



Universidad
del País Vasco Euskal Herriko
Unibertsitatea

Sentsoreen Funtzioamendua Lantzen

Robotika eta Kontrol Admineduak

Gorka Dabó

01/11/2024

Indizea

| | |
|---|---|
| 1-Oztopoak Ekiditen | 3 |
| a) Nola erabili duzu laserraren informazioa helburua lortzeko? | 3 |
| b) Nola egokitutu dituzu abiadurak, bai lineala, bai biraketa-abiadura, egoera desberdinetan? | 4 |
| c) Zenbat denboran egon daiteke funtzionamenduan talkarik jaso gabe? (gehienez 10 min testatu) | 4 |
| d) Denbora horretan ingurune osora mugitzen da robota edo eskualde jakin batera mugatzen da? | 5 |
| e) Robotaren hasierako posizioa aldatuta joera berdina erakusten du? | 5 |
| f) Zeintzuk dira egoera arriskutsuak? | 7 |

1-Oztopoak Ekiditen

a) Nola erabili duzu laserraren informazioa helburua lortzeko?

Diseinatutako estrategiaren helburua robotak oztopoak saihestuz aurrera egitea da, eta horretarako, laserrak emandako informazioa ondorengo moduan kudeatu dugu:

1. **Intereseko eremua aukeratzea:** Estrategia honek robotaren aurrealdeko 180°-ko eremuko laser-datuak soilik erabiltzen ditu. Horrela, oztopoak aurretik saihesteko aukera aztertzen da, eta robotak beharrezkoa ez den atzoko zatiko informazioa ez du prozesatzen.
2. **Segmentu libreak identifikatzea:** Algoritmoak eremu horretako laser-datuak iteratzen ditu eta segmentu libreak identifikatzen ditu, distantzia-tarte jakin batetik gorakoak hartuz (2 metro baino handiagoak). Segmentu bakoitzaren luzera eta batez besteko distantzia kalkulatzen dira, zabalena eta sakonena den segmentua aukeratzeko.
3. **Segmentu libre zabalenerantz zuzendu:** Segmentu librerenbat badago, algoritmoak zabalena eta sakonena denaren erdigunearekiko distantzia kalkulatzen du eta eremu horretarantz biratzen du robota, noranzko zuzena kalkulatzeko beharrezkoa den doikuntza eginez.
4. **Biraketa eta abiadura kontrolatzea:** Helburuko noranzkoarekiko desbideratzea errore bezala hartzen dugu, eta errorearen arabera biraketa eta abiadura erabakitzen ditugu. Zenbat eta desbideratze handiagoa egon, orduan eta biraketa handiagoa eta abiadura txikiagoa aplikatzen dugu (eta alderantziz ere), robotaren mugimenduak modu seguru eta leun batean egiteko.
5. **Bide librerik aurkitzen ez bada:** Segmentu librerik ez badago, robota geldi geratzen da eta guk modu arbitrarioan definitutako 70 graduko biraketa egiten du, hurrengo norabidean oztopoak saihesteko aukera izan dezan.

Beraz, laserraren informazioa erabiliz, robotak ingurunea aztertu eta segurtasunez aurrera egiteko beharrezkoak diren doikuntzak eta kalkuluak egiten ditu, eta estrategia honek robotari segurtasunez aurrera egiteko gaitasuna ematen dio. Honetaz gain, esan beharra dago, guk makina birtual batean egin dugula simulazioa eta robota nahiko poliki mugitzen zela klaseko ordenagailuekin konparatuz.

- b) Nola egokitu dituzu abiadurak, bai lineala, bai biraketa-abiadura, egoera desberdinetan?

Kode honetan, egoeraren arabera abiadurak modu batean edo bestean egokitu ditugu, robotak ingurunea modu seguru eta egonkorrean azter dezan.

1. **Biraketa-abiadura:** Biraketa angelua desbideratzearen arabera egokitzen dugu, ‘error’ izeneko aldagai baten bidez, zeinak 0.3 koefizientearekin biderkatuta biraketa polikiago egizarazten duen. Kalkulu honek robotak mugimenduan duen desbideratze errorea konpentsatzen du, helburantz biratu eta egokitzen baitu. Errorea handiagoa denean, biraketa angelu handia ezartzen da, eta txikia bada, biraketa txikiagoa.
2. **Abiadura lineala:** Biraketa-balioaren kosenoaren bidez kalkulatzen dugu klasean hemandako aholkoa jarraituz. Horrela, biraketa handia denean, abiadura lineala murritztu egiten da kosenoaren balio baxuarekin. Biraketa txikia denean, kosenoaren balioa handiagoa da, eta abiadura ere handiagoa da, robotak aurrera egiteko aukera handiagoa baitu. Azkenik, balioa positiboa izan dadin abs() erabiltzen da, eta abiadura gehienez 0.5 izateko biderketa egiten da, bestela kurba oso ixia egin behar zenean batzuetan talka egiten baitzuen.
3. **Segmentu librerik ez dagoenean:** Oztoporik gabeko segmenturik ez dagoenean, abiadura 0-ra ezartzen da robota gelditzeko, eta biraketa angelu handi bat (70 gradu) aplikatzen diogu, ingurua berriro eskaneatu eta noranzkoa aldatu dezan.

Abiadura eta biraketa egokitze hauei esker, robotak oztopoak saihestu eta inguruan nabigatzen du modu eraginkorrean.

- c) Zenbat denboran egon daiteke funtzionamenduan talkarik jaso gabe? (gehienez 10 min testatu)

Guk frogatzen 10 minutuz utzi dugu robota martxan eta ez du behin ere talka egin, beraz, pentsatzen dugu nahi adina egon daitekela martxan talkarik egin gabe.

- d) Denbora horretan ingurune osora mugitzen da robota edo eskualde jakin batera mugatzen da?

Denbora horretan zehar robota ingurune osora mugitzen da ondorengo irudian ikusten den moduan:



- e) Robotaren hasierako posizioa aldatuta joera berdina erakusten du?

Galdera hau erantzuteko, lehenik robotaren hasierako posizioa aldatu behar izan dugu. Horretarako, simple.launch.py fitxategiak bilatu ditugu ‘find’ komandoa erabiliz:

```
find . -name simple.launch.py
./ros2_sim/install/rosbot_gazebo/share/rosbot_gazebo/launch/simple.launch.py
./ros2_sim/src/rosbot_ros/rosbot_gazebo/launch/simple.launch.py
./build/rosbot_gazebo/launch/simple.launch.py
./install/rosbot_gazebo/share/rosbot_gazebo
```

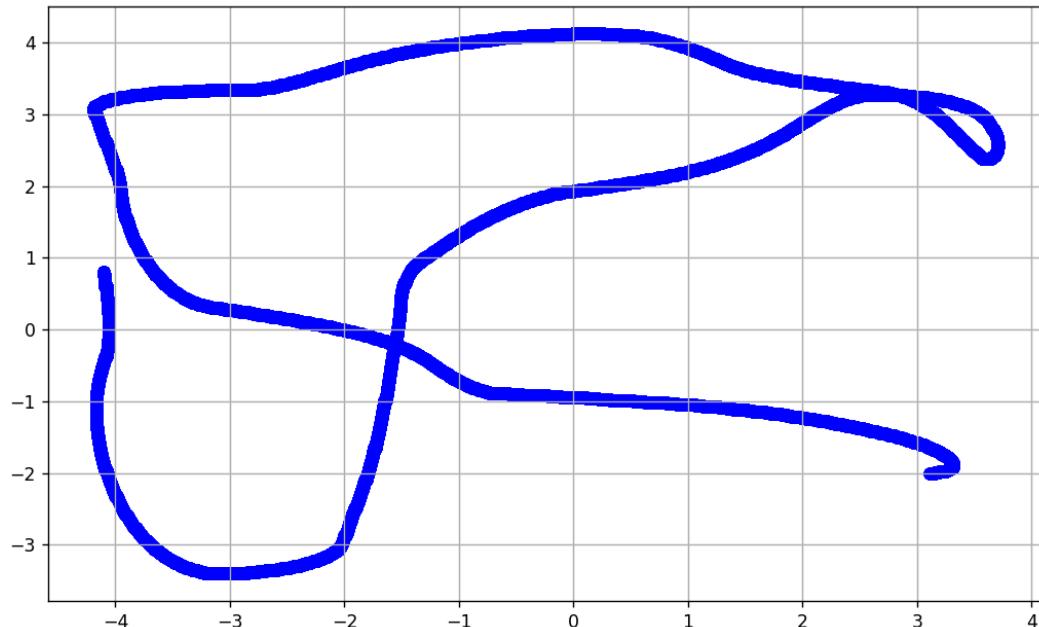
Behin fitxategiak aurkituta, ros2_sim karpetan dauden 2 fitxategietan, `launch_setup()` funtziaren x eta y aldagaien balioak aldatu ditugu -4.0, -4.0-tik 3.1 eta -2.0-ra beste leku batean has dadin robota:

```
x = LaunchConfiguration("x", default="3.1").perform(context)
y = LaunchConfiguration("y", default="-2.0").perform(context)
```

Ondoren, paketea birkonpilatu dugu ‘colcon’ komandoa erabiliz aukera hauekin:

```
colcon build --symlink-install --packages-select
rosbot_gazebo
```

Behin puntu honetara iritsita, gazebo martxan jarri dugu eta rwander exekutatu dugu 10 minutuz emaitzak aztertuz, robotak honako ibilbidea egin duela ikusteko:



Alde batetik, talkarik ez duela egin ikusi dezakegu, eta bestetik joera berdina mantentzen duela ikusi dezakegu, aurreko ibilbidean puntu horretatik aurrera, oraingoan egin duen ibilbide berdina edo oso antzekoa egin baitu.

f) **Zeintzuk dira egoera arriskutsuak?**

Robotarentzat egoera arriskutsuena ertz itxi batera iristen denean da. Izan ere, kasu horretan robota guztiz gelditu behar da eta bere ardatzarekiko biratzen hasten da bide libre bat aurkitu arte, baina ez balu aurkituko, bertan geldituko litzateke biratzen gelditu gabe. Modu guztietan, gure kasuan robotak arazo hau ez duela, edo oso gertagaitza dela uste dugu, egin ditugun saiakera guztietan arazorik atera baita egoera arriskutsuenetatik.