aurreko Direkzioa: “urrekoezk\_gurp\_steer” eta “urrekoesk\_gurp\_steer” izeneko, lotura eta artikulazio bat definitzen dira. Honi esker, aurreko gurpilak angelu batean biratu ahalko dute. Direkzio honek, base\_link-erako lotura dute.
8. Makro Aurreko gurpilak (direkzionalak): Seigarren puntuaren azaldu den antzekoa da, baina kasu honetan, 7. puntuoko “urrekoezk\_gurp\_steer” eta “urrekoesk\_gurp\_steer” izeneko direkzioekin lotura dute, “base-link-arekin” izan beharrean. Lotura honi esker, aurreko gurpilak angelu batean biratu ahalko dute.

Amaitzeko, Marti\_bot izeneko fitxategi gorena sortu da, eta hortik Xacro fitxategi guztiak abiaraziko dira. Hau, kodea garbiagoa eta egituratuagoa izan dadin egiten da. Terminal bat ireki:

```
cd deskribapena/  
touch marti_bot.urdf.xacro
```

Egin klik [marti\\_bot.urdf.xacro](#) atalean kode osoa ikusteko.

## 3.2 RViz

RViz oso erabilgarria da datu sentsorialak, robotaren transformatuak eta abar ikusteko. Beraz, oso baliotsua izango da robotak sortutako datuak bistaratzea eta aztertzeko.

Terminal batean, RViz izeneko karpeta bat sortu eta bere barruan urdf\_config.rviz izeneko fitxategi bat.

```
cd ~/marti_ws/src/marti_bot/  
mkdir rviz  
cd rviz/  
touch urdf_config.rviz
```

Egin klik [urdf\\_config.rviz](#) atalean kode osoa ikusteko.

Fitxategi hau, RViz bistaratze-konfigurazioa deskribatzeko erabiltzen da. Honi esker panelak, bistaratzeak, tresnak eta konfigurazio globalen itxurak konfiguratu daiteke.

RVizi eta bere funtzionaltasunari buruzko informazio gehiago lortzeko, kontsultatu dokumentazio ofizialean [6].

## 3.3 Launch file:

Jauritzeko fitxategia edo ezagunago “Launch file”, sortutako robota eta beharrezko nodoak kargatzeko erabiliko da. Terminal bat ireki, eta karpeta berri bat sortuko da eta bere barruan dagokion jaurtitze fitxategia:

```
cd ~/marti_ws/src/marti_bot/  
mkdir launch  
cd launch/  
touch rviz_roboata.launch.py
```

Egin klik [rviz\\_roboata.launch.py](#) atalean kode osoa ikusteko.

### Kodearen azalpena:

1. Paketeen importaketa: Fitxategi hasieran, beharrezko moduluak eta paketeak importatzenten dira, “from” eta “import” izeneko komandoen bitartez.
2. Karpeten direkzioa: Robotak behar dituen fitxategi eta karpeten direkzioa adierazten da, lehenengo zatian zehaztutako "os" eta "FindPackageShare" paketeen bidez.
3. Aldagaiak ezarri: “LaunchConfiguration” paketearen bitartez, aldagaiak ezarri dira. Honek, robotaren simulazioa martxan jartzea ahalbidetuko du.
4. start\_joint\_state\_publisher\_cmd: Nodo honek, robotaren URDF fitxategian finkoak ez diren artikulazioen egoera-balioak argitaratzeko ardura du.
5. start\_joint\_state\_publisher\_gui\_node: Aurrekoaren antzekoa da, baina egoera-balioak manipulatzeko interfaze bat eskaintzen du.
6. start\_robot\_state\_publisher\_cmd: Nodo honi esker, robotaren artikulazio egoeretara harpidetzen da eta lotura bakoitzaren orientazioa argitaratzen du.
7. start\_rviz\_cmd: Nodo honi esker, RViz aplikazioa abiarazten du.
8. return: Azkenik, aurretik deklaratutako ekintza guztiak “LaunchDescription” objetuari gehitzen zaizkio, eta emaitza gisa itzultzen da.

Fitxategi hauei buruzko informazio gehiago ROS 2ren dokumentazio ofizialean aurki daiteke [2].

### 3.4 Dependentziak (package.xml)

“Package.xml” fitxategiak, paketeari buruzko informazioa ematen du, hala nola: izena, proiektuaren deskribapena, egilea, dependentziak eta erlazionatutako beste konfigurazio batzuk. Proiektuaren mende egongo diren pakete batzuk gehitzea beharrezko da, horretarako fitxategi hori ireki eta gehitu hurrengoa:

```
<exec_depend>joint_state_publisher</exec_depend>
<exec_depend>robot_state_publisher</exec_depend>
<exec_depend>rviz2</exec_depend>
<exec_depend>xacro</exec_depend>
```

Egin klik [package.xml](#) atalean kode osoa ikusteko.

### 3.5 CMakeLists.txt

Paketea konpilatu ahal izateko, beharrezko konpilazio-konfigurazioa ezartzeaz eta iturburuko dependentzia eta artxiboei buruzko informazioa emateaz arduratzen da. Orain arte sortutako karpeta eta fitxategiak sisteman instalatu ahal izateko, hurrengo kode zatia gehitu CMakeLists.txt fitxategian:

```
install(
  DIRECTORY src launch rviz deskribapena
  DESTINATION share/${PROJECT_NAME}
)
```

Hau, if(BUILD\_TESTING) kode-lerroaren aurretik jartzea gomendatzen da. Egin klik [CMakeLists.txt](#) atalean kode osoa ikusteko.

### 3.6 Paketea konpilatu

Puntu honetan, sortutako paketea arazorik gabe konpilatu ahal da. Akzio hau, lan-espazioaren barruan egin behar da beti, hau oso garrantzitsu da. Horretarako, terminal bat ireki:

```
cd ~/marti_ws/
colcon build
```

**5. Komandoa:** konpilatzeko komandoa.

**Gogoratu:** Lan-espazioan aldaketa bat egiten den bakoitzean, egin beharreko pausu bat da eta beti lan-espazioaren barruan exekutatu behar da akzio hau.

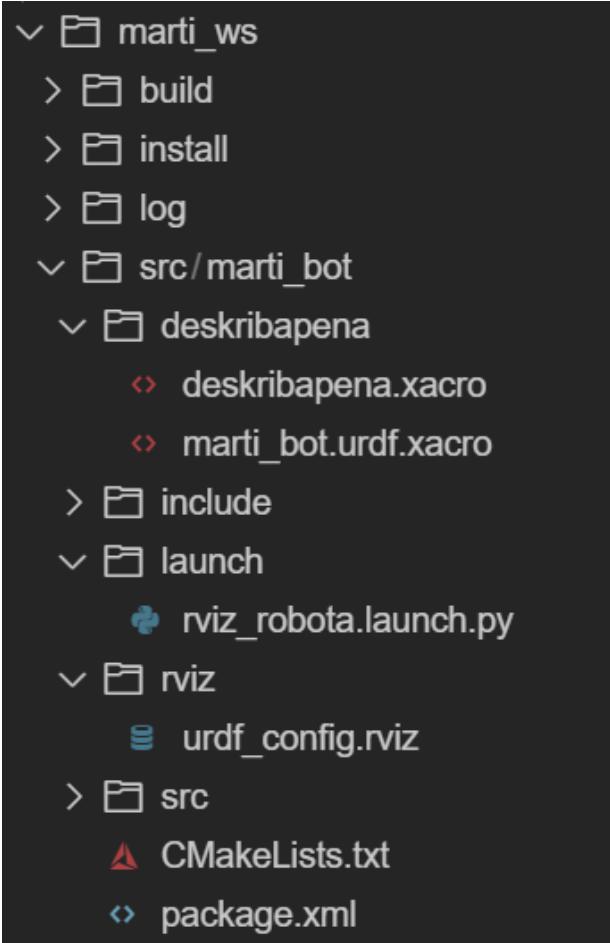
Lan-ingurunean instalatutako paketeak ezagutu eta eskuratu ahal izateko, komando hau kopiatu:

```
cd ~/marti_ws/  
source install/setup.bash
```

**6. Komandoa:** lan-espazioa kargatzeko komandoa.

Komando hau ere, lan-espazioaren barruan exekutatu behar da, proiektua erabili behar den terminal bakoitzean.

Orain arte egindakoarekin, Code Editor-era joanda, 3. irudian agertzen den egitura eduki beharko litzateke.



**3. Irudia:** proiektuaren egitura orain arte.

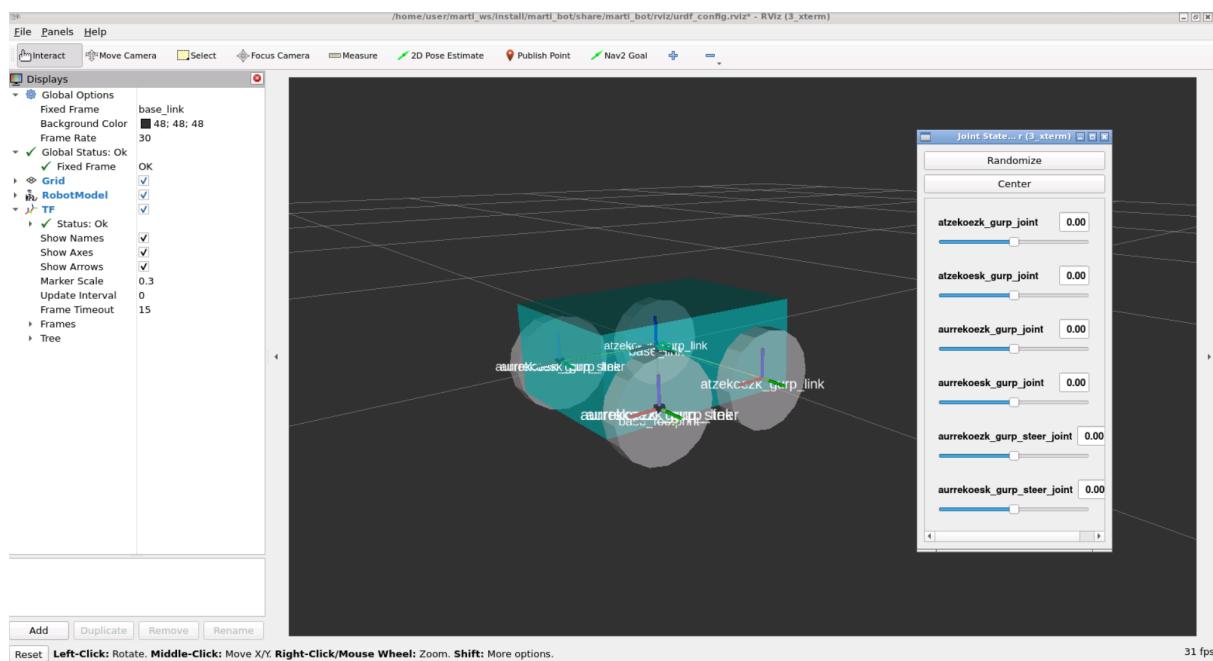
### 3.6 Jaurti robota RViz-en

Aurreko pausu guztiak jarraitu ostean, terminal berdinean exekutatu “rviz\_robota.launch.py” fitxategia:

```
ros2 launch marti_bot rviz_robota.launch.py
```

**7. Komandoa:** “launch” fitxategi bat abiarazteko komandoa.

Behin exekutatuta, RViz abiaratu beharko litzateke sortutako lau gurpildun robotarekin, 4. irudian ikus daitekeen bezala.



**4. Irudia:** lau gurpildun robota RViz-en.

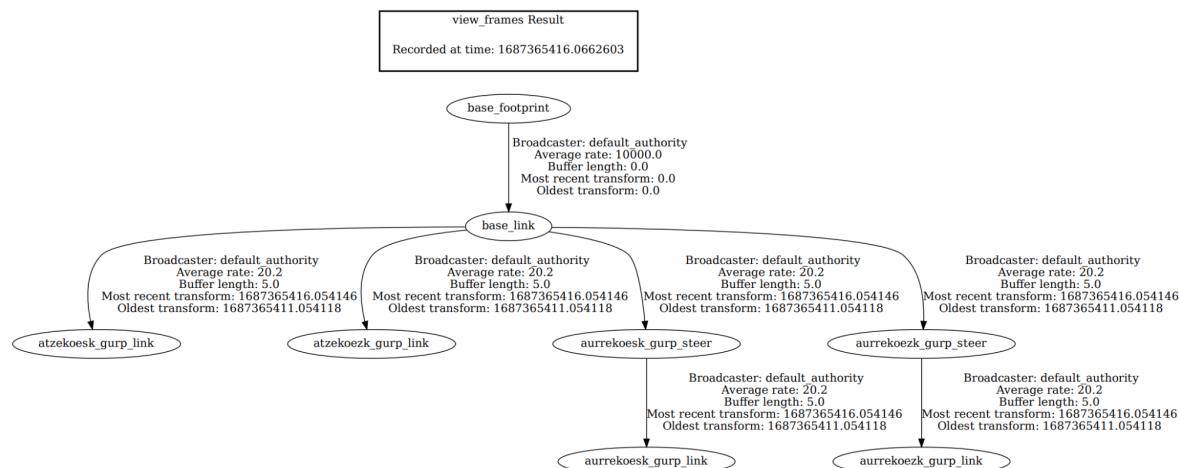
Robotarekin batera, leihoa bat ireki da. Hau, “start\_joint\_state\_publisher\_gui\_node” izeneko nodo bat da eta honi esker gurpil bakoitzaren zinematika ikustea ahalbidetuko du. Adibidez, “aurrekoesk\_gurp\_joint” artikulazioko baloreak aldatuz aurreko eskubiko gurpilak biratuko du. Eta “aurrekoesk\_gurp\_steer\_joint” artikulazioko baloreak aldatuz, gurpil berdinaren direkzioa (angelua) aldatu ahalko zaio.

Sortutako ibilgailuaren koordenatu-markoak ikusi ahal izateko, terminal berri bat ireki:

```
ros2 run tf2_tools view_frames
```

**8. Komandoa:** exekutagarri bat abiarazteko komandoa.

Aurretik aipatu den bezala, loturak modu hierarkiko batean egiten dira, base\_footprinta goian izanik eta lau gurpilen loturak zuhaitzaren azpian geratuz. Eta aurreko gurpilak direkzionalak direnez, “steer” izeneko lotura bat dute beraien gainetik, 5. irudian ikusgai.



**5. Irudia:** robotaren koordenatu-markoaren zuhaitza.

## 3.7 rqt\_graph

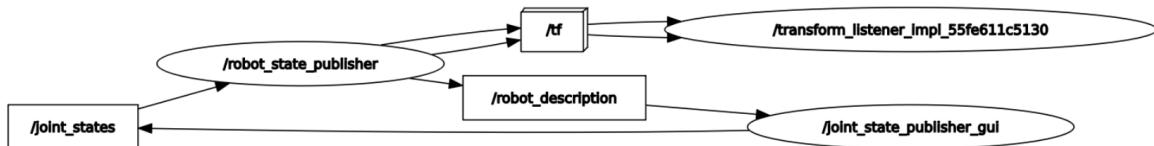
Tresna honen bitartez, aktibo dauden nodoen komunikazioa ikustea ahalbidetzen du.

Terminal bat ireki:

rqt\_graph

**9. Komandoa:** nodoen konexioak era grafiko batean ikusteko komandoa..

Komando hori exekutatu ostean, ireki berri den lehioaren goikaldean “Node/Topics (all)” aktibatu. Hori egin ondoren, nodo eta gaien komunikazioa era bisual batean ikusi beharko litzatek, 6. irudian ageri den bezala.



**6. Irudia:** nodoen komunikazioa grafikoki.

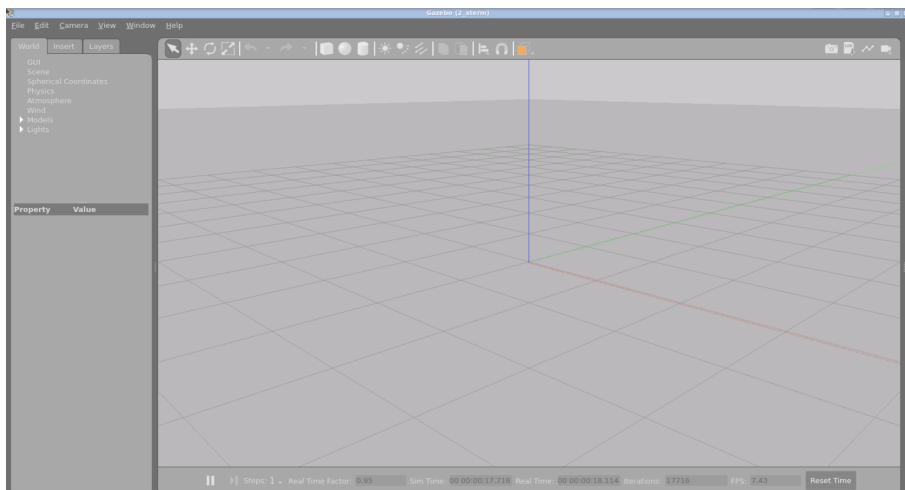
Goiko irudi honetan ikusi daiteke nola “/joint\_state\_publisher\_gui” nodoak “/joint\_state” gaiean datuak (gurpilaren mugimendua) publikatzen dituen. Datu horiek, “/robot\_state\_publisher” nodoak jasotzen ditu eta koordenatu-marko horien informazioa “tfn” argitaratzen ditu. Eta horrela, RVizen roboteko gurpilen posizio berria eguneratzen da.

## 4. GAZEBO

Gazebo, ROS 2ren barruan implementatutako aplikazio bat da, simulazio-ingurune bezala jokatzen duena. Honi esker, junki\_bot-aren algoritmoak probatu ahalko dira 3 dimentsiotan, robotak izango duen portaera ikusiz. Gazebo abiarazteko, terminal bat ireki eta izen bera idatzi:

gazebo

**10. Komandoa:** gazebo abiarazteko komandoa.



**7. Irudia:** Gazebo simulazio-ingurunea.

7. irudian, Gazeboaren simulazio-ingurunea ikus daiteke. Momentuz hutsik dago, baina hurrengo ataletan, sortutako robota bertan abiaraziko da. Aplikazio honi buruzko informazio ofiziala, bere webgunean aurki daiteke [7].

### 4.1 Ackermann Plugina

Gida hau egiten den momentuan, ROS 2k ez dauka Ackermann motako direkzioa kontrolatzen duen plugin ofizialik. Arazo horri aurre egiteko, GitHub plataformako plugin bat kopiatuko da proiektuaren barruan. Horretarako, terminal bat ireki:

cd ~/marti\_ws/src/

Ondoren, GitHub-eko pluginia lan-espazioko, src karpetaren barruan kopiatu:

```
git clone https://github.com/lukin1225/marti_gazebo_plugin.git
```

**11. Komandoa:** GitHub-eko gordailu bat kopiatzeko komandoa.

GitHub-eko edozein gordailu kopiatzeko komandoaren egitura:

```
git clone <GitHub-eko esteka>
```

**Gomendazioa:** Gida hau irakurtzen den momentuan, ROS 2k plugin ofizialik eskaintzen badu, hori bera erabiltzea gomendatzen da.

**Kodearean azalpena:**

Plugin honek aukera ematen du Ackermann (Marti\_bot) direkzioa duen robot baten portaera simulatzea ingurune birtual batean. Mugimendu-komandoak jasotzen ditu (/cmd\_vel) eta informazio garrantzitsua argitaratzen du, hala nola robotaren posizioa eta odometria, ROS 2 erabiliz. Honi esker, robotare mugimendua eta portaera kontrolatu ahal izango da.

Plugin hau ondo instalatu dela konprobatzeko, terminal bat ireki eta lan-espazioa konpilatuko da:

```
cd ~/marti_ws/  
colcon build
```

Baliteke, pluginia lehen aldiz konpilatzean ohartarazpen bat agertzea. Ez ikusiarena egin mezu horri eta bigarren aldiz konpilatu. Hori egin ondoren, paketeak lan-ingurunean instalatzeko:

```
source install/setup.bash
```

## 4.2 URDF

Gazebo simulazio-ingurunean robota ondo abiarazteko propietate espezifiko batzuk gehitu behar zaizkio eta aurreko puntuko pluginia gehitutako da ere. Horretarako terminal bat ireki eta deskribapena izeneko karpetan Xacro fitxategi berri bat sortu:

```
cd ~/marti_ws/src/marti_bot/deskribapena/  
touch gazebo_control.xacro
```

Egin klik [gazebo\\_control.xacro](#) atalean kode osoa ikusteko.

### Kodearen azalpena:

1. Kodearean hasieran, Gazebo simulazio-inguruneko lau gurpilen loturetarako propietate espezifikoak definitzen dira: marruskadura ( $\mu$ ), gurpilen kolorea, gutxieneko sakonera eta abar. Propietate horiek eragina dute robotaren gurpilen dinamika eta portaeraren simulazioan. Aurreko gurpilak, kolore morekoak jarri dira, direkziodunak direla bereizteko.
2. Kodearen amaieran "marti\_gazebo\_plugin" osagarria robotari gehitzen zaio, Gazebo ingurunean Ackermann motako direkziodun robotaren mugimendua kontrolatzeko.

Gogoratu: robotaren Xacro fitxategi guztiak `marti_bot.urdf.xacro` fitxategitik deitzen direla, era modularrago batean egoteko. Beraz ireki fitxategi hori eta gehitu sortutako azken Xacro artxiboa:

```
<xacro:include filename="gazebo\_control.xacro" />
```

## 4.3 Sortu mundua

Etxe itxurako mundu bat eraikiko da, robota ingurune horretan ibili dezan. Karpeta espezifiko bat sortuko da, bertan mundu desberdinak gorde daitezen eta bere barruan "etxea.world" izeneko fitxategia. Terminal bat ireki:

```
cd ~/marti_ws/src/marti_bot/  
mkdir worlds
```

```
cd worlds/  
touch etxea.world
```

Egin klik [etxea.world](#) atalean kode osoa ikusteko.

Kode hau, Gazebo simulagailuko eredu baten deskribapena da, SDF formatuan (Simulation Description Format). Gida honetan, ez du garrantzi berezirik fitxategi hau azaltzea. Honi buruzko informazio gehiago nahi ezkero, Gazebo dokumentazio ofizialean aurkitu daiteke [7].

Gogoan izan, “worlds” izeneko karpeta berri bat sortu dela eta konpilatu ahal izateko, [CMakeList.txt](#) eguneratu behar dela. Beraz, fitxategi horretan “worlds” karpeta gehitu.

```
install(  
    DIRECTORY src launch rviz deskribapena worlds  
    DESTINATION share/${PROJECT_NAME}  
)
```

## 4.4 Launch file

Lau gurpildun ibilgailua eta etxe itxurako mundua Gazebon martxan jartzeko, bi fitxategi sortuko dira. Lehenengo, “spawn\_car.launch.py” izeneko fitxategi bat sortuko da. Horretarako, “launch” izeneko direktoriora joan eta bertan sortu:

```
cd ~/marti_ws/src/marti_bot/launch/  
touch spawn_car.launch.py
```

Egin klik [spawn\\_car.launch.py](#) atalean kode osoa ikusteko.

## Kodearen azalpena:

“Launch” fitxategi honek hainbat nodo konfiguratzeko eta exekutatzeko ditu Gazebon robot bat simulatzeko.

1. Paketeen importaketa: Beharrezko moduluak importatzen dira, adibidez "os", sistemaren eragiketetarako, "ament\_index\_python.packages" ROS paketeei buruzko informazioa lortzeko, "launch\_ros" jaurtiketaren barruan ROS nodoak exekutatzeko eta abar.
2. "Generate\_launch\_description": Funtzio hau konfigurazio-fitxategiaren pieza nagusia da eta abiarazte-deskribapen bat itzultzen du (LaunchDescription). Funtzio horren barruan, beharrezko konfigurazioak eta definizioak egingo dira, nahi diren nodoak eta ekintzak abiarazteko.
3. Direkzioa: Robotaren Xacro fitxategiaren ibilbidea zehazten da eta robotaren paketea instalatzeko direktorioa lortzen da.
4. Aldagaien konfigurazioa: "GAZEBO\_MODEL\_PATH" eta "GAZEBO\_PLUGIN\_PATH" ingurune-aldagaiak eguneraztzen dira, marti\_bot paketearen instalazio-direktorioa lortzeko.
5. Xacro: Xacro modulua erabiltzen da robota deskribatzeko duen fitxategia prozesatzeko eta honen deskribapena (robot\_desc) XML formatuan lortzeko.
6. Nodoak: Exekuzioan zehar erabiliko diren nodoak definitzen dira. Adibidez, gazebo\_ros paketearen spawn\_entity.py nodoa erabiltzen da Gazebon modelo bat sortzeko, robotaren deskribapena erabiliz (robot\_desc).
7. “LaunchDescription”: Aurretik deklaratutako nodo eta ekintza guztiak, “LaunchDescription” gehitzen zaizkio, eta emaitza gisa itzultzen da.

Orain, “marti\_gazebo.launch.py” izeneko fitxategi bat sortuko da:

```
cd ~/marti_ws/src/marti_bot/launch/  
touch marti_gazebo.launch.py
```

Egin klik [marti\\_gazebo.launch.py](#) atalean kode osoa ikusteko.

#### Kodearen azalpena:

Aurreko “launch” fitxategiak duen egitura berdina dauka gutxi gorabehera. Baino hemendik, “etxea.world” fitxategia deklaratu eta abiarazten da. Gainera, ROS 2k Gazeboko ingurunea konfiguratzeko duen fitxategia eta aurreko “launch” fitxategia abiarazten dira.

## 4.5 Martxan jarri

Oraintxe bertan sortutako “spawn\_marti.launch.py” fitxategia abiarazteko, lehendabizi paketea konpilatu:

```
cd ~/marti_ws/  
colcon build
```

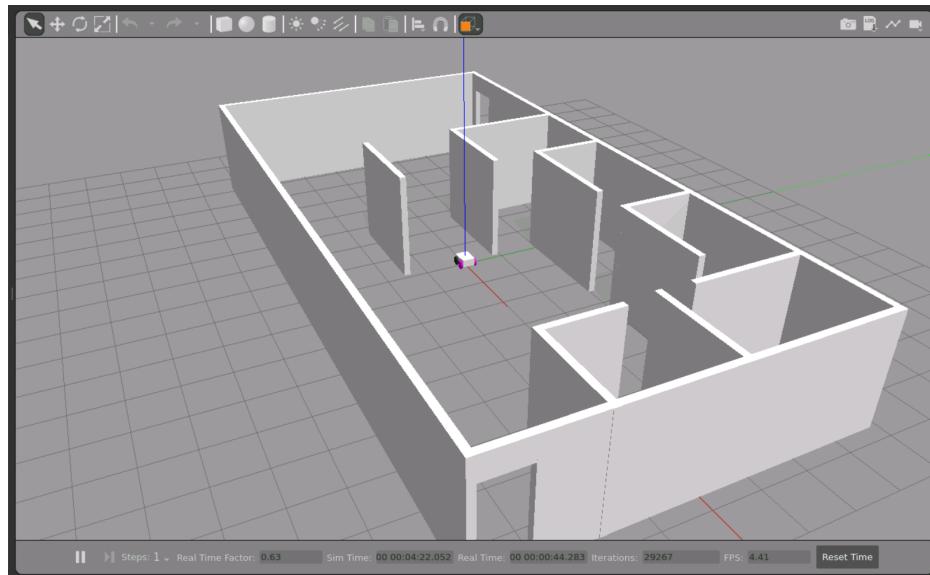
Paketeak lan-ingurunean instalatzeko:

```
source install/setup.bash
```

Eta orain bai, fitxategia exekutatu:

```
ros2 launch marti_bot marti_gazebo.launch.py
```

Behin hori exekutatuta, 8. irudian agertzen den bezala, lau gurpileko robota eta sortutako “world” fitxategia, gazebon ikusi beharko lirateke.



8. Irudia: marti\_bot-a Gazebo simulazio-ingurunean.

Goiko irudian ikus daitekeen bezala, marti\_bot-a Gazebon abiarazi da eta [gazebo\\_control.xacro](#) kodean konfiguratu den bezala, aurreko gurpilak kolore morekoak dira, bereizteko direkziodunak direla.

## 4.6 Nodoen zerrenda:

Aktibo dauden nodo guztien zerrenda bat ikusteko, terminal bat ireki:

```
ros2 node list
```

12. Komandoa: aktibo dauden nodoen zerrenda bat ikusteko komandoa.

Erantzuna:

```
/gazebo  
/marti_gazebo_plugin  
/robot_state_publisher
```

Nodoen artean, ikus daiteke robotaren direkzioa kontrolatuko duen pluginia aktibo dagoela, hurrengo atalean honi buruzko informazioa azalduko da.

## 4.7 Nodoen informazioa

“Info” komandoari esker, aktibo dauden nodoen informazioa jaso daiteke. Adibidez, “/marti\_gazebo\_plugin” ikusteko, terminal bat ireki:

```
ros2 node info /marti_gazebo_plugin
```

**13. Komandoa:** nodo baten informazioa ikusteko komandoa.

Erantzuna:

```
/marti_gazebo_plugin
```

Subscribers:

```
/clock: rosgraph_msgs/msg/Clock
/cmd_vel: geometry_msgs/msg/Twist
/joy: sensor_msgs/msg/Joy
/marti/cmd_ackermann: ackermann_msgs/msg/AckermannDriveStamped
/parameter_events: rcl_interfaces/msg/ParameterEvent
```

Publishers:

```
/joint_states: sensor_msgs/msg/JointState
/marti/cmd_ackermann: ackermann_msgs/msg/AckermannDriveStamped
/marti/odo_fl: std_msgs/msg/Int32
/marti/odo_fr: std_msgs/msg/Int32
/marti/odom: nav_msgs/msg/Odometry
/marti/pose: geometry_msgs/msg/PoseStamped
/parameter_events: rcl_interfaces/msg/ParameterEvent
/rosout: rcl_interfaces/msg/Log
/tf: tf2_msgs/msg/TFMessage
```

Service Servers:

```
/marti_gazebo_plugin/describe_parameters: rcl_interfaces/srv/DescribeParameters
/marti_gazebo_plugin/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
/marti_gazebo_plugin/get_parameters: rcl_interfaces/srv/GetParameters
/marti_gazebo_plugin/list_parameters: rcl_interfaces/srv/ListParameters
/marti_gazebo_plugin/set_parameters: rcl_interfaces/srv/SetParameters
/marti_gazebo_plugin/set_parameters_atomically:
rcl_interfaces/srv/SetParametersAtomically
```

Service Clients:

### Action Servers:

### Action Clients:

Hemendik informazio ugari atera daiteke. Alde batetik, nodo hau zein gaietatik informazioa jasotzen duen, eta gainera informazio hori ze motatakoa den adierazten du. Adibidez, “/cmd\_vel” gaira harpidetuta dago.

Beste alde batetik, nodo hau argitalatzaile bezala jokatzen duela adieratzen du. Adibidez, “/odom” izeneko gaiean robotaren posizioaren datuak bidaltzen ditu. Hau, aurrerago beharrezkoa izango da, ibilgailua bakarrik nabegatu dezan.

## 4.8 Gaien zerrenda

Aktibo dauden gai guztien zerrenda bat ikusteko, terminal bat ireki:

ros2 topic list

**14. Komandoa:** aktibo dauden gaien zerrenda bat ikusteko komandoa.

Erantzuna:

```
/clock  
/cmd_vel  
/joint_states  
/joy  
/marti/cmd_ackermann  
/marti/odo_fl  
/marti/odo_fr  
/marti/odom  
/marti/pose  
/parameter_events  
/performance_metrics  
/robot_description  
/rosout  
/tf  
/tf_static
```

Zerrenda honetan, “/cmd\_vel” izeneko gaia dago. Marti\_bot ibilgailuari abiadura linealeko eta angeluarreko komandoak bidaltzeko erabiltzen da. Aurreko puntuau (4.7) aipatu den bezala, “/marti\_gazebo\_plugin” nodoa, gai horretara harpidetuta dago.

Lista horretan, aurreko atalean (4.7) aipatutako “/odom” gaia ageri da, eta robot mugikorraren odometria datuak argitaratzeko erabiltzen da, hau Ackermann Plugina (4.1) ataleko gordailuan konfiguratuta dago.

Hurrengo atalean, “/cmd\_vel” buruzko gaiaren informazioa aztertuko da.

## 4.9 Gaien informazioa

Aktibo dauden gai guztiak informazioa ikusteko “info” komandoa erabiltzen da. Adibidez, “/cmd\_vel” gaiarena ikusteko, terminal batean:

```
ros2 topic info /cmd_vel
```

**15. Komandoa:** gai baten informazioa ikusteko komandoa.

Erantzuna:

```
Type: geometry_msgs/msg/Twist
Publisher count: 0
Subscription count: 1
```

Lehenengo lerroak, gai horretan informazioa argitaratzeko edo jasotzeko zer motatako mezuak izan behar diren adierazten du. Ondoren, “topic-ak” harpidedun bat eta zero argitalatzaile duela esaten du. Ondorengo atalean, gai honetan informazioa argitaratuko duen nodo bat exekutatuko da, robota teklatuaren bitartez kontrolatuko duena.

Beste gai baten informazioa ikusteko, adibidez “/marti/odo\_fl”:

```
ros2 topic info /marti/odo_fl
```

Erantzuna:

Type: std\_msgs/msg/Int32

Publisher count: 1

Subscription count: 0

Kasu honetan ikus daiteke gai honen datu mota desberdina dela. Gainera, aurrekoarekin konparatuta, ez du harpidetuta dagoen nodorik, baina bertan argitalatzen duen nodo bat bai.

## 4.10 Robota kontrolatu

Gaien informazioaren atalean ikusi den bezala, /cmd\_vel “topic-a”, simulatutako ibilgailuei abiadura linealeko eta angeluarreko komandoak bidaltzeko erabiltzen da. Bainaz momentuz, ez du bertan argitaratzen duen nodorik, soluzio gisa terminal berri bat ireki eta hurrengo exekutagarria abiatu:

```
ros2 run teleop_twist_keyboard teleop_twist_keyboard
```

Erantzuna:

This node takes keypresses from the keyboard and publishes them as Twist messages. It works best with a US keyboard layout.

Moving around:

u	i	o
j	k	l
m	,	.

For Holonomic mode (strafing), hold down the shift key:

U	I	O
J	K	L
M	<	>

t : up (+z)  
b : down (-z)

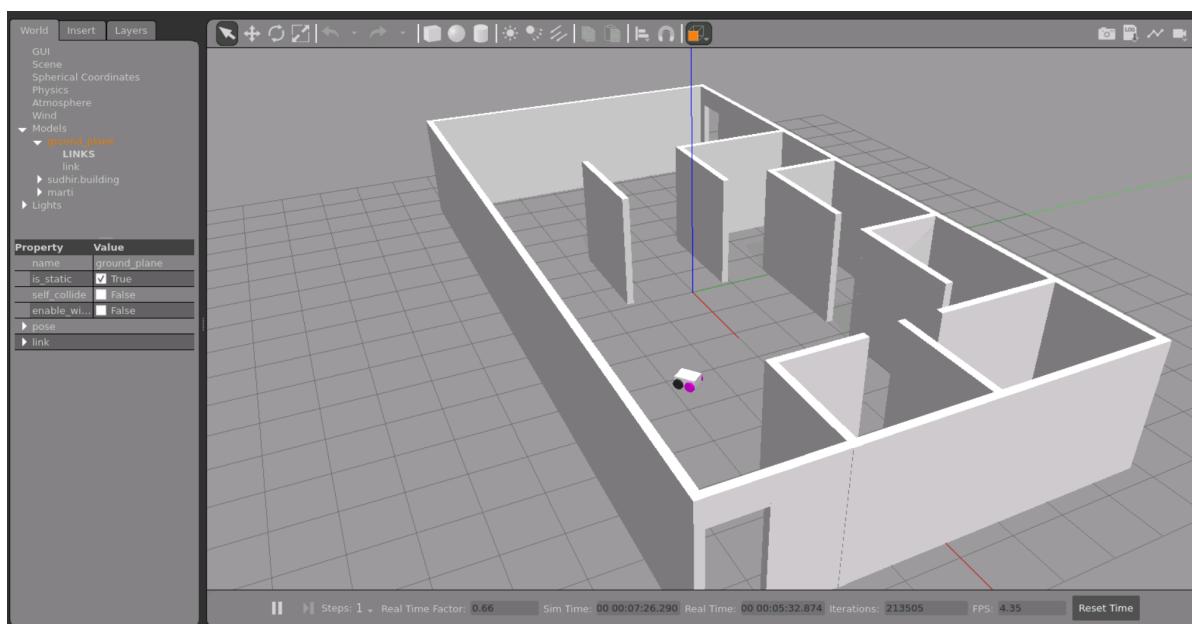
anything else : stop

q/z : increase/decrease max speeds by 10%  
 w/x : increase/decrease only linear speed by 10%  
 e/c : increase/decrease only angular speed by 10%

CTRL-C to quit

currently: speed 0.5 turn 1.0

Orain, teklatua erabiliz ibilgailua kontrolatu ahal izango da. 9. irudian ikus daiteke, nodo honi esker marti\_bot-a mugitu dela.



**9. Irudia:** marti\_bot-aren posizio berria.

Aktibo dauden nodoen zerrendari erreparatuz, exekutatu berri den nodoa agertu beharko litzateke. Terminal bat ireki:

ros2 node list

Erantzuna:

```
/gazebo
/marti_gazebo_plugin
/robot_state_publisher
/teleop_twist_keyboard
```

Beraz, ikusi daiteke “/teleop\_twist\_keyboard” nodoa zerrenda honetara gehitu dela.

Nodo honi buruzko informazio gehiago ikusteko:

```
ros2 node info /teleop_twist_keyboard
```

Erantzuna:

/teleop\_twist\_keyboard

Subscribers:

Publishers:

/cmd\_vel: geometry\_msgs/msg/Twist

/parameter\_events: rcl\_interfaces/msg/ParameterEvent

/rosout: rcl\_interfaces/msg/Log

Service Servers:

/teleop\_twist\_keyboard/describe\_parameters: rcl\_interfaces/srv/DescribeParameters

/teleop\_twist\_keyboard/get\_parameter\_types: rcl\_interfaces/srv/GetParameterTypes

/teleop\_twist\_keyboard/get\_parameters: rcl\_interfaces/srv/GetParameters

/teleop\_twist\_keyboard/list\_parameters: rcl\_interfaces/srv/ListParameters

/teleop\_twist\_keyboard/set\_parameters: rcl\_interfaces/srv/SetParameters

/teleop\_twist\_keyboard/set\_parameters\_atomically:

rcl\_interfaces/srv/SetParametersAtomically

Service Clients:

Action Servers:

Action Clients:

Nodo honek, ez du daturik jasotzen beste gai batengandik. Baino, informazioa “/cmd\_vel” gaieran argitalatzen du beste batzuen artean, robotaren mugimendua ahalbidetuz.

Beraz, “/cmd\_vel” gaiak eskaintzen duen informazioa berriz erreparatuz:

```
ros2 topic info /cmd_vel
```

Erantzuna:

Type: geometry\_msgs/msg/Twist

Publisher count: 1

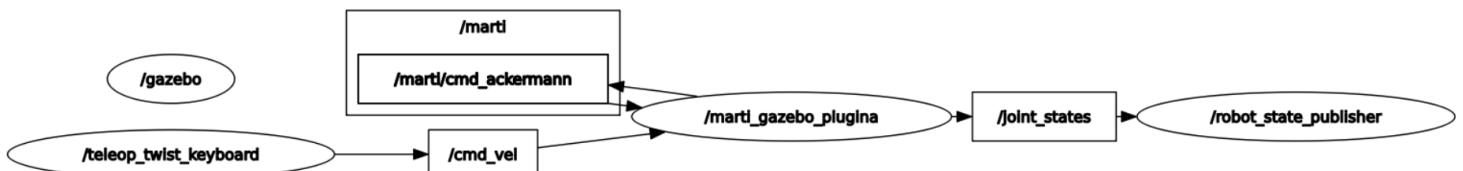
Subscription count: 1

Komentatu den bezala, `/teleop_twist_keyboard` nodoa exekutatu denetik, orain “topic” honek argitalatzale bat du, nodo bera hain zuzen ere.

## 4.11 Rqt\_graph

Orain arte aipatu diren nodoen eta haien arteko conexioen irudikapen grafiko bat emateko, “rqt\_graph” komandoa erabiliko da. Horretarako terminal berri bat ireki:

rqt\_graph



10. Irudia: nodoen conexioen irudikapena.

10. irudian nodoen conexioa ikus daiteke era grafiko batean. Aurreko atalean aipatu den bezala, “/teleop\_twist\_keyboard” nodoak datuak bidaltzen ditu “/cmd\_vel gaien”. Baino robotaren direkzioa Ackermann motakoa denez, “/marti\_gazebo\_plugin” gehitu da mugimendu mota hori kontrolatzeko. Azkenik, datu guzti hauek “/robot\_state\_publisher” nodora bidaltzen dira, non honek robotaren egoera berria eguneratzen du.

## 5. NABEGAZIOA

Nabigazioa, robot bat puntu batetik bestera autonomiaz mugitzeko aukera ematen duen tresna da. Gida honetan Navigation2 [3] erabiliko da. Nabigaziorako robotak bi gauza jakin behar ditu: lehengoa, non dagoen (*localization*), eta bigarrena, nolakoa den mapa (mapping).

1. Lokalizazioa (*localization*): Honen helburua robotaren egungo posizioa eta orientazioa denbora errealean zehaztasunez kalkulatzea da. Hori lortzeko, odometria eta sentsoreak (laserrak adibidez) erabiltzen dira, robotaren posizio erlatiboa hautemateko eta ulertzeko. Odometria ibilgailu gurpildunen posizioaren estimazioaren azterketa da nabigazioan zehar.
2. Mapaketa (mapping): Hau, robota dagoen inguruneko eredu bat sortzeko prozesua da, ingurunearren egitura zehatz bat irudikatuz, oztopoak, hormak, atea eta abar barne. Gida honetan bi erabiliko dira, SLAM edo AMCL.

Nabegazioari buruzko informazio gehiago eta zehatzagoa, Navigation 2ren orrialde ofizialean aurki daiteke [3].

### 5.1 Lokalizazioa

#### 5.1.1 LIDAR sentsorea

LiDAR sentsorea distantziak neurtzeko eta bere inguran duenaren irudikapenak sortzeko erabiltzen den gailua da, laser argizko pultsuak erabiliz. Beraz, gailu honen bidez une oro jakingo da non dagoen robota.

Gailu hau marti\_bot-ean gehitzeko, xacro fitxategi berri bat eraikiko da deskribapena izeneko karpetan. Horretarako, terminal bat ireki:

```
cd marti_ws/src/marti_bot/deskribapena/  
touch lidar.xacro
```

Egin klik lidar.xacro atalean kode osoa ikusteko.

#### Kodearen azalpena:

Laburbilduz, lehenengo bisualki sortzen da LIDARRA, bere inertziekin, talkekin, loturekin eta abar. Ondoren, LIDAR puglina integratzen da, bere inguruko oztopoei buruzko eskaneoak egin eta datu horiek "/scan" topikoan argitara ditzan. Gazeboko plugin-ei buruzko informazio gehiago, webgune ofizialean aurkitu daiteke [7].

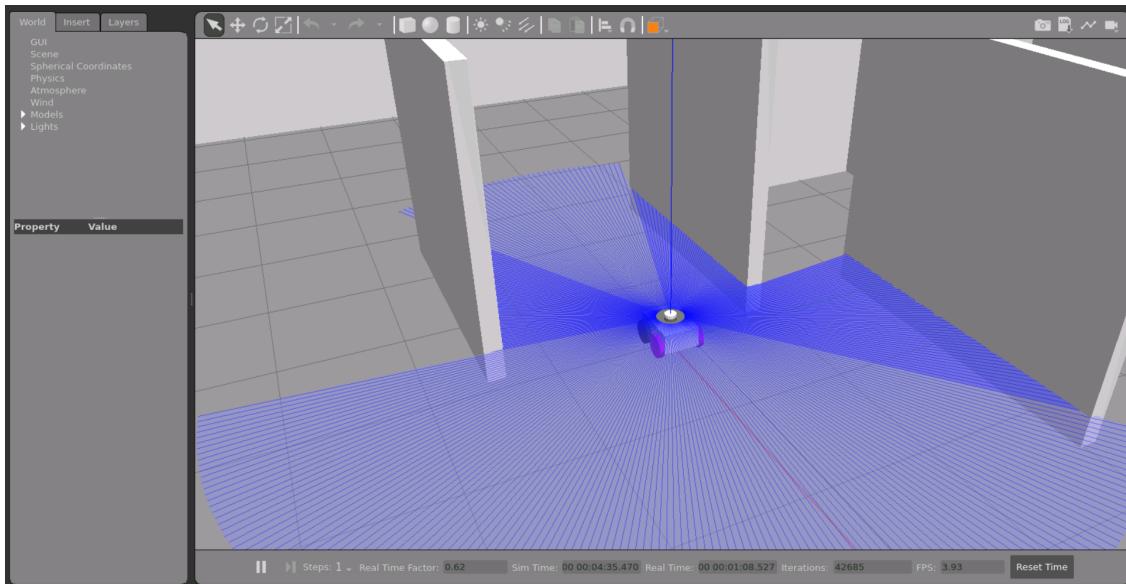
Laser-sentsorea robotean integratzeko, oraindik pausu bat falta da, oraintxe sortutako xacro fitxategia Marti\_bot.urdf.xacro fitxategian gehitzea:

```
<xacro:include filename="lidar.xacro" />
```

#### 5.1.2 Gazebo

Guzti hau behar bezala konfiguratu dela ziurtatzeko, Gazebon bistaratuko da robota. Horretarako terminal batean, konpilatu eta jaurti fitxategia:

```
cd marti_ws/  
colcon build  
source install/setup.bash  
ros2 launch marti_bot marti_gazebo.launch.py
```



11. Irudia: Marti\_bot-a LIDAR sentsorearekin.

Terminal berri bat ireki:

```
ros2 node list
```

Erantzuna:

```
/gazebo
/marti_gazebo_plugin
/robot_state_publisher
/scan
```

Ikusi daiteke, “/scan” izeneko nodo berri bat agertu dela eta honi buruzko informazioa jasotzeko:

```
ros2 node info /scan
```

Erantzuna:

```
/scan
Subscribers:
/clock: rosgraph_msgs/msg/Clock
/parameter_events: rcl_interfaces/msg/ParameterEvent
```

Publishers:

/parameter\_events: rcl\_interfaces/msg/ParameterEvent  
/rosout: rcl\_interfaces/msg/Log  
/scan: sensor\_msgs/msg/LaserScan

Service Servers:

/scan/describe\_parameters: rcl\_interfaces/srv/DescribeParameters  
/scan/get\_parameter\_types: rcl\_interfaces/srv/GetParameterTypes  
/scan/get\_parameters: rcl\_interfaces/srv/GetParameters  
/scan/list\_parameters: rcl\_interfaces/srv/ListParameters  
/scan/set\_parameters: rcl\_interfaces/srv/SetParameters  
/scan/set\_parameters\_atomically: rcl\_interfaces/srv/SetParametersAtomically

Service Clients:

Action Servers:

Action Clients:

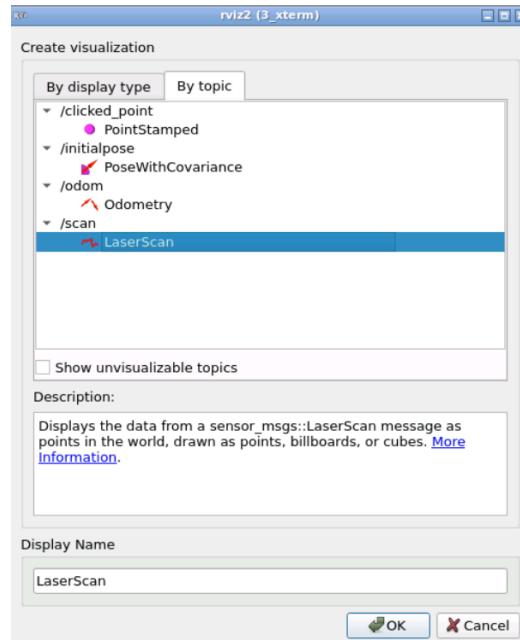
Nodo honen informazioari erreparatuz, ikus daiteke “sensor\_msgs/msg/LaserScan” motatako datuak “/scan” izeneko gaiean argitaratzen dituela.

### 5.1.3 RViz

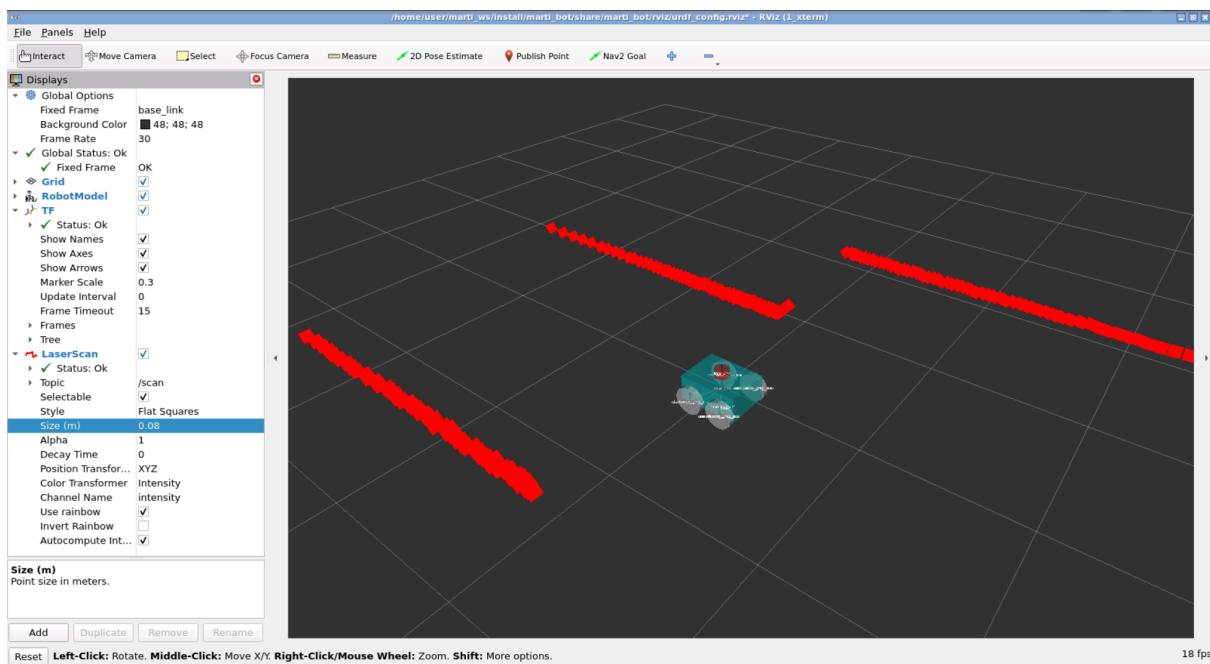
Laserra ondo konfiguratu dela ziurtatzeko, robota RVizen irekiko da, horretarako beste terminal bat ireki:

```
cd marti_ws/  
source install/setup.bash  
ros2 launch rviz_robota.launch.py
```

Aurreko atalean kodea konpilatu denez eta aldaketarik egin ez denez, bakarrik terminalean paketeak egunerautu dira. Behin, RViz ireki eta robota bertan ageri dela, laserraren funtzionamendua konprobatzeko, egin klik RViz leiharen beheko aldean dagoen “ADD” izeneko botoian. Ondoren, “By topic” atalera joan eta “LaserScan” aukera hautatu 12. irudian ikus daitekeen bezala.



**12. Irudia:** LaserScan aktibatu RVizen.



**13. Irudia:** oztopoen detekzioa RVizen, lidarrari esker.

13. irudian ikus daitekeenez, LIDAR sentsorea behar bezala konfiguratu da, eta, horri esker, robota bere ingurunean oztopoak detektatzen ari dela ikus daiteke.

## 5.2 Mapaketa

### 5.2.1 Gehitu nabegazio-parametroak

Orain, ROS 2 nabegazio parametroak gehituko dira. Nahiz eta parametro ugari egon, garrantzitsuenak Costmap 2D paketearenak dira. Pakete hau, sare-zelula ugariz osatutako mapa bat da eta Navigation 2 [3] paketeak bi motatakoak erabiltzen ditu inguruneko oztopoen informazioa gordetzeko:

1. Global costmap: kostu-mapa hau epe luzerako planak sortzeko erabiltzen da, adibidez, A puntutik B puntura bitarteko biderik laburrena kalkulatzeko.
2. Local costmap: kostu-mapa hau epe motzeko planak sortzeko erabiltzen da, adibidez, oztopoak saihesteko.

Honi buruzko informazio gehiago Nav2ren dokumentazio ofizialean aurki daiteke [3].

Terminal bat ireki, eta proiektuaren barruan “params” izeneko karpeta berri bat sortu, eta bere barruan nabegazio-parametroak konfiguratuko duen fitxategia sortu:

```
cd marti_ws/src/marti_bot/  
mkdir params  
cd params  
touch nav2_params.yaml
```

Egin klik [nav2\\_params.yaml](#) atalean kode osoa ikusteko. Fitxategi hau Marti\_bot robotaren nabegazioa planifikatzeko eta kontrolatzeko gaitasunak ematen ditu. Gida Ofizialak parametro guztien konfigurazioak xehatzen ditu, Nav2 [3].

Baina “ROS2ren Oinarrizko Gida II”-ko fitxategiarekin konparatuz, desberdintasun nabarienak honako hauek dira:

1. Kontrolagailua: Beste gidakoak "dwb\_core:: DWBLocalPlanner" pluginia erabiltzen du, eta gida honetakoak, berriz, RPP pluginia "nav2\_regulated\_pure\_pursuit\_controller:: RegulatedPurePursuitController". Biak beharrezkoak dira nabigaziorako, eta desberdintasun bakarra da

algoritmo desberdinak erabiltzen dituztela robotaren mugimendu seguru eta zehatza lortzeko.

2. Ibilbide-planifikatzailea: Kasu honetan, "nav2\_smac\_planner/SmacPlannerHybrid" pluginia erabiltzen da planifikatzaile bezala. Honi esker, robotak atzera joateko ahalmena izango du eta gainera, sortutako ibilbideak intuitiboagoak izango dira.

Gogoan izan, "params" karpeta berri bat sortu dela eta konpilatu ahal izateko, CMakeList.txt eguneratu behar dela. Beraz, fitxategi horretan "params" karpeta gehitu.

```
install(  
  DIRECTORY src launch rviz deskribapena worlds params  
  DESTINATION share/${PROJECT_NAME}  
)
```

## 5.2.2 RViz nabegazioa

Robotarekin nabegatu ahal izateko, RViz fitxategian konfigurazio batzuk egin behar dira. Beraz, terminal bat ireki eta RViz karpetara joan:

```
cd marti_ws/src/marti_bot/rviz/  
touch nav2_default_view.rviz
```

Egin klik [nav2\\_default\\_view.rviz](#) atalean kode osoa ikusteko.

### Kodearen azalpena:

Laburbilduz, fitxategi honetan RViz konfiguratu egin da, nabigazioa ahalbidetuko duten panelen eta bistaratzeen konfigurazioa deskribatuz. Bertan, zer panel erakutsiko diren eta zer bistaratze sartuko diren zehazten dira, eta bistaratze bakoitzaren propietateak eta parametroak ere. "Robot Operating System 2ren Oinarrizko Gida II"-rekin konparatuta, kasu honetan robota hasieratik RViz-en ikusgai azaltzeko ahalbidetuta dago.

### 5.2.3 Launch fitxategia

Terminal bat ireki eta “launch” karpetaren barruan fitxategi berri bat sortu:

```
cd martii_ws/src/marti_bot/launch/  
touch rviz.launch.py
```

Fitxategi hau, RViz hasteaz arduratzen da nabigazioa bistaratzeko. Egin klik [rviz.launch.py](#) atalean kode osoa ikusteko.

Orain beste fitxategi bat sortuko da “launch” karpetaren barruan, horretarako berriz terminal batean:

```
cd marti_ws/src/marti_bot/launch/  
touch nabegazioa.launch.py
```

Honek, RViz aplikazioa abiatuko du, robotaren nabigazioa ahalbidetuko duen konfigurazio guztiarekin. Fitxategi hau funtzionatu dezan, lehenengo [marti\\_gazebo.launch.py](#) fitxategia abiarazi behar da.

Egin klik [nabegazioa.launch.py](#) atalean kode osoa ikusteko.

#### Kodearen azalpena:

Aurreko “launch” fitxategiekin konparatuz, kasu honetan nabegazioari dagokion konfigurazio batzuk ezarriko dira.

1. “nav2 Bringup” direktoriorako bidea bilatu eta biltegiratzen du, aurrerago jaurtiketa-fitxategian erabiltzeko.
2. “DeclareLaunchArgument” klasea erabiliz, SLAM zehaztu da “1” (true) bezala, ingurune ezezagunaren mapa estatiko bat eraikitzea, robota mapa horren barruan mugitzen den bitartean
3. Azkenik, “bringup\_launch.py” nabegazio-fitxategia ejekutatzen da, hainbat argudio desberdinak, haien artean SLAM.

Kodea exekutatu aurretik, [package.xml](#) fitxategia egunerautu behar da, nabegazioari buruzko dependentziak gehituz:

```
<build_depend>nav2_common</build_depend>
<build_depend>navigation2</build_depend>
<build_depend>launch_ros</build_depend>

<exec_depend>launch_ros</exec_depend>
<exec_depend>navigation2</exec_depend>
<exec_depend>nav2_common</exec_depend>
<exec_depend>slam_toolbox</exec_depend>
<exec_depend>robot_localization</exec_depend>
```

#### 5.2.4 SLAM exekutatu

Terminal bat ireki eta lan-espazioa konpilatu:

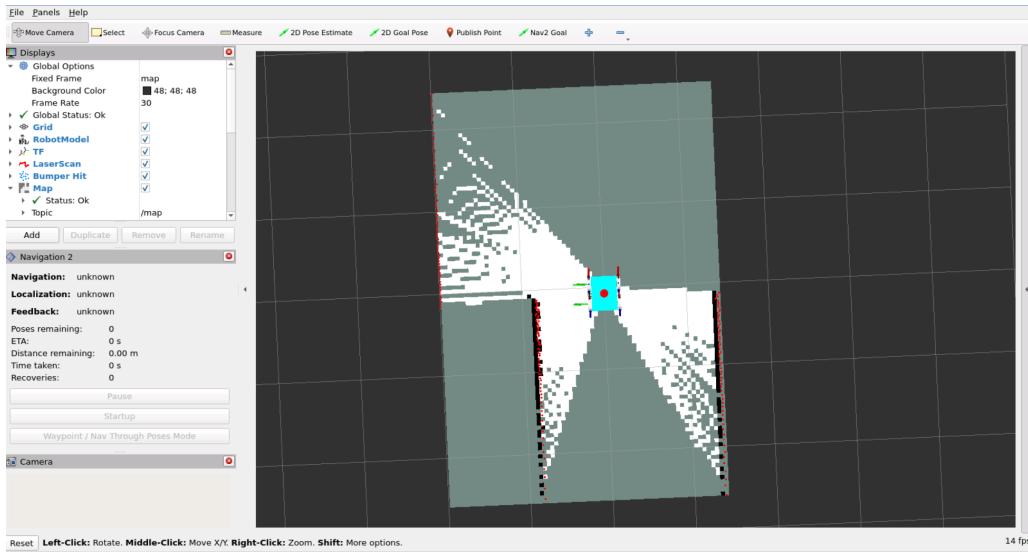
```
cd marti_ws/
colcon build
source install/setup.bash
```

Terminal berri batean, Marti\_bot-a Gazeboko ingurunean abiarazten duen fitxategia jaurti:

```
ros2 launch marti_bot marti_gazebo.launch.py
```

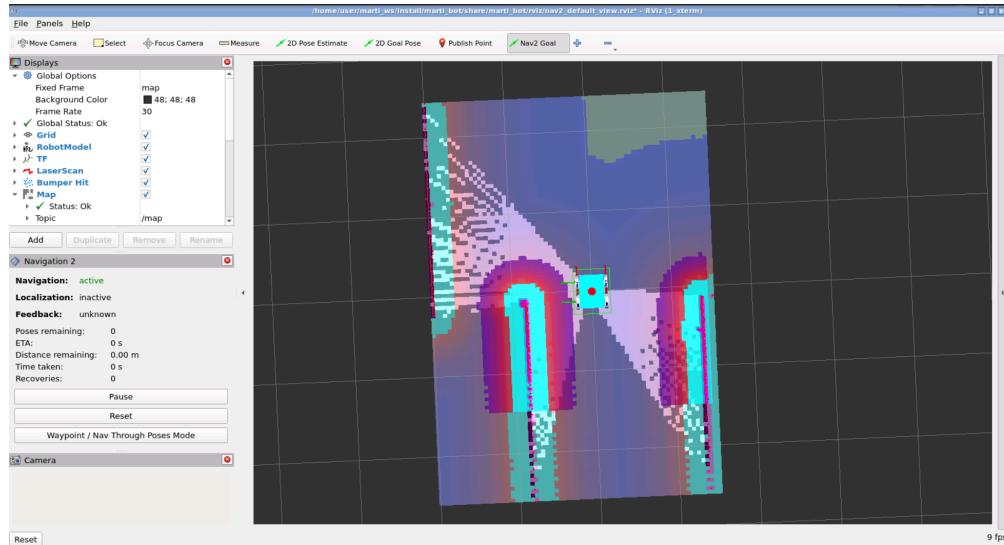
Fitxategi hau ez da berria; beraz, ikusten denaren adibidea 11. irudian dago. Beste terminal bat ireki:

```
source install/setup.bash
ros2 launch marti_bot nabegazioa.launch.py
```



14. Irudia: marti\_bot-a SLAMarekin.

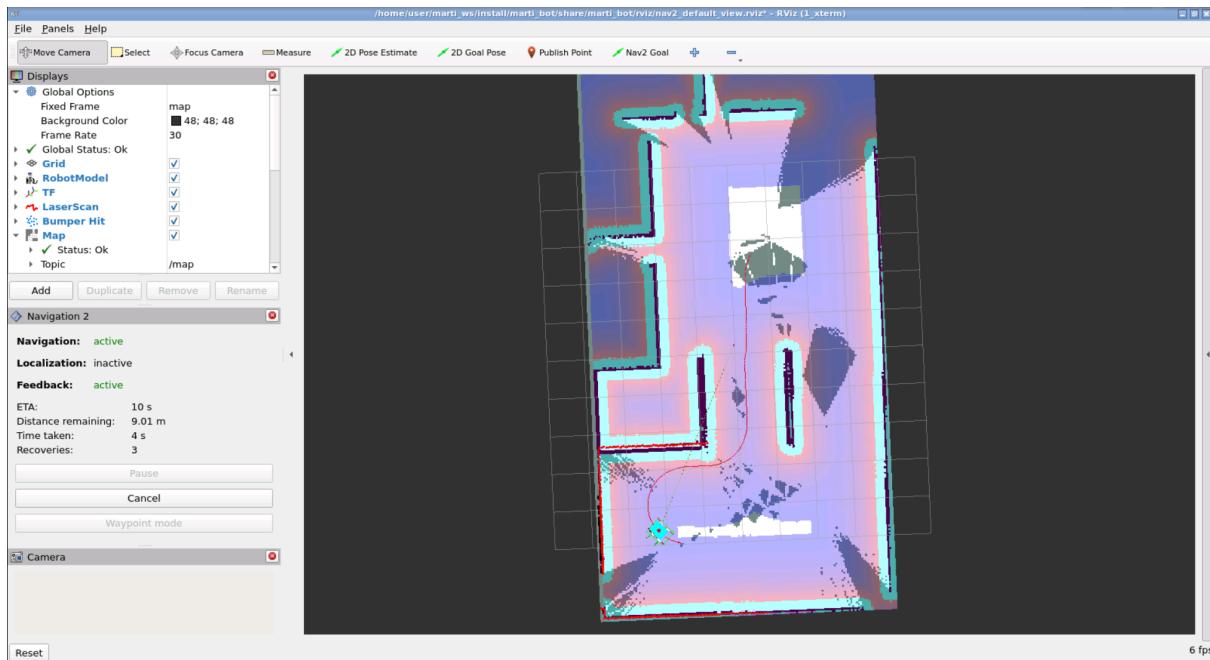
Nabegaziorako erabiltzen ari den planifikatzaileari kosta egiten zaio kargatzea, beraz hasieran 14. irudian bezala agertu beharko litzateke, baina denbora bat igaro ahala 15. irudian ageri den bezala, nabegazioarekin kargatuta agertu beharko litzateke.



15. Irudia: marti\_bot-a nabegazioarekin.

15. irudian ikus daitekeen bezala, RViz aplikazioan marti\_bot-a ageri da SLAMarekin. Beraz, guzti hau izanda, marti-bot-a puntu batetik beste batera nabigatzeko prest dago. Horretarako, goiko aldeko "Nav2 Goal" tresna erabiliz, robota leku berri batera joateko agindua eman daiteke.

16. irudian ikus daitekeen bezala, robota mapako leku berri batean dago eta linea gorri batez markaturik dauka berak autonomikoki egingo duen ibilbidea.



16. Irudia: marti\_bot-a nabegatzen puntu berri batera.

Sortutako ibilgailuaren koordenatu-markoak ikusi ahal izateko, terminal berri bat ireki:

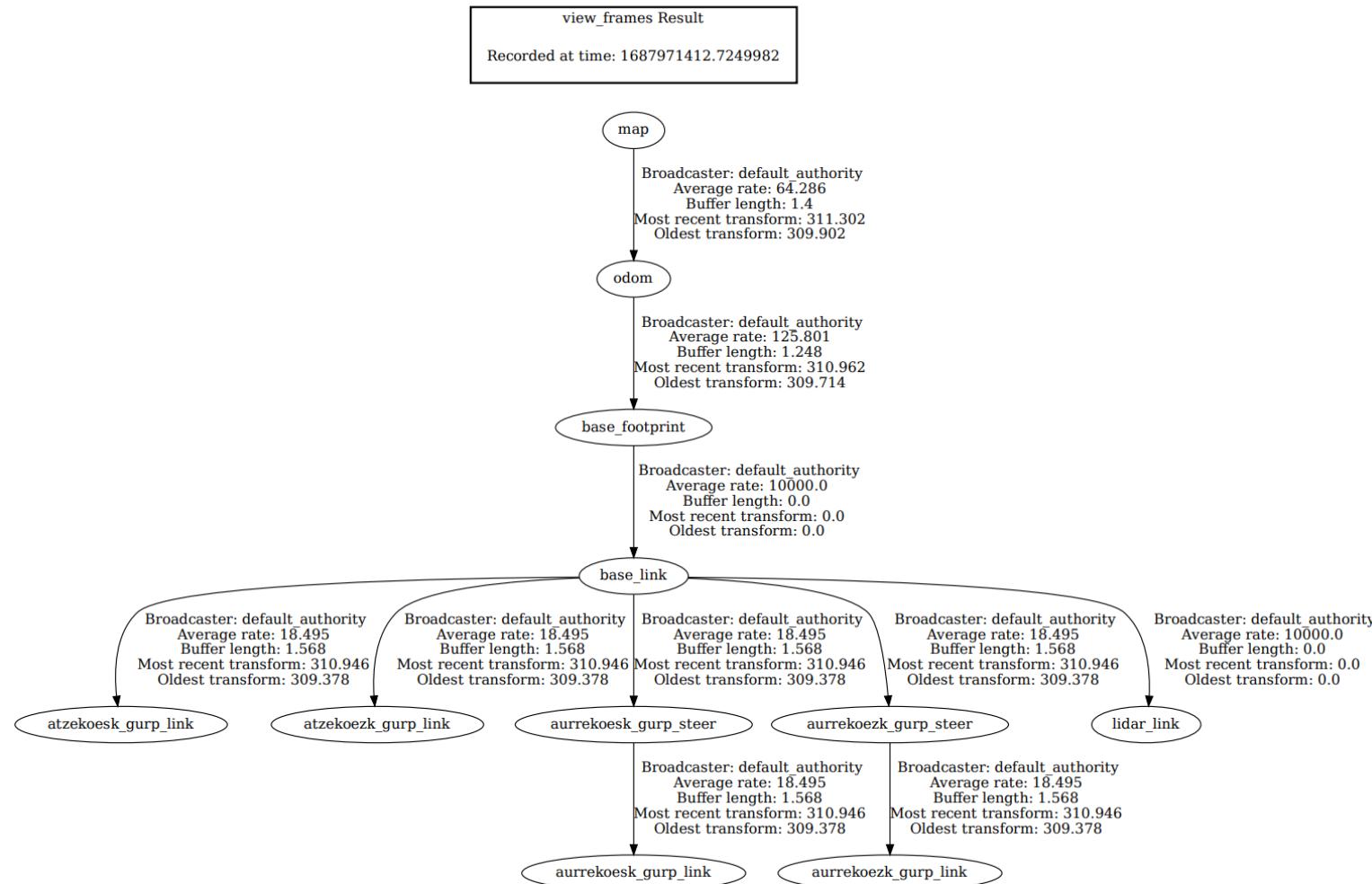
```
ros2 run tf2_tools view_frames
```

Komando hau exekutatu ostean, hurrengo orriko 20. irudian bezala, proiektuko transformatu guztien zuhaitz bat sortu beharko da. Egitura hurrengoa da:

Map → Odom → Base\_footprint → Base\_link → ...

Orden hau robotaren kokapena eta orientazioa kalkulatzeko erabilitako transformazioen sekuentzia islatzen du, maparen erreferentzia-esparru globalari dagokionez (Map).

Eraldaketa horiei esker, sistemaren osagaietako (sentsoreak, gurpilak, etab.) maparen erreferentzia-esparru orokorrean lan egin dezakete, eta elkarren artean komunikatu, harreman egokiak erabiliz. Hori funtsezkoa da nabigazio zehatza egiteko eta inguruneko ibilbideak planifikatzeko.



**20. Irudia:** transformatu zuhaitza.

## 6. BIBLIOGRAFIA

[1] The Construct plataforma:

<https://www.theconstructsim.com/>

[2] ROS 2 Humble Hawksbill bertsio ofiziala deskargatzeko:

<https://docs.ros.org/en/humble/Installation.html>

[3] Navigation 2 paketearen dokumentu ofiziala:

<https://docs.ros.org/en/humble/Installation.html>

[4] URDF fitxategiei buruzko dokumentazio ofiziala:

<https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/URDF-Main.html>

[5] Xacro fitxategiei buruzko dokumentazio ofiziala:

<https://docs.ros.org/en/humble/Tutorials/Intermediate/URDF/Using-Xacro-to-Clean-Up-a-URDF-File.html>

[6] RViz tresnari buruzko dokumentazio ofiziala:

<https://wiki.ros.org/rviz>

[7] Gazebo simulazio-inguruneari buruzko dokumentazio ofiziala:

<https://classic.gazebosim.org/tutorials>

## 7. Eranskina: Programazio-kodeak

### 7.1 deskribapena.xacro

```

1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro">
3
4 <!-- Robotaren balio konstanteak -->
5 <xacro:property name="oinarri_zabalera" value="0.31"/>
6 <xacro:property name="oinarri_luzera" value="0.42"/>
7 <xacro:property name="oinarri_altuera" value="0.18"/>
8
9 <xacro:property name="gurpil_erradioa" value="0.10"/>
10 <xacro:property name="gurpil_zabalera" value="0.04"/>
11 <xacro:property name="wheel_ygap" value="0.025"/>
12 <xacro:property name="wheel_zoff" value="0.05"/>
13 <xacro:property name="wheel_xoff" value="0.12"/>
14
15 <xacro:property name="gurpil_marruskadura" value="0.8"/>
16
17 <!-- Definitu proietatee inertzialak -->
18 <xacro:macro name="kutxa_inerzia" params="m w h d">
19   <inertial>
20     <origin xyz="0 0 0" rpy="${pi/2} 0 ${pi/2}"/>
21     <mass value="${m}"/>
22     <inertia ixx="${(m/12) * (h*h + d*d)}" ixy="0.0" ixz="0.0" iyy="${(m/12) * (w*w + d*d)}" iyz="0.0" izz="${(m/12) * (w*w + h*h)}"/>
23   </inertial>
24 </xacro:macro>
25
26 <xacro:macro name="zilindro_inerzia" params="m r h">
27   <inertial>
28     <origin xyz="0 0 0" rpy="${pi/2} 0 0" />
29     <mass value="${m}"/>
30     <inertia ixx="${(m/12) * (3*r*r + h*h)}" ixy = "0" ixz = "0" iyy="${(m/12) * (3*r*r + h*h)}" iyz = "0" izz="${(m/2) * (r*r)}"/>
31   </inertial>
32 </xacro:macro>
33
34 <!-- Robotaren basea -->
35 <link name="base_link">
36   <visual>
37     <geometry>
38       <box size="${oinarri_luzera} ${oinarri_zabalera} ${oinarri_altuera}"/>
39     </geometry>
40     <material name="Cyan">
41       <color rgba="0 1.0 1.0 1.0"/>
42     </material>
43   </visual>
44   <collision>
45     <geometry>
46       <box size="${oinarri_luzera} ${oinarri_zabalera} ${oinarri_altuera}"/>
47     </geometry>
48   </collision>
49
50   <xacro:kutxa_inerzia m="0.4" w="${oinarri_zabalera}" d="${oinarri_luzera}" h="${oinarri_altuera}"/>
51 </link>
52
53 <!-- Robotaren Footprinta (itzala) -->
54 <link name="base_footprint">
55
56   <joint name="base_joint" type="fixed">
57     <parent link="base_footprint"/>
58     <child link="base_link"/>
59     <origin xyz="0.0 0.0 ${gurpil_erradioa+wheel_zoff}" rpy="0 0 0"/>
60   </joint>
61
62 <!-- Atzeko gurpilak -->

```

```

63 <!-- Ez dira ackermanak -->
64 <xacro:macro name="Atzeko_gurpilak" params="aurrizkia x_norantza y_norantza">
65   <link name="${aurrizkia}_link">
66     <visual>
67       <origin xyz="0 0 0" rpy="${pi/2} 0 0"/>
68       <geometry>
69         <cylinder radius="${gurpil_erradioa}" length="${gurpil_zabalera}"/>
70       </geometry>
71       <material name="txuria">
72         <color rgba="1 1 1 1"/>
73       </material>
74     </visual>
75
76     <collision>
77       <origin xyz="0 0 0" rpy="${pi/2} 0 0"/>
78       <geometry>
79         <cylinder radius="${gurpil_erradioa}" length="${gurpil_zabalera}"/>
80       </geometry>
81     </collision>
82
83   <xacro:zilindro_inertzia m="0.1" r="${gurpil_erradioa}" h="${gurpil_zabalera}"/>
84 </link>
85   <joint name="${aurrizkia}_joint" type="continuous">
86     <parent link="base_link"/>
87     <child link="${aurrizkia}_link"/>
88     <origin xyz="${x_norantza*wheel_xoff} ${y_norantza*(oinarri_zabalera/2+wheel_ygap)} ${-wheel_zoff}" rpy="0 0 0"/>
89     <axis xyz="0 1 0"/>
90   </joint>
91 </xacro:macro>
92
93 <xacro:Atzeko_gurpilak aurrizkia="atzekoezk_gurp" x_norantza="-1" y_norantza="1" />
94 <xacro:Atzeko_gurpilak aurrizkia="atzekoesk_gurp" x_norantza="-1" y_norantza="-1" />
95
96 <!-- Aurreko gurpilak -->
97 <!-- Ackermann-ak-->
98 <xacro:macro name="Aurreko_gurpila" params="aurrizkia2 x_norantza2 y_norantza2">
99   <link name="${aurrizkia2}_link">
100    <visual>
101      <origin xyz="0 0 0" rpy="${pi/2} 0 0"/>
102      <geometry>
103        <cylinder radius="${gurpil_erradioa}" length="${gurpil_zabalera}"/>
104      </geometry>
105      <material name="txuria">
106        <color rgba="1 1 1 1"/>
107      </material>
108    </visual>
109
110    <collision>
111      <origin xyz="0 0 0" rpy="${pi/2} 0 0"/>
112      <geometry>
113        <cylinder radius="${gurpil_erradioa}" length="${gurpil_zabalera}"/>
114      </geometry>
115    </collision>
116
117   <xacro:zilindro_inertzia m="0.1" r="${gurpil_erradioa}" h="${gurpil_zabalera}"/>
118 </link>
119
120 </xacro:macro>
121
122 <xacro:Aurreko_gurpila aurrizkia2="aurrekoezk_gurp" x_norantza2="1" y_norantza2="1" />
123 <xacro:Aurreko_gurpila aurrizkia2="aurrekoesk_gurp" x_norantza2="1" y_norantza2="-1" />
124
125 <joint name="aurrekoezk_gurp_joint" type="continuous">
126   <parent link="aurrekoezk_gurp_steer"/>
127   <child link="aurrekoezk_gurp_link"/>
128   <origin xyz="0 0 0" rpy="0 0 0"/>
129   <axis xyz="0 1 0"/>
130 </joint>
131 <joint name="aurrekoesk_gurp_joint" type="continuous">
132   <parent link="aurrekoesk_gurp_steer"/>
```

```

133 <child link="urrekoesk_gurp_link"/>
134 <origin xyz="0 0 0" rpy="0 0 0"/>
135 <axis xyz="0 1 0"/>
136 </joint>
137
138 <link name="urrekoezk_gurp_steer">
139 <visual>
140   <geometry>
141     <cylinder length="0.01" radius="0.01"/>
142   </geometry>
143   <material name="black">
144     <color rgba="0 0 0 1"/>
145   </material>
146 </visual>
147 <collision>
148   <geometry>
149     <cylinder length="0.01" radius="0.01"/>
150   </geometry>
151 </collision>
152 <inertial>
153   <mass value="0.1"/>
154   <inertia ixx="5.1458e-5" iyy="5.1458e-5" izz="6.125e-5"
155     ixy="0" ixz="0" iyz="0"/>
156 </inertial>
157 </link>
158
159 <link name="urrekoesk_gurp_steer">
160 <visual>
161   <geometry>
162     <cylinder length="0.01" radius="0.01"/>
163   </geometry>
164   <material name="black">
165     <color rgba="0 0 0 1"/>
166   </material>
167 </visual>
168 <collision>
169   <geometry>
170     <cylinder length="0.01" radius="0.01"/>
171   </geometry>
172 </collision>
173 <inertial>
174   <mass value="0.1"/>
175   <inertia ixx="5.1458e-5" iyy="5.1458e-5" izz="6.125e-5"
176     ixy="0" ixz="0" iyz="0"/>
177 </inertial>
178 </link>
179
180 <joint name="urrekoezk_gurp_steer_joint" type="revolute">
181   <limit lower="-1" upper="1" effort="1" velocity="5"/>
182   <axis xyz="0 0 1"/>
183   <parent link="base_link"/>
184   <child link="urrekoezk_gurp_steer"/>
185   <origin xyz="${wheel_xoff} ${oinarri_zabalera/2+wheel_ygap} ${-wheel_zoff}" rpy="0 0 0"/>
186 </joint>
187
188 <joint name="urrekoesk_gurp_steer_joint" type="revolute">
189   <limit lower="-1" upper="1" effort="1" velocity="5"/>
190   <axis xyz="0 0 1"/>
191   <parent link="base_link"/>
192   <child link="urrekoesk_gurp_steer"/>
193   <origin xyz="${wheel_xoff} ${-1*(oinarri_zabalera/2+wheel_ygap)} ${-wheel_zoff}" rpy="0 0 0"/>
194 </joint>
195
196 </robot>
```

### 1. Programazio-kodea: deskribapena.xacro

## 7.2 marti\_bot.urdf.xacro

```
1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro" name="marti_bot">
3
4   <xacro:include filename="deskribapena.xacro" />
5
6 </robot>
```

2. Programazio-kodea: marti\_bot.urdf.xacro

## 7.3 urdf\_config.rviz

```
1 Panels:
2   - Class: rviz_common/Displays
3     Help Height: 78
4     Name: Displays
5     Property Tree Widget:
6       Expanded:
7         - /Global Options1
8         - /Status1
9         - /RobotModel1/Links1
10        - /TF1
11     Splitter Ratio: 0.5
12   Tree Height: 557
13 Visualization Manager:
14   Class: ""
15   Displays:
16     - Alpha: 0.5
17     Cell Size: 1
18     Class: rviz_default_plugins/Grid
19     Color: 160; 160; 164
20     Enabled: true
21     Name: Grid
22     - Alpha: 0.6
23     Class: rviz_default_plugins/RobotModel
24     Description Topic:
25       Depth: 5
26       Durability Policy: Volatile
27       History Policy: Keep Last
28       Reliability Policy: Reliable
29       Value: /robot_description
30     Enabled: true
31     Name: RobotModel
32     Visual Enabled: true
33     - Class: rviz_default_plugins/TF
34     Enabled: true
35     Name: TF
36     Marker Scale: 0.3
37     Show Arrows: true
38     Show Axes: true
39     Show Names: true
40   Enabled: true
41 Global Options:
42   Background Color: 48; 48; 48
43   Fixed Frame: base_link
44   Frame Rate: 30
45   Name: root
46 Tools:
47   - Class: rviz_default_plugins/Interact
48     Hide Inactive Objects: true
49   - Class: rviz_default_plugins/MoveCamera
50   - Class: rviz_default_plugins/Select
51   - Class: rviz_default_plugins/FocusCamera
52   - Class: rviz_default_plugins/Measure
53   Line color: 128; 128; 0
```

```

54 - Class: rviz_default_plugins/SetInitialPose
55 Topic:
56   Depth: 5
57   Durability Policy: Volatile
58   History Policy: Keep Last
59   Reliability Policy: Reliable
60   Value: /initialpose
61 - Class: rviz_default_plugins/PublishPoint
62   Single click: true
63   Topic:
64     Depth: 5
65     Durability Policy: Volatile
66     History Policy: Keep Last
67     Reliability Policy: Reliable
68     Value: /clicked_point
69 - Class: nav2_rviz_plugins/GoalTool
70 Transformation:
71   Current:
72     Class: rviz_default_plugins/TF
73   Value: true
74 Views:
75   Current:
76     Class: rviz_default_plugins/Orbit
77     Name: Current View
78     Target Frame: <Fixed Frame>
79     Value: Orbit (rviz)
80 Saved: ~
  
```

### 3. Programazio-kodea: urdf\_config.rviz

## 7.4 rviz\_robota.launch.py

```

1 import os
2 from launch import LaunchDescription
3 from launch.actions import DeclareLaunchArgument
4 from launch.conditions import IfCondition, UnlessCondition
5 from launch.substitutions import Command, LaunchConfiguration
6 from launch_ros.actions import Node
7 from launch_ros.substitutions import FindPackageShare
8
9 def generate_launch_description():
10
11   # Set the path to different files and folders.
12   pkg_share = FindPackageShare(package='marti_bot').find('marti_bot')
13   default_model_path = os.path.join(pkg_share, 'deskribapena/marti_bot.urdf.xacro')
14   default_rviz_config_path = os.path.join(pkg_share, 'rviz/urdf_config.rviz')
15
16   # Launch configuration variables specific to simulation
17   gui = LaunchConfiguration('gui')
18   model = LaunchConfiguration('model')
19   rviz_config_file = LaunchConfiguration('rviz_config_file')
20   use_robot_state_pub = LaunchConfiguration('use_robot_state_pub')
21   use_rviz = LaunchConfiguration('use_rviz')
22   use_sim_time = LaunchConfiguration('use_sim_time')
23
24   # Declare the launch arguments
25   declare_model_path_cmd = DeclareLaunchArgument(
26     name='model',
27     default_value=default_model_path,
28     description='Absolute path to robot urdf file')
29
30   declare_rviz_config_file_cmd = DeclareLaunchArgument(
31     name='rviz_config_file',
32     default_value=default_rviz_config_path,
33     description='Full path to the RVIZ config file to use')
34
35   declare_use_joint_state_publisher_cmd = DeclareLaunchArgument(
36     name='gui',
  
```

```

37 default_value='True',
38 description='Flag to enable joint_state_publisher_gui')
39
40 declare_use_robot_state_pub_cmd = DeclareLaunchArgument(
41   name='use_robot_state_pub',
42   default_value='True',
43   description='Whether to start the robot state publisher')
44
45 declare_use_rviz_cmd = DeclareLaunchArgument(
46   name='use_rviz',
47   default_value='True',
48   description='Whether to start RVIZ')
49
50 declare_use_sim_time_cmd = DeclareLaunchArgument(
51   name='use_sim_time',
52   default_value='True',
53   description='Use simulation (Gazebo) clock if true')
54 # Specify the actions
55 # Publish the joint state values for the non-fixed joints in the URDF file.
56 start_joint_state_publisher_cmd = Node(
57   condition=UnlessCondition(gui),
58   package='joint_state_publisher',
59   executable='joint_state_publisher',
60   name='joint_state_publisher')
61 # A GUI to manipulate the joint state values
62 start_joint_state_publisher_gui_node = Node(
63   condition=IfCondition(gui),
64   package='joint_state_publisher_gui',
65   executable='joint_state_publisher_gui',
66   name='joint_state_publisher_gui')
67 # Subscribe to the joint states of the robot, and publish the 3D pose of each link.
68 start_robot_state_publisher_cmd = Node(
69   condition=IfCondition(use_robot_state_pub),
70   package='robot_state_publisher',
71   executable='robot_state_publisher',
72   parameters=[{'use_sim_time': use_sim_time,
73 'robot_description': Command(['xacro ', model])}],
74   arguments=[default_model_path])
75 # Launch RViz
76 start_rviz_cmd = Node(
77   condition=IfCondition(use_rviz),
78   package='rviz2',
79   executable='rviz2',
80   name='rviz2',
81   output='screen',
82   arguments=['-d', rviz_config_file])
83
84 # Create the launch description and populate
85 Id = LaunchDescription()
86
87 # Declare the launch options
88 Id.add_action(declare_model_path_cmd)
89 Id.add_action(declare_rviz_config_file_cmd)
90 Id.add_action(declare_use_joint_state_publisher_cmd)
91 Id.add_action(declare_use_robot_state_pub_cmd)
92 Id.add_action(declare_use_rviz_cmd)
93 Id.add_action(declare_use_sim_time_cmd)
94
95 # Add any actions
96 Id.add_action(start_joint_state_publisher_cmd)
97 Id.add_action(start_joint_state_publisher_gui_node)
98 Id.add_action(start_robot_state_publisher_cmd)
99 Id.add_action(start_rviz_cmd)
100
101 return Id

```

#### 4. Programazio-kodea: rviz\_robota.launch.py

## 7.5 package.xml

```

1 <?xml version="1.0"?>
2 <?xml-model href="http://download.ros.org/schema/package_format3.xsd"
3   schematypens="http://www.w3.org/2001/XMLSchema"?>
4 <package format="3">
5   <name>marti_bot</name>
6   <version>0.0.0</version>
7   <description>TODO: Package description</description>
8   <maintainer email="user@todo.todo">user</maintainer>
9   <license>TODO: License declaration</license>
10
11  <buildtool_depend>ament_cmake</buildtool_depend>
12
13  <exec_depend>joint_state_publisher</exec_depend>
14  <exec_depend>robot_state_publisher</exec_depend>
15  <exec_depend>rviz2</exec_depend>
16  <exec_depend>xacro</exec_depend>
17
18  <test_depend>ament_lint_auto</test_depend>
19  <test_depend>ament_lint_common</test_depend>
20
21  <export>
22    <build_type>ament_cmake</build_type>
23  </export>
</package>

```

**5. Programazio-kodea:** package.xml

## 7.6 CMakeLists.txt

```

1 cmake_minimum_required(VERSION 3.8)
2 project(marti_bot)
3
4 if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
5   add_compile_options(-Wall -Wextra -Wpedantic)
6 endif()
7 # find dependencies
8 find_package(ament_cmake REQUIRED)
9 # uncomment the following section in order to fill in
10 # further dependencies manually.
11 #find_package(<dependency> REQUIRED)
12
13 install(
14   DIRECTORY src launch rviz deskribapena
15   DESTINATION share/${PROJECT_NAME}
16 )
17
18 if(BUILD_TESTING)
19   find_package(ament_lint_auto REQUIRED)
20   # the following line skips the linter which checks for copyrights
21   # comment the line when a copyright and license is added to all source files
22   set(ament_cmake_copyright_FOUND TRUE)
23   # the following line skips cpplint (only works in a git repo)
24   # comment the line when this package is in a git repo and when
25   # a copyright and license is added to all source files
26   set(ament_cmake_cpplint_FOUND TRUE)
27   ament_lint_auto_find_test_dependencies()
28 endif()
29
30 ament_package()

```

**6. Programazio-kodea:** CMakeLists.txt

## 7.7 gazebo\_control.xacro

```
1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro">
3
4   <gazebo reference="aurrekoezk_gurp_link">
5     <mu1>${gurpil_marruskadura}</mu1>
6     <material>Gazebo/Purple</material>
7     <mu1>0.9</mu1>
8     <mu2>0.9</mu2>
9     <minDepth>0.001</minDepth>
10    <kp>100000000</kp>
11    <kd>1</kd>
12    <implicitSpringDamper>true</implicitSpringDamper>
13  </gazebo>
14  <gazebo reference="aurrekoesk_gurp_link">
15
16    <mu1>${gurpil_marruskadura}</mu1>
17    <material>Gazebo/Purple</material>
18    <mu1>0.9</mu1>
19    <mu2>0.9</mu2>
20    <minDepth>0.001</minDepth>
21    <kp>100000000</kp>
22    <kd>1</kd>
23    <implicitSpringDamper>true</implicitSpringDamper>
24  </gazebo>
25  <gazebo reference="atzekoezk_gurp_link">
26    <mu1>${gurpil_marruskadura}</mu1>
27    <material>Gazebo/DarkGrey</material>
28    <mu1>0.9</mu1>
29    <mu2>0.9</mu2>
30    <minDepth>0.001</minDepth>
31    <kp>100000000</kp>
32    <kd>1</kd>
33    <implicitSpringDamper>true</implicitSpringDamper>
34  </gazebo>
35  <gazebo reference="atzekoesk_gurp_link">
36    <mu1>${gurpil_marruskadura}</mu1>
37    <material>Gazebo/DarkGrey</material>
38    <mu1>0.9</mu1>
39    <mu2>0.9</mu2>
40    <minDepth>0.001</minDepth>
41    <kp>100000000</kp>
42    <kd>1</kd>
43    <implicitSpringDamper>true</implicitSpringDamper>
44  </gazebo>
45
46  <gazebo reference="base_link">
47    <dampingFactor>0.00007</dampingFactor>
48  </gazebo>
49  <gazebo>
50    <plugin name="marti_gazebo_plugin" filename="libmarti_gazebo_plugin.so">
51      </plugin>
52  </gazebo>
53</robot>
```

7. Programazio-kodea: gazebo\_control.xacro

## 7.8 etxea.world

```

1 <sdf version='1.6'>
2   <world name='default'>
3     <light name='sun' type='directional'>
4       <cast_shadows>1</cast_shadows>
5       <pose frame="">0 0 10 0 -0 0</pose>
6       <diffuse>0.8 0.8 0.8 1</diffuse>
7       <specular>0.2 0.2 0.2 1</specular>
8       <attenuation>
9         <range>1000</range>
10        <constant>0.9</constant>
11        <linear>0.01</linear>
12        <quadratic>0.001</quadratic>
13      </attenuation>
14      <direction>-0.5 0.1 -0.9</direction>
15    </light>
16    <model name='ground_plane'>
17      <static>1</static>
18      <link name='link'>
19        <collision name='collision'>
20          <geometry>
21            <plane>
22              <normal>0 0 1</normal>
23              <size>100 100</size>
24            </plane>
25          </geometry>
26          <surface>
27            <contact>
28              <collide_bitmask>65535</collide_bitmask>
29              <ode/>
30            </contact>
31            <friction>
32              <ode>
33                <mu>100</mu>
34                <mu2>50</mu2>
35              </ode>
36              <torsional>
37                <ode/>
38              </torsional>
39            </friction>
40            <bounce/>
41          </surface>
42          <max_contacts>10</max_contacts>
43        </collision>
44        <visual name='visual'>
45          <cast_shadows>0</cast_shadows>
46          <geometry>
47            <plane>
48              <normal>0 0 1</normal>
49              <size>100 100</size>
50            </plane>
51          </geometry>
52          <material>
53            <script>
54              <uri>file:///media/materials/scripts/gazebo.material</uri>
55              <name>Gazebo/Grey</name>
56            </script>
57          </material>
58        </visual>
59        <self_collide>0</self_collide>
60        <enable_wind>0</enable_wind>
61        <kinematic>0</kinematic>

```

```

62    </link>
63    </model>
64    <gravity>0 0 -9.8</gravity>
65    <magnetic_field>6e-06 2.3e-05 -4.2e-05</magnetic_field>
66    <atmosphere type='adiabatic' />
67    <physics name='default_physics' default='0' type='ode'>
68        <max_step_size>0.001</max_step_size>
69        <real_time_factor>1</real_time_factor>
70        <real_time_update_rate>1000</real_time_update_rate>
71    </physics>
72    <scene>
73        <ambient>0.4 0.4 0.4 1</ambient>
74        <background>0.7 0.7 0.7 1</background>
75        <shadows>1</shadows>
76    </scene>
77    <audio>
78        <device>default</device>
79    </audio>
80    <wind/>
81    <spherical_coordinates>
82        <surface_model>EARTH_WGS84</surface_model>
83        <latitude_deg>0</latitude_deg>
84        <longitude_deg>0</longitude_deg>
85        <elevation>0</elevation>
86        <heading_deg>0</heading_deg>
87    </spherical_coordinates>
88    <model name='sudhir.building'>
89        <pose frame="">-1.27854 7.44696 0 0 -0 0</pose>
90        <link name='Wall_0'>
91            <collision name='Wall_0_Collision'>
92                <geometry>
93                    <box>
94                        <size>4 0.15 2.5</size>
95                    </box>
96                </geometry>
97                <pose frame="">0 0 1.25 0 -0 0</pose>
98                <max_contacts>10</max_contacts>
99                <surface>
100                    <contact>
101                        <ode/>
102                    </contact>
103                    <bounce/>
104                    <friction>
105                        <torsional>
106                            <ode/>
107                        </torsional>
108                    <ode/>
109                </friction>
110            </surface>
111        </collision>
112        <visual name='Wall_0_Visual'>
113            <pose frame="">0 0 1.25 0 -0 0</pose>
114            <geometry>
115                <box>
116                    <size>4 0.15 2.5</size>
117                </box>
118            </geometry>
119            <meta>
120                <layer>0</layer>
121            </meta>
122        </visual>
123        <pose frame="">-7.92943 1.85 0 0 -0 -1.5708</pose>
124        <self_collide>0</self_collide>

```

```

125      <enable_wind>0</enable_wind>
126      <kinematic>0</kinematic>
127    </link>
128
129    <link name='Wall_11'>
130      <collision name='Wall_11_Collision'>
131        <geometry>
132          <box>
133            <size>2.5 0.15 2.5</size>
134          </box>
135        </geometry>
136        <pose frame="">0 0 1.25 0 -0 0</pose>
137        <max_contacts>10</max_contacts>
138      <surface>
139        <contact>
140          <ode/>
141        </contact>
142        <bounce/>
143        <friction>
144          <torsional>
145            <ode/>
146          </torsional>
147          <ode/>
148        </friction>
149      </surface>
150    </collision>
151    <visual name='Wall_11_Visual'>
152      <pose frame="">0 0 1.25 0 -0 0</pose>
153      <geometry>
154        <box>
155          <size>2.5 0.15 2.5</size>
156        </box>
157      </geometry>
158      <meta>
159        <layer>0</layer>
160      </meta>
161    </visual>
162    <pose frame="">-2.06679 -1.31994 0 0 -0 3.14159</pose>
163    <self_collide>0</self_collide>
164    <enable_wind>0</enable_wind>
165    <kinematic>0</kinematic>
166  </link>
167  <link name='Wall_13'>
168    <collision name='Wall_13_Collision'>
169      <geometry>
170        <box>
171          <size>1.75 0.15 2.5</size>
172        </box>
173      </geometry>
174      <pose frame="">0 0 1.25 0 -0 0</pose>
175      <max_contacts>10</max_contacts>
176    <surface>
177      <contact>
178        <ode/>
179      </contact>
180      <bounce/>
181      <friction>
182        <torsional>
183          <ode/>
184        </torsional>
185        <ode/>
186      </friction>
187    </surface>
```

```
188 </collision>
189 <visual name='Wall_13_Visual'>
190 <pose frame="">0 0 1.25 0 -0 0</pose>
191 <geometry>
192   <box>
193     <size>1.75 0.15 2.5</size>
194   </box>
195 </geometry>
196
197 <meta>
198   <layer>0</layer>
199 </meta>
200 </visual>
201 <pose frame="">-0.908613 2.9996 0 0 -0 -1.5708</pose>
202 <self_collide>0</self_collide>
203 <enable_wind>0</enable_wind>
204 <kinematic>0</kinematic>
205 </link>
206 <link name='Wall_14'>
207   <collision name='Wall_14_Collision'>
208     <geometry>
209       <box>
210         <size>2.75 0.15 2.5</size>
211       </box>
212     </geometry>
213     <pose frame="">0 0 1.25 0 -0 0</pose>
214     <max_contacts>10</max_contacts>
215     <surface>
216       <contact>
217         <ode/>
218       </contact>
219       <bounce/>
220       <friction>
221         <torsional>
222           <ode/>
223         </torsional>
224           <ode/>
225         </friction>
226       </surface>
227     </collision>
228     <visual name='Wall_14_Visual'>
229       <pose frame="">0 0 1.25 0 -0 0</pose>
230       <geometry>
231         <box>
232           <size>2.75 0.15 2.5</size>
233         </box>
234       </geometry>
235
236     <meta>
237       <layer>0</layer>
238     </meta>
239   </visual>
240   <pose frame="">0.391387 2.19961 0 0 -0 0</pose>
241   <self_collide>0</self_collide>
242   <enable_wind>0</enable_wind>
243   <kinematic>0</kinematic>
244 </link>
245 <link name='Wall_17'>
246   <collision name='Wall_17_Collision'>
247     <geometry>
248       <box>
249         <size>1.75 0.15 2.5</size>
250       </box>
```

```

251   </geometry>
252   <pose frame="">0 0 1.25 0 -0 0</pose>
253   <max_contacts>10</max_contacts>
254   <surface>
255     <contact>
256       <ode/>
257     </contact>
258     <bounce/>
259     <friction>
260       <torsional>
261         <ode/>
262       </torsional>
263         <ode/>
264     </friction>
265   </surface>
266 </collision>
267 <visual name='Wall_17_Visual'>
268   <pose frame="">0 0 1.25 0 -0 0</pose>
269   <geometry>
270     <box>
271       <size>1.75 0.15 2.5</size>
272     </box>
273   </geometry>
274   <meta>
275     <layer>0</layer>
276   </meta>
277 </visual>
278 <pose frame="">2.63138 3.01972 0 0 -0 -1.5708</pose>
279 <self_collide>0</self_collide>
280 <enable_wind>0</enable_wind>
281 <kinematic>0</kinematic>
282 </link>
283 <link name='Wall_18'>
284   <collision name='Wall_18_Collision'>
285     <geometry>
286       <box>
287         <size>2.5 0.15 2.5</size>
288       </box>
289     </geometry>
290     <pose frame="">0 0 1.25 0 -0 0</pose>
291     <max_contacts>10</max_contacts>
292     <surface>
293       <contact>
294         <ode/>
295       </contact>
296       <bounce/>
297       <friction>
298         <torsional>
299           <ode/>
300         </torsional>
301         <ode/>
302       </friction>
303     </surface>
304   </collision>
305   <visual name='Wall_18_Visual'>
306     <pose frame="">0 0 1.25 0 -0 0</pose>
307     <geometry>
308       <box>
309         <size>2.5 0.15 2.5</size>
310       </box>
311     </geometry>
312     <meta>
313       <layer>0</layer>

```

```

314      </meta>
315      </visual>
316      <pose frame="">3.80638 2.21972 0 0 -0 0</pose>
317      <self_collide>0</self_collide>
318      <enable_wind>0</enable_wind>
319      <kinematic>0</kinematic>
320      </link>
321      <link name="Wall_20">
322          <collision name="Wall_20_Collision">
323              <geometry>
324                  <box>
325                      <size>2.5 0.15 2.5</size>
326                  </box>
327              </geometry>
328              <pose frame="">0 0 1.25 0 -0 0</pose>
329              <max_contacts>10</max_contacts>
330              <surface>
331                  <contact>
332                      <ode/>
333                  </contact>
334                  <bounce/>
335                  <friction>
336                      <torsional>
337                          <ode/>
338                      </torsional>
339                          <ode/>
340                      </friction>
341                  </surface>
342              </collision>
343              <visual name="Wall_20_Visual">
344                  <pose frame="">0 0 1.25 0 -0 0</pose>
345                  <geometry>
346                      <box>
347                          <size>2.5 0.15 2.5</size>
348                      </box>
349                  </geometry>
350                  <meta>
351                      <layer>0</layer>
352                  </meta>
353              </visual>
354              <pose frame="">5.74898 2.68495 0 0 -0 -1.5708</pose>
355              <self_collide>0</self_collide>
356              <enable_wind>0</enable_wind>
357              <kinematic>0</kinematic>
358          </link>
359          <link name="Wall_22">
360              <collision name="Wall_22_Collision">
361                  <geometry>
362                      <box>
363                          <size>2.25 0.15 2.5</size>
364                      </box>
365                  </geometry>
366                  <pose frame="">0 0 1.25 0 -0 0</pose>
367                  <max_contacts>10</max_contacts>
368                  <surface>
369                      <contact>
370                          <ode/>
371                      </contact>
372                      <bounce/>
373                      <friction>
374                          <torsional>
375                              <ode/>
376                          </torsional>
```

```

377      <ode/>
378      </friction>
379      </surface>
380  </collision>
381  <visual name="Wall_22_Visual">
382    <pose frame="">0 0 1.25 0 -0 0</pose>
383    <geometry>
384      <box>
385        <size>2.25 0.15 2.5</size>
386      </box>
387    </geometry>
388    <meta>
389      <layer>0</layer>
390    </meta>
391  </visual>
392  <pose frame="">6.81909 0.722229 0 0 -0 0</pose>
393  <self_collide>0</self_collide>
394  <enable_wind>0</enable_wind>
395  <kinematic>0</kinematic>
396 </link>
397 <link name='Wall_24'>
398   <collision name='Wall_24_Collision'>
399     <geometry>
400       <box>
401         <size>1.75 0.15 2.5</size>
402       </box>
403     </geometry>
404     <pose frame="">0 0 1.25 0 -0 0</pose>
405     <max_contacts>10</max_contacts>
406   <surface>
407     <contact>
408       <ode/>
409     </contact>
410     <bounce/>
411     <friction>
412       <torsional>
413         <ode/>
414       </torsional>
415         <ode/>
416       </friction>
417     </surface>
418   </collision>
419   <visual name="Wall_24_Visual">
420     <pose frame="">0 0 1.25 0 -0 0</pose>
421     <geometry>
422       <box>
423         <size>1.75 0.15 2.5</size>
424       </box>
425     </geometry>
426     <meta>
427       <layer>0</layer>
428     </meta>
429   </visual>
430   <pose frame="">5.82943 -0.902428 0 0 -0 -1.5708</pose>
431   <self_collide>0</self_collide>
432   <enable_wind>0</enable_wind>
433   <kinematic>0</kinematic>
434 </link>
435 <link name='Wall_25'>
436   <collision name='Wall_25_Collision'>
437     <geometry>
438       <box>
439         <size>2.25 0.15 2.5</size>

```

```

440      </box>
441      </geometry>
442      <pose frame="">0 0 1.25 0 -0 0</pose>
443      <max_contacts>10</max_contacts>
444      <surface>
445          <contact>
446              <ode/>
447          </contact>
448          <bounce/>
449          <friction>
450              <torsional>
451                  <ode/>
452          </torsional>
453          <ode/>
454      </friction>
455      </surface>
456  </collision>
457  <visual name="Wall_25_Visual">
458      <pose frame="">0 0 1.25 0 -0 0</pose>
459      <geometry>
460          <box>
461              <size>2.25 0.15 2.5</size>
462          </box>
463      </geometry>
464      <meta>
465          <layer>0</layer>
466      </meta>
467  </visual>
468  <pose frame="">6.87943 -1.70243 0 0 -0 0</pose>
469  <self_collide>0</self_collide>
470  <enable_wind>0</enable_wind>
471  <kinematic>0</kinematic>
472 </link>
473 <link name="Wall_3">
474     <collision name="Wall_3_Collision">
475         <geometry>
476             <box>
477                 <size>4 0.15 2.5</size>
478             </box>
479         </geometry>
480         <pose frame="">0 0 1.25 0 -0 0</pose>
481         <max_contacts>10</max_contacts>
482         <surface>
483             <contact>
484                 <ode/>
485             </contact>
486             <bounce/>
487             <friction>
488                 <torsional>
489                     <ode/>
490                 </torsional>
491                 <ode/>
492             </friction>
493         </surface>
494     </collision>
495     <visual name="Wall_3_Visual">
496         <pose frame="">0 0 1.25 0 -0 0</pose>
497         <geometry>
498             <box>
499                 <size>4 0.15 2.5</size>
500             </box>
501         </geometry>
502         <meta>
```

```

503     <layer>0</layer>
504     </meta>
505   </visual>
506   <pose frame="-7.92943 -2 0 0 -0 -1.5708"></pose>
507   <self_collide>0</self_collide>
508   <enable_wind>0</enable_wind>
509   <kinematic>0</kinematic>
510 </link>
511 <link name='Wall_4'>
512   <collision name='Wall_4_Collision'>
513     <geometry>
514       <box>
515         <size>16 0.15 2.5</size>
516       </box>
517     </geometry>
518   <pose frame="0 0 1.25 0 -0 0"></pose>
519   <max_contacts>10</max_contacts>
520   <surface>
521     <contact>
522       <ode/>
523     </contact>
524     <bounce/>
525     <friction>
526       <torsional>
527         <ode/>
528       </torsional>
529         <ode/>
530       </friction>
531     </surface>
532   </collision>
533   <visual name='Wall_4_Visual'>
534     <pose frame="0 0 1.25 0 -0 0"></pose>
535     <geometry>
536       <box>
537         <size>16 0.15 2.5</size>
538       </box>
539     </geometry>
540     <meta>
541       <layer>0</layer>
542     </meta>
543   </visual>
544   <pose frame="-0.004434 -3.925 0 0 -0 0"></pose>
545   <self_collide>0</self_collide>
546   <enable_wind>0</enable_wind>
547   <kinematic>0</kinematic>
548 </link>
549 <link name='Wall_5'>
550   <pose frame="7.92057 0 0 0 -0 1.5708"></pose>
551   <visual name='Wall_5_Visual_0'>
552     <pose frame="-3.92553 0 1.25 0 -0 0"></pose>
553     <geometry>
554       <box>
555         <size>0.148947 0.15 2.5</size>
556       </box>
557     </geometry>
558     <meta>
559       <layer>0</layer>
560     </meta>
561   </visual>
562   <collision name='Wall_5_Collision_0'>
563     <geometry>
564       <box>
565         <size>0.148947 0.15 2.5</size>

```

```
566    </box>
567    </geometry>
568    <pose frame="-3.92553 0 1.25 0 -0 0</pose>
569    <max_contacts>10</max_contacts>
570    <surface>
571        <contact>
572            <ode/>
573        </contact>
574        <bounce/>
575        <friction>
576            <torsional>
577                <ode/>
578            </torsional>
579        <ode/>
580    </friction>
581    </surface>
582  </collision>
583  <visual name="Wall_5_Visual_1">
584      <pose frame="0.524473 0 1.25 0 -0 0</pose>
585      <geometry>
586          <box>
587              <size>6.95105 0.15 2.5</size>
588          </box>
589      </geometry>
590      <meta>
591          <layer>0</layer>
592      </meta>
593  </visual>
594  <collision name="Wall_5_Collision_1">
595      <geometry>
596          <box>
597              <size>6.95105 0.15 2.5</size>
598          </box>
599      </geometry>
600      <pose frame="0.524473 0 1.25 0 -0 0</pose>
601      <max_contacts>10</max_contacts>
602      <surface>
603          <contact>
604              <ode/>
605          </contact>
606          <bounce/>
607          <friction>
608              <torsional>
609                  <ode/>
610              </torsional>
611          <ode/>
612      </friction>
613      </surface>
614  </collision>
615  <visual name="Wall_5_Visual_2">
616      <pose frame="-3.40105 0 2.25 0 -0 0</pose>
617      <geometry>
618          <box>
619              <size>0.9 0.15 0.5</size>
620          </box>
621      </geometry>
622      <meta>
623          <layer>0</layer>
624      </meta>
625  </visual>
626  <collision name="Wall_5_Collision_2">
627      <geometry>
628          <box>
```

```

629      <size>0.9 0.15 0.5</size>
630      </box>
631      </geometry>
632      <pose frame="">-3.40105 0 2.25 0 -0 0</pose>
633      <max_contacts>10</max_contacts>
634      <surface>
635          <contact>
636              <ode/>
637          </contact>
638          <bounce/>
639          <friction>
640              <torsional>
641                  <ode/>
642              </torsional>
643                  <ode/>
644          </friction>
645      </surface>
646      </collision>
647      <self_collide>0</self_collide>
648      <enable_wind>0</enable_wind>
649      <kinematic>0</kinematic>
650      </link>
651      <link name='Wall_6'>
652          <pose frame="">-0.004434 3.925 0 0 -0 3.14159</pose>
653          <visual name='Wall_6_Visual_0'>
654              <pose frame="">-0.655307 0 1.25 0 -0 0</pose>
655              <geometry>
656                  <box>
657                      <size>14.6894 0.15 2.5</size>
658                  </box>
659              </geometry>
660              <meta>
661                  <layer>0</layer>
662              </meta>
663          </visual>
664          <collision name='Wall_6_Collision_0'>
665              <geometry>
666                  <box>
667                      <size>14.6894 0.15 2.5</size>
668                  </box>
669              </geometry>
670              <pose frame="">-0.655307 0 1.25 0 -0 0</pose>
671              <max_contacts>10</max_contacts>
672              <surface>
673                  <contact>
674                      <ode/>
675                  </contact>
676                  <bounce/>
677                  <friction>
678                      <torsional>
679                          <ode/>
680                      </torsional>
681                          <ode/>
682                      </friction>
683                  </surface>
684          </collision>
685          <visual name='Wall_6_Visual_1'>
686              <pose frame="">7.79469 0 1.25 0 -0 0</pose>
687              <geometry>
688                  <box>
689                      <size>0.410614 0.15 2.5</size>
690                  </box>
691              </geometry>
```

```
692      <meta>
693          <layer>0</layer>
694      </meta>
695  </visual>
696  <collision name='Wall_6_Collision_1'>
697      <geometry>
698          <box>
699              <size>0.410614 0.15 2.5</size>
700          </box>
701      </geometry>
702      <pose frame="">7.79469 0 1.25 0 -0 0</pose>
703      <max_contacts>10</max_contacts>
704      <surface>
705          <contact>
706              <ode/>
707          </contact>
708          <bounce/>
709          <friction>
710              <torsional>
711                  <ode/>
712              </torsional>
713          <ode/>
714          </friction>
715      </surface>
716  </collision>
717  <visual name='Wall_6_Visual_2'>
718      <pose frame="">7.13939 0 2.25 0 -0 0</pose>
719      <geometry>
720          <box>
721              <size>0.9 0.15 0.5</size>
722          </box>
723      </geometry>
724      <meta>
725          <layer>0</layer>
726      </meta>
727  </visual>
728  <collision name='Wall_6_Collision_2'>
729      <geometry>
730          <box>
731              <size>0.9 0.15 0.5</size>
732          </box>
733      </geometry>
734      <pose frame="">7.13939 0 2.25 0 -0 0</pose>
735      <max_contacts>10</max_contacts>
736      <surface>
737          <contact>
738              <ode/>
739          </contact>
740          <bounce/>
741          <friction>
742              <torsional>
743                  <ode/>
744              </torsional>
745          <ode/>
746          </friction>
747      </surface>
748  </collision>
749  <self_collide>0</self_collide>
750  <enable_wind>0</enable_wind>
751  <kinematic>0</kinematic>
752  </link>
753  <link name='Wall_8'>
754      <collision name='Wall_8_Collision'>
```

```

755 <geometry>
756   <box>
757     <size>3 0.15 2.5</size>
758   </box>
759 </geometry>
760 <pose frame="">0 0 1.25 0 -0 0</pose>
761 <max_contacts>10</max_contacts>
762 <surface>
763   <contact>
764     <ode/>
765   </contact>
766   <bounce/>
767   <friction>
768     <torsional>
769       <ode/>
770     </torsional>
771     <ode/>
772   </friction>
773 </surface>
774 </collision>
775 <visual name="Wall_8_Visual">
776   <pose frame="">0 0 1.25 0 -0 0</pose>
777   <geometry>
778     <box>
779       <size>3 0.15 2.5</size>
780     </box>
781   </geometry>
782   <meta>
783     <layer>0</layer>
784   </meta>
785 </visual>
786 <pose frame="">-3.24179 2.45506 0 0 -0 -1.5708</pose>
787 <self_collide>0</self_collide>
788 <enable_wind>0</enable_wind>
789 <kinematic>0</kinematic>
790 </link>
791 <link name="Wall_9">
792   <collision name="Wall_9_Collision">
793     <geometry>
794       <box>
795         <size>2.5 0.15 2.5</size>
796       </box>
797     </geometry>
798     <pose frame="">0 0 1.25 0 -0 0</pose>
799     <max_contacts>10</max_contacts>
800   <surface>
801     <contact>
802       <ode/>
803     </contact>
804     <bounce/>
805     <friction>
806       <torsional>
807         <ode/>
808       </torsional>
809       <ode/>
810     </friction>
811   </surface>
812 </collision>
813 <visual name="Wall_9_Visual">
814   <pose frame="">0 0 1.25 0 -0 0</pose>
815   <geometry>
816     <box>
817       <size>2.5 0.15 2.5</size>

```

```

818      </box>
819      </geometry>
820      <meta>
821          <layer>0</layer>
822      </meta>
823      </visual>
824      <pose frame=">-2.06679 1.03006 0 0 -0 0</pose>
825      <self_collide>0</self_collide>
826      <enable_wind>0</enable_wind>
827      <kinematic>0</kinematic>
828      </link>
829      <static>1</static>
830      </model>
831      <state world_name='default'>
832          <sim_time>232 785000000</sim_time>
833          <real_time>118 702538223</real_time>
834          <wall_time>1551031600 831011974</wall_time>
835          <iterations>118381</iterations>
836          <model name='ground_plane'>
837              <pose frame=">0 0 0 0 -0 0</pose>
838              <scale>1 1 1</scale>
839              <link name='link'>
840                  <pose frame=">0 0 0 0 -0 0</pose>
841                  <velocity>0 0 0 0 -0 0</velocity>
842                  <acceleration>0 0 0 0 -0 0</acceleration>
843                  <wrench>0 0 0 0 -0 0</wrench>
844              </link>
845          </model>
846          <model name='sudhir.building'>
847              <pose frame=">0.673102 -0.134423 0 0 -0 0</pose>
848              <scale>1 1 1</scale>
849              <link name='Wall_0'>
850                  <pose frame=">-7.25633 1.71558 0 0 0 -1.5708</pose>
851                  <velocity>0 0 0 0 -0 0</velocity>
852                  <acceleration>0 0 0 0 -0 0</acceleration>
853                  <wrench>0 0 0 0 -0 0</wrench>
854              </link>
855              <link name='Wall_10'>
856                  <pose frame=">-0.218688 -0.279363 0 0 0 -1.5708</pose>
857                  <velocity>0 0 0 0 -0 0</velocity>
858                  <acceleration>0 0 0 0 -0 0</acceleration>
859                  <wrench>0 0 0 0 -0 0</wrench>
860              </link>
861              <link name='Wall_11'>
862                  <pose frame=">-1.39369 -1.45436 0 0 -0 3.14159</pose>
863                  <velocity>0 0 0 0 -0 0</velocity>
864                  <acceleration>0 0 0 0 -0 0</acceleration>
865                  <wrench>0 0 0 0 -0 0</wrench>
866              </link>
867              <link name='Wall_13'>
868                  <pose frame=">-0.235508 2.86522 0 0 0 -1.5708</pose>
869                  <velocity>0 0 0 0 -0 0</velocity>
870                  <acceleration>0 0 0 0 -0 0</acceleration>
871                  <wrench>0 0 0 0 -0 0</wrench>
872              </link>
873              <link name='Wall_14'>
874                  <pose frame=">1.06449 2.06519 0 0 -0 0</pose>
875                  <velocity>0 0 0 0 -0 0</velocity>
876                  <acceleration>0 0 0 0 -0 0</acceleration>
877                  <wrench>0 0 0 0 -0 0</wrench>
878              </link>
879              <link name='Wall_17'>
880                  <pose frame=">3.30448 2.88532 0 0 0 -1.5708</pose>

```

```

881 <velocity>0 0 0 0 -0 0</velocity>
882 <acceleration>0 0 0 0 -0 0</acceleration>
883 <wrench>0 0 0 0 -0 0</wrench>
884 </link>
885 <link name="Wall_18">
886   <pose frame="">4.47948 2.0853 0 0 -0 0</pose>
887   <velocity>0 0 0 0 -0 0</velocity>
888   <acceleration>0 0 0 0 -0 0</acceleration>
889   <wrench>0 0 0 0 -0 0</wrench>
890 </link>
891 <link name="Wall_20">
892   <pose frame="">6.42208 2.55052 0 0 0 -1.5708</pose>
893   <velocity>0 0 0 0 -0 0</velocity>
894   <acceleration>0 0 0 0 -0 0</acceleration>
895   <wrench>0 0 0 0 -0 0</wrench>
896 </link>
897 <link name="Wall_22">
898   <pose frame="">7.49219 0.587807 0 0 -0 0</pose>
899   <velocity>0 0 0 0 -0 0</velocity>
900   <acceleration>0 0 0 0 -0 0</acceleration>
901   <wrench>0 0 0 0 -0 0</wrench>
902 </link>
903 <link name="Wall_24">
904   <pose frame="">6.50253 -1.03685 0 0 0 -1.5708</pose>
905   <velocity>0 0 0 0 -0 0</velocity>
906   <acceleration>0 0 0 0 -0 0</acceleration>
907   <wrench>0 0 0 0 -0 0</wrench>
908 </link>
909 <link name="Wall_25">
910   <pose frame="">7.55253 -1.83685 0 0 -0 0</pose>
911   <velocity>0 0 0 0 -0 0</velocity>
912   <acceleration>0 0 0 0 -0 0</acceleration>
913   <wrench>0 0 0 0 -0 0</wrench>
914 </link>
915 <link name="Wall_3">
916   <pose frame="">-7.25633 -2.13442 0 0 0 -1.5708</pose>
917   <velocity>0 0 0 0 -0 0</velocity>
918   <acceleration>0 0 0 0 -0 0</acceleration>
919   <wrench>0 0 0 0 -0 0</wrench>
920 </link>
921 <link name="Wall_4">
922   <pose frame="">0.668672 -4.05942 0 0 -0 0</pose>
923   <velocity>0 0 0 0 -0 0</velocity>
924   <acceleration>0 0 0 0 -0 0</acceleration>
925   <wrench>0 0 0 0 -0 0</wrench>
926 </link>
927 <link name="Wall_5">
928   <pose frame="">8.59367 -0.134423 0 0 -0 1.5708</pose>
929   <velocity>0 0 0 0 -0 0</velocity>
930   <acceleration>0 0 0 0 -0 0</acceleration>
931   <wrench>0 0 0 0 -0 0</wrench>
932 </link>
933 <link name="Wall_6">
934   <pose frame="">0.668672 3.79062 0 0 -0 3.14159</pose>
935   <velocity>0 0 0 0 -0 0</velocity>
936   <acceleration>0 0 0 0 -0 0</acceleration>
937   <wrench>0 0 0 0 -0 0</wrench>
938 </link>
939 <link name="Wall_8">
940   <pose frame="">-2.56869 2.32064 0 0 0 -1.5708</pose>
941   <velocity>0 0 0 0 -0 0</velocity>
942   <acceleration>0 0 0 0 -0 0</acceleration>
943   <wrench>0 0 0 0 -0 0</wrench>
```

```

944   </link>
945   <link name='Wall_9'>
946     <pose frame="">-1.39369 0.895637 0 0 -0 0</pose>
947     <velocity>0 0 0 -0 0</velocity>
948     <acceleration>0 0 0 -0 0</acceleration>
949     <wrench>0 0 0 -0 0</wrench>
950   </link>
951 </model>
952 <light name='sun'>
953   <pose frame="">0 0 10 0 -0 0</pose>
954 </light>
955 </state>
956 <gui fullscreen='0'>
957   <camera name='user_camera'>
958     <pose frame="">15.4065 -7.32713 9.3415 -0 0.531643 2.6842</pose>
959     <view_controller>orbit</view_controller>
960     <projection_type>perspective</projection_type>
961   </camera>
962 </gui>
963 </world>
964 </sdf>

```

**8. Programazio-kodea:** etxea.world

## 7.9 spawn\_car.launch.py

```

1 import os
2 from ament_index_python.packages import get_package_prefix
3 from ament_index_python.packages import get_package_share_directory
4 from launch import LaunchDescription
5 from launch.actions import DeclareLaunchArgument
6 from launch.substitutions import LaunchConfiguration
7 from launch_ros.actions import Node
8 from launch.actions import IncludeLaunchDescription, ExecuteProcess, TimerAction
9 from launch.launch_description_sources import PythonLaunchDescriptionSource
10
11 import xacro
12
13 def generate_launch_description():
14
15   xacro_file = os.path.join(get_package_share_directory('marti_bot'), 'deskribapena', 'marti_bot.urdf.xacro')
16   assert os.path.exists(xacro_file), "The xacro doesn't exist in "+str(xacro_file)
17
18   install_dir = get_package_prefix('marti_bot')
19
20   if 'GAZEBO_MODEL_PATH' in os.environ:
21     os.environ['GAZEBO_MODEL_PATH'] = os.environ['GAZEBO_MODEL_PATH'] + ':' + install_dir + '/share'
22   else:
23     os.environ['GAZEBO_MODEL_PATH'] = install_dir + "/share"
24
25   if 'GAZEBO_PLUGIN_PATH' in os.environ:
26     os.environ['GAZEBO_PLUGIN_PATH'] = os.environ['GAZEBO_PLUGIN_PATH'] + ':' + install_dir + '/lib'
27   else:
28     os.environ['GAZEBO_PLUGIN_PATH'] = install_dir + '/lib'
29
30
31   robot_description_config = xacro.process_file(xacro_file)
32   robot_desc = robot_description_config.toxml()
33
34   spawn_car = Node(package="gazebo_ros", executable='spawn_entity.py', arguments=
35     ["-topic", "/robot_description", "-entity", "marti"])
36   robot_state_publisher = Node(
37     package="robot_state_publisher",
38     executable="robot_state_publisher",
39     name="robot_state_publisher",

```

```

40     parameters=[  

41         {"robot_description": robot_desc}],  

42         output="screen")  

43  

44     ld = LaunchDescription([  

45         DeclareLaunchArgument(  

46             'use_sim_time',  

47             default_value='true',  

48             description='Use simulation (Gazebo) clock if true'),  

49         TimerAction(  

50             period=0.0,  

51             actions=[spawn_car]),  

52         TimerAction(  

53             period=0.0,  

54             actions=[robot_state_publisher]  

55         )  

56     ])  

57  

58     return ld
  
```

#### 9. Programazio-kodea: spawn\_car.launch.py

## 7.10 marti\_gazebo.launch.py

```

1 import os  

2  

3 from ament_index_python.packages import get_package_share_directory  

4 from launch import LaunchDescription  

5 from launch.actions import DeclareLaunchArgument  

6 from launch.actions import IncludeLaunchDescription, ExecuteProcess, TimerAction  

7 from launch.conditions import IfCondition  

8 from launch.launch_description_sources import PythonLaunchDescriptionSource  

9 from launch.substitutions import LaunchConfiguration  

10 from launch_ros.actions import Node  

11 from launch.launch_description_sources import PythonLaunchDescriptionSource  

12 from launch.substitutions import LaunchConfiguration, PythonExpression  

13  

14  

15 def generate_launch_description():  

16     pkg_gazebo_ros = get_package_share_directory('gazebo_ros')  

17     pkg_marti = get_package_share_directory('marti_bot')  

18  

19     world_config = DeclareLaunchArgument(  

20         'world',  

21         default_value=[os.path.join(pkg_marti, 'worlds', 'etxea.world'), ""],  

22         description='SDF world file')  

23     # # Gazebo launch  

24     gazebo = IncludeLaunchDescription(  

25         PythonLaunchDescriptionSource(  

26             os.path.join(pkg_gazebo_ros, 'launch', 'gazebo.launch.py'),  

27         )  

28     )  

29  

30     marti = IncludeLaunchDescription(  

31         PythonLaunchDescriptionSource(  

32             os.path.join(pkg_marti, 'launch', 'spawn_car.launch.py'),  

33         )  

34     )  

35  

36     ld = LaunchDescription()  

37     ld.add_action(world_config)  

38     ld.add_action(gazebo)  

39     ld.add_action(marti)  

40  

41     return ld
  
```

#### 10. Programazio-kodea: marti\_gazebo.launch.py

## 7.11 lidar.xacro

```
1 <?xml version="1.0"?>
2 <robot xmlns:xacro="http://www.ros.org/wiki/xacro">
3
4   <link name="lidar_link">
5     <inertial>
6       <origin xyz="0 0 0" rpy="0 0 0"/>
7       <mass value="0.125"/>
8       <inertia ixz="0.001" ixy="0" ixz="0" iyy="0.001" iyz="0" izz="0.001" />
9     </inertial>
10
11   <collision>
12     <origin xyz="0 0 0" rpy="0 0 0"/>
13     <geometry>
14       <cylinder radius="0.0508" length="0.055"/>
15     </geometry>
16   </collision>
17
18   <visual>
19     <origin xyz="0 0 0" rpy="0 0 0"/>
20     <geometry>
21       <cylinder radius="0.0508" length="0.055"/>
22     </geometry>
23   </visual>
24 </link>
25
26   <joint name="lidar_joint" type="fixed">
27     <parent link="base_link"/>
28     <child link="lidar_link"/>
29     <origin xyz="0 0 0.12" rpy="0 0 0"/>
30   </joint>
31
32   <gazebo reference="lidar_link">
33     <sensor name="lidar" type="ray">
34       <always_on>true</always_on>
35       <visualize>true</visualize>
36       <update_rate>5</update_rate>
37       <ray>
38         <scan>
39           <horizontal>
40             <samples>360</samples>
41             <resolution>1.000000</resolution>
42             <min_angle>0.000000</min_angle>
43             <max_angle>6.280000</max_angle>
44           </horizontal>
45         </scan>
46         <range>
47           <min>0.120000</min>
48           <max>3.5</max>
49           <resolution>0.015000</resolution>
50         </range>
51         <noise>
52           <type>gaussian</type>
53           <mean>0.0</mean>
54           <stddev>0.01</stddev>
55         </noise>
56       </ray>
57     <plugin name="scan" filename="libgazebo_ros_ray_sensor.so">
58       <ros>
59         <remapping>~out:=scan</remapping>
60       </ros>
61       <output_type>sensor_msgs/LaserScan</output_type>
```

```

62      <frame_name>lidar_link</frame_name>
63    </plugin>
64  </sensor>
65 </gazebo>
66 </robot>
```

11. Programazio-kodea: lidar.xacro

## 7.12 CMakeLists.txt

```

1 cmake_minimum_required(VERSION 3.8)
2 project(marti_bot)
3
4 if(CMAKE_COMPILER_IS_GNUCXX OR CMAKE_CXX_COMPILER_ID MATCHES "Clang")
5   add_compile_options(-Wall -Wextra -Wpedantic)
6 endif()
7
8 # find dependencies
9 find_packageament_cmake REQUIRED)
10 # uncomment the following section in order to fill in
11 # further dependencies manually.
12 # find_package(<dependency> REQUIRED)
13
14 install(
15   DIRECTORY src launch rviz deskribapena worlds params
16   DESTINATION share/${PROJECT_NAME}
17 )
18
19 if(BUILD_TESTING)
20   find_packageament_lint_auto REQUIRED)
21   # the following line skips the linter which checks for copyrights
22   # comment the line when a copyright and license is added to all source files
23   set(ament_cmake_copyright_FOUND TRUE)
24   # the following line skips cpplint (only works in a git repo)
25   # comment the line when this package is in a git repo and when
26   # a copyright and license is added to all source files
27   set(ament_cmake_cpplint_FOUND TRUE)
28   ament_lint_auto_find_test_dependencies()
29 endif()
30
31 ament_package()
```

12. Programazio-kodea: CMakeLists.txt

## 6.13 nav2\_params.yaml

```

1 amcl:
2   ros_parameters:
3     use_sim_time: True
4     alpha1: 0.2 # Expected process noise in odometry's rotation estimate from rotation.
5     alpha2: 0.2 # Expected process noise in odometry's rotation estimate from translation.
6     alpha3: 0.2 # Expected process noise in odometry's translation estimate from translation.
7     alpha4: 0.2 # Expected process noise in odometry's translation estimate from rotation.
8     alpha5: 0.2 # For Omni models only: translation noise.
9     base_frame_id: "base_footprint"
10    beam_skip_distance: 0.5
11    beam_skip_error_threshold: 0.9
12    beam_skip_threshold: 0.3
13    do_beamskip: false
14    global_frame_id: "map"
15    lambda_short: 0.1
16    laser_likelihood_max_dist: 2.0
```

```

17    laser_max_range: -1.0 # Maximum scan range to be considered, -1.0 will cause the laser's reported maximum range to
be used.
18    laser_min_range: -1.0 # Minimum scan range to be considered, -1.0 will cause the laser's reported minimum range to be
used.
19    laser_model_type: "likelihood_field"
20    max_beams: 60
21    max_particles: 2000
22    min_particles: 500
23    odom_frame_id: "odom"
24    pf_err: 0.05
25    pf_z: 0.99
26    recovery_alpha_fast: 0.0
27    recovery_alpha_slow: 0.0
28    resample_interval: 1
29    robot_model_type: "differential" #"nav2_amcl::DifferentialMotionModel"
30    save_pose_rate: 0.5
31    sigma_hit: 0.2
32    tf_broadcast: true
33    transform_tolerance: 1.0
34    update_min_a: 0.2
35    update_min_d: 0.25
36    z_hit: 0.5
37    z_max: 0.05
38    z_rand: 0.5
39    z_short: 0.05
40    scan_topic: scan
41    map_topic: map
42    set_initial_pose: false
43    always_reset_initial_pose: false
44    first_map_only: false
45    initial_pose:
46      x: 0.0
47      y: 0.0
48      z: 0.0
49      yaw: 0.0
50
51 amcl_map_client:
52   ros_parameters:
53     use_sim_time: True
54
55 amcl_rclcpp_node:
56   ros_parameters:
57     use_sim_time: True
58
59 bt_navigator:
60   ros_parameters:
61     use_sim_time: True
62     global_frame: map
63     robot_base_frame: base_link
64     odom_topic: marti/odom
65     bt_loop_duration: 10
66     default_server_timeout: 20
67     enable_groot_monitoring: True
68     groot_zmq_publisher_port: 1666
69     groot_zmq_server_port: 1667
70     # 'default_nav_through_poses_bt_xml' and 'default_nav_to_pose_bt_xml' are use defaults:
71     # nav2_bt_navigator/navigate_to_pose_w_replanning_and_recovery.xml
72     # nav2_bt_navigator/navigate_through_poses_w_replanning_and_recovery.xml
73     # They can be set here or via a RewrittenYaml remap from a parent launch file to Nav2.
74     # default_nav_to_pose_bt_xml:
75     /opt/ros/galactic/share/nav2_bt_navigator/behavior_trees/navigate_w_replanning_time.xml
76     plugin_lib_names:
77       - nav2_compute_path_to_pose_action_bt_node
78       - nav2_compute_path_through_poses_action_bt_node
79       - nav2_follow_path_action_bt_node
80       - nav2_back_up_action_bt_node
81       - nav2_spin_action_bt_node
82       - nav2_wait_action_bt_node
83       - nav2_clear_costmap_service_bt_node
84       - nav2_is_stuck_condition_bt_node

```

```

84 - nav2_goal_reached_condition_bt_node
85 - nav2_goal_updated_condition_bt_node
86 - nav2_initial_pose_received_condition_bt_node
87 - nav2_reinitialize_global_localization_service_bt_node
88 - nav2_rate_controller_bt_node
89 - nav2_distance_controller_bt_node
90 - nav2_speed_controller_bt_node
91 - nav2_truncate_path_action_bt_node
92 - nav2_goal_updater_node_bt_node
93 - nav2_recovery_node_bt_node
94 - nav2_pipeline_sequence_bt_node
95 - nav2_round_robin_node_bt_node
96 - nav2_transform_available_condition_bt_node
97 - nav2_time_expired_condition_bt_node
98 - nav2_distance_traveled_condition_bt_node
99 - nav2_single_trigger_bt_node
100 - nav2_is_battery_low_condition_bt_node
101 - nav2_navigate_through_poses_action_bt_node
102 - nav2_navigate_to_pose_action_bt_node
103 - nav2_remove_passed_goals_action_bt_node
104 - nav2_planner_selector_bt_node
105 - nav2_controller_selector_bt_node
106 - nav2_goal_checker_selector_bt_node
107
108 bt_navigator_rclcpp_node:
109   ros__parameters:
110     use_sim_time: True
111
112 controller_server:
113   ros__parameters:
114     use_sim_time: True
115     controller_frequency: 20.0
116     odom_topic: marti/odom
117     min_x_velocity_threshold: 0.001
118     min_y_velocity_threshold: 0.5
119     min_theta_velocity_threshold: 0.001
120     failure_tolerance: 0.3
121     progress_checker_plugin: "progress_checker"
122     goal_checker_plugins: ["general_goal_checker"] # "precise_goal_checker"
123     controller_plugins: ["FollowPath"]
124
125 # Progress checker parameters
126 progress_checker:
127   plugin: "nav2_controller::SimpleProgressChecker"
128   required_movement_radius: 0.5
129   movement_time_allowance: 10.0
130 general_goal_checker:
131   stateful: True
132   plugin: "nav2_controller::SimpleGoalChecker"
133   xy_goal_tolerance: 0.1
134   yaw_goal_tolerance: 0.25
135
136
137 # RPP
138 FollowPath:
139   plugin: "nav2_regulated_pure_pursuit_controller::RegulatedPurePursuitController"
140   allow_reversing: true
141   approach_velocity_scaling_dist: 0.6
142   cost_scaling_dist: 0.6
143   cost_scaling_gain: 1.0
144   desired_linear_vel: 0.5
145   inflation_cost_scaling_factor: 3.0
146   lookahead_dist: 0.6
147   lookahead_time: 1.5
148   max_allowed_time_to_collision_up_to_carrot: 1.0
149   max_angular_accel: 3.2
150   max_lookahead_dist: 0.9
151   max_robot_pose_search_dist: 1.4875
152   min_approach_linear_velocity: 0.05
153   min_lookahead_dist: 0.3

```

```

154     regulated_linear_scaling_min_radius: 0.9
155     regulated_linear_scaling_min_speed: 0.25
156     transform_tolerance: 0.1
157     use_cost_regulated_linear_velocity_scaling: true
158     use_interpolation: true
159     use_regulated_linear_velocity_scaling: true
160     use_rotate_to_heading: false
161     use_velocity_scaled_lookahead_dist: false
162
163 controller_server_rclcpp_node:
164   ros_parameters:
165     use_sim_time: True
166
167
168 local_costmap:
169   local_costmap:
170     ros_parameters:
171       update_frequency: 5.0
172       publish_frequency: 5.0
173       global_frame: odom
174       robot_base_frame: base_link
175       use_sim_time: True
176       rolling_window: true
177       width: 3
178       height: 3
179       resolution: 0.05
180       footprint: "[ [0.25, 0.200], [0.25, -0.200], [-0.25, -0.200], [-0.25, 0.200] ]"
181       footprint_padding: 0.03
182     plugins: ["static_layer", "inflation_layer", "obstacle_layer"]
183     static_layer:
184       plugin: "nav2_costmap_2d::StaticLayer"
185     # map_subscribe_transient_local: False
186     # map_topic: /map
187     # enabled: True
188     obstacle_layer:
189       plugin: "nav2_costmap_2d::ObstacleLayer"
190       enabled: True
191       observation_sources: scan
192       scan:
193         topic: /scan
194         sensor_frame: laser_scanner_link
195         max_obstacle_height: 2.0
196         clearing: True
197         marking: True
198         data_type: "LaserScan"
199         raytrace_max_range: 10.0
200         raytrace_min_range: 0.2
201         obstacle_max_range: 5.0
202         obstacle_min_range: 0.2
203     inflation_layer:
204       plugin: "nav2_costmap_2d::InflationLayer"
205       cost_scaling_factor: 3.0
206       inflation_radius: 0.55
207       enabled: True
208       always_send_full_costmap: True
209
210
211 global_costmap:
212   global_costmap:
213     ros_parameters:
214       update_frequency: 1.0
215       publish_frequency: 1.0
216       global_frame: map
217       robot_base_frame: base_link
218       use_sim_time: True
219       # robot_radius: 0.32
220       footprint: "[ [0.25, 0.200], [0.25, -0.200], [-0.25, -0.200], [-0.25, 0.200] ]"
221       footprint_padding: 0.05
222
223     resolution: 0.05

```

```

224 track_unknown_space: false # bke
225 plugins: ["static_layer", "obstacle_layer", "inflation_layer"]
226 obstacle_layer:
227   plugin: "nav2_costmap_2d::ObstacleLayer"
228   enabled: True
229   observation_sources: scan
230   scan:
231     topic: /scan
232     sensor_frame: lidar_link
233     max_obstacle_height: 2.0
234     clearing: True
235     marking: True
236     data_type: "LaserScan"
237     raytrace_max_range: 10.0
238     raytrace_min_range: 0.2
239     obstacle_max_range: 5.0
240     obstacle_min_range: 0.2
241 static_layer:
242   plugin: "nav2_costmap_2d::StaticLayer"
243   map_subscribe_transient_local: True
244 inflation_layer:
245   plugin: "nav2_costmap_2d::InflationLayer"
246   cost_scaling_factor: 2.0
247   inflation_radius: 2.0
248   always_send_full_costmap: True
249 global_costmap_client:
250   ros_parameters:
251     use_sim_time: True
252 global_costmap_rclcpp_node:
253   ros_parameters:
254     use_sim_time: True
255
256 map_server:
257   ros_parameters:
258     use_sim_time: True
259     yaml_filename: "etxea_mapa.yaml"
260
261 map_saver:
262   ros_parameters:
263     use_sim_time: True
264     save_map_timeout: 5.0
265     free_thresh_default: 0.25
266     occupied_thresh_default: 0.65
267     map_subscribe_transient_local: True
268
269 # planner_server:
270 #   ros_parameters:
271 #     expected_planner_frequency: 20.0
272 #     use_sim_time: True
273 #     planner_plugins: ["GridBased"]
274 #     GridBased:
275 #       plugin: "nav2_navfn_planner/NavfnPlanner"
276 #       tolerance: 0.5
277 #       use_astar: false
278 #       allow_unknown: true
279
280 planner_server:
281   ros_parameters:
282     planner_plugins: ["GridBased"]
283     use_sim_time: True
284
285
286 GridBased:
287   plugin: "nav2_smac_planner/SmacPlannerHybrid"
288   downsample_costmap: false      # whether or not to downsample the map
289   downsampling_factor: 1        # multiplier for the resolution of the costmap layer (e.g. 2 on a 5cm costmap would be
10cm)
290   allow_unknown: true          # allow traveling in unknown space
291   max_iterations: 1000000      # maximum total iterations to search for before failing (in case unreachable), set to -1 to
disable
  
```

```

292   max_planning_time: 5.0          # max time in s for planner to plan, smooth
293   motion_model_for_search: "REEDS_SHEPP" # Hybrid-A* Dubin, Redds-Shepp
294   angle_quantization_bins: 72      # Number of angle bins for search
295   analytic_expansion_ratio: 3.5   # The ratio to attempt analytic expansions during search for final approach.
296   analytic_expansion_max_length: 3.0 # For Hybrid/Lattice nodes: The maximum length of the analytic expansion to be
considered valid to prevent unsafe shortcutting
297   minimum_turning_radius: 1.0     # minimum turning radius in m of path / vehicle
298   reverse_penalty: 50.0          # Penalty to apply if motion is reversing, must be => 1
299   change_penalty: 0.0            # Penalty to apply if motion is changing directions (L to R), must be >= 0
300   non_straight_penalty: 1.2     # Penalty to apply if motion is non-straight, must be => 1
301   cost_penalty: 3.0             # Penalty to apply to higher cost areas when adding into the obstacle map dynamic
programming distance expansion heuristic. This drives the robot more towards the center of passages. A value between 1.3
- 3.5 is reasonable.
302   lookup_table_size: 20.0        # Size of the dubin/reeds-sheep distance window to cache, in meters.
303   cache_obstacle_heuristic: false # Cache the obstacle map dynamic programming distance expansion heuristic
between subsequent replannings of the same goal location. Dramatically speeds up replanning performance (40x) if costmap
is largely static.
304   smooth_path: False # was True           # If true, does a simple and quick smoothing post-processing to the path
305
306 smoother:
307   max_iterations: 1000
308   w_smooth: 0.1 # was 0.3
309   w_data: 0.3 # was 0.2
310   tolerance: 1e-10
311   do_refinement: true
312
313 planner_server_rclcpp_node:
314   ros__parameters:
315     use_sim_time: True
316
317 recoveries_server:
318   ros__parameters:
319     costmap_topic: local_costmap/costmap_raw
320     footprint_topic: local_costmap/published_footprint
321     cycle_frequency: 10.0
322     recovery_plugins: [ "backup", "wait" ]
323     backup:
324       plugin: "nav2_recoveries/BackUp"
325     wait:
326       plugin: "nav2_recoveries/Wait"
327     global_frame: odom
328     robot_base_frame: base_link
329     transform_timeout: 0.1
330     use_sim_time: true
331     simulate_ahead_time: 2.0
332     max_rotational_vel: 1.0
333     min_rotational_vel: 0.4
334     rotational_acc_lim: 3.2
335
336 robot_state_publisher:
337   ros__parameters:
338     use_sim_time: True
339
340 velocity_smoother:
341   ros__parameters:
342     smoothing_frequency: 20.0      # Hz
343     scale_velocities: true        # true = adjust angular to keep same trajectory
344     feedback: "OPEN_LOOP"         # CLOSED_LOOP means use velocities from odom, odom must be high rate
compared to smoothing_frequency
345     max_velocity: [5.0, 0.0, 92.5] # x, y, theta
346     min_velocity: [-5.0, 0.0, -92.5] # x, y, theta, maximum reverse velocities (should be negative)
347     deadband_velocity: [0.0, 0.0, 0.0] # velocities less than this will be translted to zero
348     velocity_timeout: 1.0          # if no cmd_vel is sent after this time, zeros will be sent
349     max_accel: [2.5, 0.0, 933.2]
350     max_decel: [-2.5, 0.0, -933.2]
351     odom_topic: "marti/odom"      # topic for closed loop velocities
352     odom_duration: 0.1            # time to buffer odometry to esimate speed
353
354 slam_toolbox:

```

```

356 ros__parameters:
357     minimum_travel_distance: 0.01
358     minimum_travel_heading: 0.01
359     map_update_interval: 1.0
360     loop_search_maximum_distance: 0.5
361
362
363
364 waypoint_follower:
365     ros__parameters:
366         loop_rate: 20
367         stop_on_failure: false
368         waypoint_task_executor_plugin: "wait_at_waypoint"
369         wait_at_waypoint:
370             plugin: "nav2_waypoint_follower::WaitAtWaypoint"
371             enabled: True
372             waypoint_pause_duration: 200

```

**13. Programazio-kodea:** nav2\_params.yaml

## 7.14 nav2 default view.rviz

```

1 Panels:
2     - Class: rviz_common/Displays
3     Help Height: 0
4     Name: Displays
5     Property Tree Widget:
6         Expanded:
7             - /Global Options1
8             - /TF1/Frames1
9             - /TF1/Tree1
10            - /Camera1/Visibility1
11            - /Map1
12            Splitter Ratio: 0.5833333134651184
13     Tree Height: 606
14     - Class: rviz_common/Selection
15     Name: Selection
16     - Class: rviz_common/Tool Properties
17         Expanded:
18             - /Publish Point1
19             Name: Tool Properties
20             Splitter Ratio: 0.5886790156364441
21     - Class: rviz_common/Views
22         Expanded:
23             - /Current View1
24             Name: Views
25             Splitter Ratio: 0.5
26         - Class: nav2_rviz_plugins/Navigation 2
27             Name: Navigation 2
28     Visualization Manager:
29         Class: ""
30     Displays:
31         - Alpha: 0.5
32         Cell Size: 1
33         Class: rviz_default_plugins/Grid
34         Color: 160; 160; 164
35         Enabled: true
36         Line Style:
37             Line Width: 0.029999999329447746
38             Value: Lines
39             Name: Grid
40             Normal Cell Count: 0
41             Offset:
42                 X: 0
43                 Y: 0
44                 Z: 0
45             Plane: XY
46             Plane Cell Count: 10
47             Reference Frame: <Fixed Frame>

```

```

48   Value: true
49   - Alpha: 1
50   Class: rviz_default_plugins/RobotModel
51   Collision Enabled: false
52   Description File: ""
53   Description Source: Topic
54   Description Topic:
55     Depth: 5
56     Durability Policy: Volatile
57     History Policy: Keep Last
58     Reliability Policy: Reliable
59     Value: /robot_description
60   Enabled: true
61   Links:
62     All Links Enabled: true
63     Expand Joint Details: false
64     Expand Link Details: false
65     Expand Tree: false
66     Link Tree Style: Links in Alphabetic Order
67   atzekoesk_gurp_link:
68     Alpha: 1
69     Show Axes: false
70     Show Trail: false
71     Value: true
72   atzekoezk_gurp_link:
73     Alpha: 1
74     Show Axes: false
75     Show Trail: false
76     Value: true
77   aurrekoesk_gurp_link:
78     Alpha: 1
79     Show Axes: false
80     Show Trail: false
81     Value: true
82   aurrekoesk_gurp_steer:
83     Alpha: 1
84     Show Axes: false
85     Show Trail: false
86     Value: true
87   aurrekoezk_gurp_link:
88     Alpha: 1
89     Show Axes: false
90     Show Trail: false
91     Value: true
92   aurrekoezk_gurp_steer:
93     Alpha: 1
94     Show Axes: false
95     Show Trail: false
96     Value: true
97   base_footprint:
98     Alpha: 1
99     Show Axes: false
100    Show Trail: false
101  base_link:
102    Alpha: 1
103    Show Axes: false
104    Show Trail: false
105    Value: true
106  lidar_link:
107    Alpha: 1
108    Show Axes: false
109    Show Trail: false
110    Value: true
111  Mass Properties:
112    Inertia: false
113    Mass: false
114    Name: RobotModel
115    TF Prefix: ""
116    Update Interval: 0
117    Value: true

```

```

118     Visual Enabled: true
119     - Class: rviz_default_plugins/TF
120         Enabled: true
121         Frame Timeout: 15
122         Frames:
123             All Enabled: false
124             atzekoesk_gurp_link:
125                 Value: true
126             atzekoezk_gurp_link:
127                 Value: true
128             aurrekoesk_gurp_link:
129                 Value: true
130             aurrekoesk_gurp_steer:
131                 Value: true
132             aurrekoezk_gurp_link:
133                 Value: true
134             aurrekoezk_gurp_steer:
135                 Value: true
136             base_footprint:
137                 Value: true
138             base_link:
139                 Value: true
140             lidar_link:
141                 Value: false
142             map:
143                 Value: false
144             odom:
145                 Value: false
146             Marker Scale: 1
147             Name: TF
148             Show Arrows: true
149             Show Axes: true
150             Show Names: false
151             Tree:
152             map:
153                 odom:
154                     base_footprint:
155                         base_link:
156                             atzekoezk_gurp_link:
157                                 {}
158                             atzekoesk_gurp_link:
159                                 {}
160                             aurrekoezk_gurp_steer:
161                             aurrekoezk_gurp_link:
162                                 {}
163                             aurrekoesk_gurp_steer:
164                             aurrekoesk_gurp_link:
165                                 {}
166                             lidar_link:
167                                 {}
168             Update Interval: 0
169             Value: true
170             - Alpha: 1
171                 Autocompute Intensity Bounds: true
172                 Autocompute Value Bounds:
173                     Max Value: 10
174                     Min Value: -10
175                     Value: true
176             Axis: Z
177             Channel Name: intensity
178             Class: rviz_default_plugins/LaserScan
179             Color: 255; 255; 255
180             Color Transformer: Intensity
181             Decay Time: 0
182             Enabled: true
183             Invert Rainbow: false
184             Max Color: 255; 255; 255
185             Max Intensity: 0
186             Min Color: 0; 0; 0
187             Min Intensity: 0

```

```

188     Name: LaserScan
189     Position Transformer: XYZ
190     Selectable: true
191     Size (Pixels): 3
192     Size (m): 0.009999999776482582
193     Style: Points
194     Topic:
195         Depth: 5
196         Durability Policy: Volatile
197         Filter size: 10
198         History Policy: Keep Last
199         Reliability Policy: Best Effort
200         Value: /scan
201     Use Fixed Frame: true
202     Use rainbow: true
203     Value: true
204     - Alpha: 1
205     Autocompute Intensity Bounds: true
206     Autocompute Value Bounds:
207         Max Value: 10
208         Min Value: -10
209         Value: true
210     Axis: Z
211     Channel Name: intensity
212     Class: rviz_default_plugins/PointCloud2
213     Color: 255; 255; 255
214     Color Transformer: ""
215     Decay Time: 0
216     Enabled: true
217     Invert Rainbow: false
218     Max Color: 255; 255; 255
219     Max Intensity: 4096
220     Min Color: 0; 0; 0
221     Min Intensity: 0
222     Name: Bumper Hit
223     Position Transformer: ""
224     Selectable: true
225     Size (Pixels): 3
226     Size (m): 0.07999999821186066
227     Style: Spheres
228     Topic:
229         Depth: 5
230         Durability Policy: Volatile
231         Filter size: 10
232         History Policy: Keep Last
233         Reliability Policy: Best Effort
234         Value: /mobile_base/sensors/bumper_pointcloud
235     Use Fixed Frame: true
236     Use rainbow: true
237     Value: true
238     - Alpha: 1
239     Class: rviz_default_plugins/Map
240     Color Scheme: map
241     Draw Behind: true
242     Enabled: true
243     Name: Map
244     Topic:
245         Depth: 1
246         Durability Policy: Transient Local
247         History Policy: Keep Last
248         Reliability Policy: Reliable
249         Value: /map
250     Use Timestamp: false
251     Value: true
252     - Alpha: 1
253     Class: nav2_rviz_plugins/ParticleCloud
254     Color: 0; 180; 0
255     Enabled: true
256     Max Arrow Length: 0.3
257     Min Arrow Length: 0.02

```

```

258   Name: Amcl Particle Swarm
259   Shape: Arrow (Flat)
260   Topic:
261     Depth: 5
262     Durability Policy: Volatile
263     History Policy: Keep Last
264     Reliability Policy: Best Effort
265     Value: /particle_cloud
266     Value: true
267   - Class: rviz_common/Group
268   Displays:
269     - Alpha: 0.3
270     Class: rviz_default_plugins/Map
271     Color Scheme: costmap
272     Draw Behind: false
273     Enabled: true
274     Name: Global Costmap
275     Topic:
276       Depth: 1
277       Durability Policy: Transient Local
278       History Policy: Keep Last
279       Reliability Policy: Reliable
280       Value: /global_costmap/costmap
281     Use Timestamp: false
282     Value: true
283   - Alpha: 0.3
284   Class: rviz_default_plugins/Map
285   Color Scheme: costmap
286   Draw Behind: false
287   Enabled: true
288   Name: Downsampled Costmap
289   Topic:
290     Depth: 1
291     Durability Policy: Transient Local
292     Filter size: 10
293     History Policy: Keep Last
294     Reliability Policy: Reliable
295     Value: /downsampled_costmap
296   Update Topic:
297     Depth: 5
298     Durability Policy: Volatile
299     History Policy: Keep Last
300     Reliability Policy: Reliable
301     Value: /downsampled_costmap_updates
302   Use Timestamp: false
303   Value: true
304   - Alpha: 1
305   Buffer Length: 1
306   Class: rviz_default_plugins/Path
307   Color: 255; 0; 0
308   Enabled: true
309   Head Diameter: 0.0199999952965164
310   Head Length: 0.0199999952965164
311   Length: 0.30000001192092896
312   Line Style: Lines
313   Line Width: 0.029999999329447746
314   Name: Path
315   Offset:
316     X: 0
317     Y: 0
318     Z: 0
319   Pose Color: 255; 85; 255
320   Pose Style: Arrows
321   Radius: 0.02999999329447746
322   Shaft Diameter: 0.00499999888241291
323   Shaft Length: 0.0199999952965164
324   Topic:
325     Depth: 5
326     Durability Policy: Volatile
327     Filter size: 10

```

```

328     History Policy: Keep Last
329     Reliability Policy: Reliable
330     Value: /plan
331     Value: true
332 - Alpha: 1
333     Autocompute Intensity Bounds: true
334     Autocompute Value Bounds:
335         Max Value: 10
336         Min Value: -10
337         Value: true
338     Axis: Z
339     Channel Name: intensity
340     Class: rviz_default_plugins/PointCloud2
341     Color: 125; 125; 125
342     Color Transformer: FlatColor
343     Decay Time: 0
344     Enabled: true
345     Invert Rainbow: false
346     Max Color: 255; 255; 255
347     Max Intensity: 4096
348     Min Color: 0; 0; 0
349     Min Intensity: 0
350     Name: VoxelGrid
351     Position Transformer: XYZ
352     Selectable: true
353     Size (Pixels): 3
354     Size (m): 0.05000000074505806
355     Style: Boxes
356     Topic:
357         Depth: 5
358         Durability Policy: Volatile
359         Filter size: 10
360         History Policy: Keep Last
361         Reliability Policy: Reliable
362         Value: /global_costmap/voxel_marked_cloud
363     Use Fixed Frame: true
364     Use rainbow: true
365     Value: true
366 - Alpha: 1
367     Class: rviz_default_plugins/Polygon
368     Color: 25; 255; 0
369     Enabled: false
370     Name: Polygon
371     Topic:
372         Depth: 5
373         Durability Policy: Volatile
374         Filter size: 10
375         History Policy: Keep Last
376         Reliability Policy: Reliable
377         Value: /global_costmap/published_footprint
378     Value: false
379     Enabled: true
380     Name: Global Planner
381 - Class: rviz_common/Group
382     Displays:
383 - Alpha: 0.699999988079071
384     Class: rviz_default_plugins/Map
385     Color Scheme: costmap
386     Draw Behind: false
387     Enabled: true
388     Name: Local Costmap
389     Topic:
390         Depth: 1
391         Durability Policy: Transient Local
392         Filter size: 10
393         History Policy: Keep Last
394         Reliability Policy: Reliable
395         Value: /local_costmap/costmap
396     Update Topic:
397         Depth: 5

```

```

398 Durability Policy: Volatile
399 History Policy: Keep Last
400 Reliability Policy: Reliable
401 Value: /local_costmap/costmap_updates
402 Use Timestamp: false
403 Value: true
404 - Alpha: 1
405 Buffer Length: 1
406 Class: rviz_default_plugins/Path
407 Color: 0; 12; 255
408 Enabled: true
409 Head Diameter: 0.30000001192092896
410 Head Length: 0.20000000298023224
411 Length: 0.30000001192092896
412 Line Style: Lines
413 Line Width: 0.029999999329447746
414 Name: Local Plan
415 Offset:
416   X: 0
417   Y: 0
418   Z: 0
419 Pose Color: 255; 85; 255
420 Pose Style: None
421 Radius: 0.029999999329447746
422 Shaft Diameter: 0.10000000149011612
423 Shaft Length: 0.10000000149011612
424 Topic:
425   Depth: 5
426   Durability Policy: Volatile
427   Filter size: 10
428   History Policy: Keep Last
429   Reliability Policy: Reliable
430   Value: /local_plan
431 Value: true
432 - Class: rviz_default_plugins/MarkerArray
433 Enabled: false
434 Name: Trajectories
435 Namespaces:
436   {}
437 Topic:
438   Depth: 5
439   Durability Policy: Volatile
440   History Policy: Keep Last
441   Reliability Policy: Reliable
442   Value: /marker
443 Value: false
444 - Alpha: 1
445 Class: rviz_default_plugins/Polygon
446 Color: 25; 255; 0
447 Enabled: true
448 Name: Polygon
449 Topic:
450   Depth: 5
451   Durability Policy: Volatile
452   Filter size: 10
453   History Policy: Keep Last
454   Reliability Policy: Reliable
455   Value: /local_costmap/published_footprint
456 Value: true
457 - Alpha: 1
458 Autocompute Intensity Bounds: true
459 Autocompute Value Bounds:
460   Max Value: 10
461   Min Value: -10
462   Value: true
463 Axis: Z
464 Channel Name: intensity
465 Class: rviz_default_plugins/PointCloud2
466 Color: 255; 255; 255
467 Color Transformer: RGB8

```

```

468     Decay Time: 0
469     Enabled: true
470     Invert Rainbow: false
471     Max Color: 255; 255; 255
472     Max Intensity: 4096
473     Min Color: 0; 0; 0
474     Min Intensity: 0
475     Name: VoxelGrid
476     Position Transformer: XYZ
477     Selectable: true
478     Size (Pixels): 3
479     Size (m): 0.009999999776482582
480     Style: Flat Squares
481     Topic:
482         Depth: 5
483         Durability Policy: Volatile
484         Filter size: 10
485         History Policy: Keep Last
486         Reliability Policy: Reliable
487         Value: /local_costmap/voxel_marked_cloud
488     Use Fixed Frame: true
489     Use rainbow: true
490     Value: true
491     Enabled: true
492     Name: Controller
493     - Class: rviz_common/Group
494     Displays:
495         - Class: rviz_default_plugins/Image
496             Enabled: true
497             Max Value: 1
498             Median window: 5
499             Min Value: 0
500             Name: RealsenseCamera
501             Normalize Range: true
502             Topic:
503                 Depth: 5
504                 Durability Policy: Volatile
505                 History Policy: Keep Last
506                 Reliability Policy: Reliable
507                 Value: /intel_realsense_r200_depth/image_raw
508             Value: true
509             - Alpha: 1
510             Autocompute Intensity Bounds: true
511             Autocompute Value Bounds:
512                 Max Value: 10
513                 Min Value: -10
514                 Value: true
515                 Axis: Z
516                 Channel Name: intensity
517                 Class: rviz_default_plugins/PointCloud2
518                 Color: 255; 255; 255
519                 Color Transformer: RGB8
520                 Decay Time: 0
521                 Enabled: true
522                 Invert Rainbow: false
523                 Max Color: 255; 255; 255
524                 Max Intensity: 4096
525                 Min Color: 0; 0; 0
526                 Min Intensity: 0
527                 Name: RealsenseDepthImage
528                 Position Transformer: XYZ
529                 Selectable: true
530                 Size (Pixels): 3
531                 Size (m): 0.009999999776482582
532                 Style: Flat Squares
533                 Topic:
534                     Depth: 5
535                     Durability Policy: Volatile
536                     Filter size: 10
537                     History Policy: Keep Last

```

```

538     Reliability Policy: Reliable
539     Value: /intel_realsense_r200_depth/points
540     Use Fixed Frame: true
541     Use rainbow: true
542     Value: true
543     Enabled: false
544     Name: Realsense
545     - Class: rviz_default_plugins/MarkerArray
546     Enabled: true
547     Name: MarkerArray
548     Namespaces:
549     {}
550     Topic:
551     Depth: 5
552     Durability Policy: Volatile
553     History Policy: Keep Last
554     Reliability Policy: Reliable
555     Value: /waypoints
556     Value: true
557     - Class: rviz_default_plugins/Camera
558     Enabled: true
559     Image Rendering: background and overlay
560     Name: Camera
561     Overlay Alpha: 0.5
562     Topic:
563     Depth: 5
564     Durability Policy: Volatile
565     History Policy: Keep Last
566     Reliability Policy: Reliable
567     Value: /camera/image_raw
568     Value: true
569     Visibility:
570     Grid: true
571     LaserScan: true
572     RobotModel: false
573     TF: false
574     Value: true
575     Zoom Factor: 1
576     Enabled: true
577     Global Options:
578     Background Color: 48; 48; 48
579     Fixed Frame: map
580     Frame Rate: 30
581     Name: root
582     Tools:
583     - Class: rviz_default_plugins/MoveCamera
584     - Class: rviz_default_plugins/Select
585     - Class: rviz_default_plugins/FocusCamera
586     - Class: rviz_default_plugins/Measure
587     Line color: 128; 128; 0
588     - Class: rviz_default_plugins/SetInitialPose
589     Covariance x: 0.25
590     Covariance y: 0.25
591     Covariance yaw: 0.06853891909122467
592     Topic:
593     Depth: 5
594     Durability Policy: Volatile
595     History Policy: Keep Last
596     Reliability Policy: Reliable
597     Value: /initialpose
598     - Class: rviz_default_plugins/SetGoal
599     Topic:
600     Depth: 5
601     Durability Policy: Volatile
602     History Policy: Keep Last
603     Reliability Policy: Reliable
604     Value: /goal_pose
605     - Class: rviz_default_plugins/PublishPoint
606     Single click: true
607     Topic:

```



## 7.15 rviz\_launch.py

```
1 import os
2 from ament_index_python.packages import get_package_share_directory
3
4 from launch import LaunchDescription
5 from launch.actions import DeclareLaunchArgument, EmitEvent, RegisterEventHandler
6 from launch.conditions import IfCondition, UnlessCondition
7 from launch.event_handlers import OnProcessExit
8 from launch.events import Shutdown
9 from launch.substitutions import LaunchConfiguration
10 from launch_ros.actions import Node
11 from nav2_common.launch import ReplaceString
12
13
14 def generate_launch_description():
15     # Get the launch directory
16     bringup_dir = get_package_share_directory('marti_bot')
17
18     # Create the launch configuration variables
19     namespace = LaunchConfiguration('namespace')
20     use_namespace = LaunchConfiguration('use_namespace')
21     rviz_config_file = LaunchConfiguration('rviz_config')
22
23     # Declare the launch arguments
24     declare_namespace_cmd = DeclareLaunchArgument(
25         'namespace',
26         default_value='navigation',
27         description=('Top-level namespace. The value will be used to replace the '
28                     '<robot_namespace> keyword on the rviz config file.'))
29
30     declare_use_namespace_cmd = DeclareLaunchArgument(
31         'use_namespace',
32         default_value='false',
33         description='Whether to apply a namespace to the navigation stack')
34
35     declare_rviz_config_file_cmd = DeclareLaunchArgument(
36         'rviz_config',
37         default_value=os.path.join(bringup_dir, 'rviz', 'nav2_default_view.rviz'),
38         description='Full path to the RVIZ config file to use')
39
40     # Launch rviz
41     start_rviz_cmd = Node(
42         condition=UnlessCondition(use_namespace),
43         package='rviz2',
44         executable='rviz2',
45         arguments=['-d', rviz_config_file],
46         output='screen')
47
48     namespaced_rviz_config_file = ReplaceString(
49         source_file=rviz_config_file,
50         replacements={'<robot_namespace>': ('/', namespace)})
51
52     start_namespaced_rviz_cmd = Node(
53         condition=IfCondition(use_namespace),
54         package='rviz2',
55         executable='rviz2',
56         namespace=namespace,
57         arguments=['-d', namespaced_rviz_config_file],
58         output='screen',
59         remappings=[('/tf', 'tf'),
60                     ('/tf_static', 'tf_static')],
```

```

1      ('/goal_pose', 'goal_pose'),
2      ('/clicked_point', 'clicked_point'),
3      ('/initialpose', 'initialpose'))]
4
5      exit_event_handler = RegisterEventHandler(
6          condition=UnlessCondition(use_namespace),
7          event_handler=OnProcessExit(
8              target_action=start_rviz_cmd,
9              on_exit=EmitEvent(event=Shutdown(reason='rviz exited'))))
10
11      exit_event_handler_namespaced = RegisterEventHandler(
12          condition=IfCondition(use_namespace),
13          event_handler=OnProcessExit(
14              target_action=start_namespaced_rviz_cmd,
15              on_exit=EmitEvent(event=Shutdown(reason='rviz exited'))))
16
17      # Create the launch description and populate
18      ld = LaunchDescription()
19
20      # Declare the launch options
21      ld.add_action(declare_namespace_cmd)
22      ld.add_action(declare_use_namespace_cmd)
23      ld.add_action(declare_rviz_config_file_cmd)
24
25      # Add any conditioned actions
26      ld.add_action(start_rviz_cmd)
27      ld.add_action(start_namespaced_rviz_cmd)
28
29      # Add other nodes and processes we need
30      ld.add_action(exit_event_handler)
31      ld.add_action(exit_event_handler_namespaced)
32
33      return ld

```

### 15. Programazio-kodea: rviz\_launch.py

## 7.16 nabegazioa.launch.py

```

1 import os
2
3 from ament_index_python.packages import get_package_share_directory
4 from launch import LaunchDescription
5 from launch.actions import DeclareLaunchArgument, ExecuteProcess, IncludeLaunchDescription
6 from launch.conditions import IfCondition
7 from launch.launch_description_sources import PythonLaunchDescriptionSource
8 from launch.substitutions import LaunchConfiguration, PythonExpression
9 from launch_ros.actions import Node
10 from launch_ros.substitutions import FindPackageShare
11
12 from nav2_common.launch import RewrittenYaml
13
14 def generate_launch_description():
15     # Get the launch directory
16     bringup_dir = get_package_share_directory('marti_bot')
17     launch_dir = os.path.join(bringup_dir, 'launch')
18
19     nav2_dir = FindPackageShare(package='nav2_bringup').find('nav2_bringup')
20     nav2_launch_dir = os.path.join(nav2_dir, 'launch')
21
22     # Create the launch configuration variables
23     slam = LaunchConfiguration('slam')
24     namespace = LaunchConfiguration('namespace')
25     use_namespace = LaunchConfiguration('use_namespace')
26     map_yaml_file = LaunchConfiguration('map')

```

```

27 use_sim_time = LaunchConfiguration('use_sim_time')
28 params_file = LaunchConfiguration('params_file')
29 autostart = LaunchConfiguration('autostart')
30 use_composition = LaunchConfiguration('use_composition')
31 use_respawn = LaunchConfiguration('use_respawn')
32
33 # Launch configuration variables specific to simulation
34 rviz_config_file = LaunchConfiguration('rviz_config_file')
35 use_rviz = LaunchConfiguration('use_rviz')
36
37 # Declare the launch arguments
38 declare_namespace_cmd = DeclareLaunchArgument(
39   'namespace',
40   default_value='',
41   description='Top-level namespace')
42
43 declare_use_namespace_cmd = DeclareLaunchArgument(
44   'use_namespace',
45   default_value='false',
46   description='Whether to apply a namespace to the navigation stack')
47
48 declare_slam_cmd = DeclareLaunchArgument(
49   'slam',
50   default_value='False',
51   description='Whether run a SLAM')
52
53 declare_map_yaml_cmd = DeclareLaunchArgument(
54   'map',
55   default_value=os.path.join(
56     bringup_dir, 'maps', 'etxea_mapa.yaml'),
57   description='Full path to map file to load')
58
59 declare_use_sim_time_cmd = DeclareLaunchArgument(
60   'use_sim_time',
61   default_value='true',
62   description='Use simulation (Gazebo) clock if true')
63
64 declare_params_file_cmd = DeclareLaunchArgument(
65   'params_file',
66   default_value=os.path.join(bringup_dir, 'params', 'nav2_params.yaml'),
67   description='Full path to the ROS2 parameters file to use for all launched nodes')
68
69 declare_autostart_cmd = DeclareLaunchArgument(
70   'autostart', default_value='true',
71   description='Automatically startup the nav2 stack')
72
73 declare_use_composition_cmd = DeclareLaunchArgument(
74   'use_composition', default_value='True',
75   description='Whether to use composed bringup')
76
77 declare_use_respawn_cmd = DeclareLaunchArgument(
78   'use_respawn', default_value='False',
79   description='Whether to respawn if a node crashes. Applied when composition is disabled.')
80
81 declare_rviz_config_file_cmd = DeclareLaunchArgument(
82   'rviz_config_file',
83   default_value=os.path.join(
84     bringup_dir, 'rviz', 'nav2_default_view.rviz'),
85   description='Full path to the RVIZ config file to use')
86
87 declare_use_rviz_cmd = DeclareLaunchArgument(
88   'use_rviz',
89   default_value='True',
90   description='Whether to start RVIZ')
91
92 # Specify the actions
93 urdf = os.path.join(bringup_dir, 'deskribapena', 'marti_bot.urdf.xacro')
94 with open(urdf, 'r') as infp:
95   robot_description = infp.read()
96

```

```

97 rviz_cmd = IncludeLaunchDescription(
98   PythonLaunchDescriptionSource(
99     os.path.join(launch_dir, 'rviz_launch.py')),
100   condition=IfCondition(use_rviz),
101   launch_arguments={'namespace': namespace,
102     'use_namespace': use_namespace,
103     'rviz_config': rviz_config_file}.items())
104
105 bringup_cmd = IncludeLaunchDescription(
106   PythonLaunchDescriptionSource(
107     os.path.join(nav2_launch_dir, 'bringup.launch.py')),
108   launch_arguments={'namespace': namespace,
109     'use_namespace': use_namespace,
110     'slam': "1",
111     'map': map_yaml_file,
112     'use_sim_time': use_sim_time,
113     'params_file': params_file,
114     '#params_file': configured_nav2_params,
115     'autostart': autostart,
116     'use_composition': use_composition,
117     'use_respawn': use_respawn}.items())
118
119 # Create the launch description and populate
120 ld = LaunchDescription()
121
122 # Declare the launch options
123 ld.add_action(declare_namespace_cmd)
124 ld.add_action(declare_use_namespace_cmd)
125 ld.add_action(declare_slam_cmd)
126 ld.add_action(declare_map_yaml_cmd)
127 ld.add_action(declare_use_sim_time_cmd)
128 ld.add_action(declare_params_file_cmd)
129 ld.add_action(declare_autostart_cmd)
130 ld.add_action(declare_use_composition_cmd)
131
132 ld.add_action(declare_rviz_config_file_cmd)
133 ld.add_action(declare_use_rviz_cmd)
134
135 ld.add_action(declare_use_respawn_cmd)
136
137 # Add the actions to launch all of the navigation nodes
138 ld.add_action(rviz_cmd)
139 ld.add_action(bringup_cmd)
140
141 return ld

```

#### 16. Programazio-kodea: nabegazioa.launch.py