

Pizza banatzailea

Robotika eta Kontrol Admineduak

Jhon Díaz
Gorka Dabó

19/11/2024

Indizea

Pizza banatzailea	1
1- Sarrera:	3
2- Kontrol estrategia:	4
3- Ondorioak eta hobekuntzak:	6
4- Bibliografia:	7

1- Sarrera:

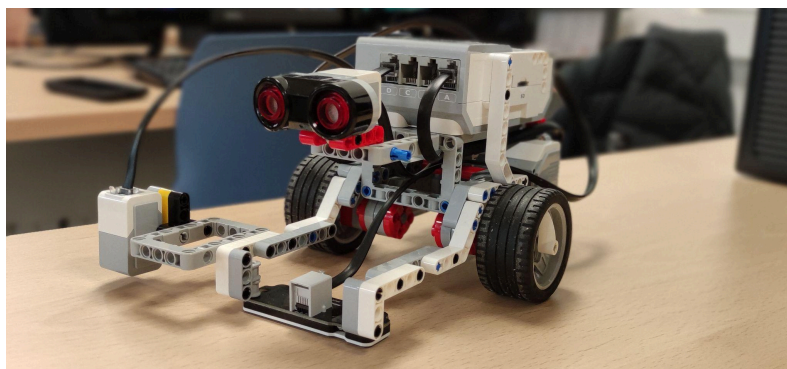
Lan honetan, **Lego Mindstorms EV3** robota erabiliz, pizza-banaketa sistema bat simulatzea eskatu zaigu. Robotak lurrean lurrean itsatsita dagoen zinta beltz bat jarraitu eta kodean definitutako ibilbidearen arabera lurrean egongo diren semaforoetan irteerak hartu beharko ditu. Adibidez, ibilbidea R2 - G1 - B3 bada, robotak lehenengo pizza kolore gorria bigarren posizioan duen semaforoaren irteeran entregatu beharko du, kolore berdea lehen posizioan duen semaforoaren irteeran, eta azkenik, kolore urdina hirugarren posizioan duen semaforoaren irteeran entregatu beharko du.

Horrez gain, robotak inguruko baldintza dinamikoak kontuan izan behar ditu, hala nola, aurrean topa ditzakeen oztopoak, adibidez, beste pizza banatzaile robotak. Sistema hau garatzeko, robotak hainbat sensore eta diseinu estrategiko integratzen ditu (1. Irudia):

- **Argi-sentsorea:** Robotaren aurrealdean, lurretik oso gertu kokatuta dago, zinta beltza erraz identifikatzeko eta ibilbidea zehaztasunez jarraitzeko.
- **Kolore-sentsorea:** Robotaren aurrealdean kokatzen da, baina erdian zentratu egon partez eskuinaldean dago, semaforoak identifikatzeko eta irteera egokiak aukeratzeko. Posizio horrek sentsorea semaforoaren kokapenarekin lerrokatuta egotea bermatzen du, informazioa modu zehatzean jasoz.
- **Ultrasoinu-sentsorea:** Robotaren aurrealdean kokatuta dago, baina altuera egokian jarri da inguruan aurki daitezkeen beste robotak edo objektuak detektatzeko eta talkak saihesteko.

Robotaren diseinu hau bere funtzionaltasunaren arabera egin da, banaketak modu egokian egiteko eta ingurunearen baldintzetara moldatzeko.

Robotaren erronkak ez dira soilik ibilbidearen jarraipenean edo semaforoaren interpretazioan oinarritzen; ingurune dinamiko batean modu egokian moldatzeko gaitasuna ere badu, hala nola, oztopoak detektatzea eta geldialdiak egitea. Horrela, proiektu honek sistema autonomo baten erronka errealak simulatzeko aukera ematen du.



1. Irudia

2- Kontrol estrategia:

Kontrol-arazoaren deskonposaketa azpi-portaeratan eta erabilitako portaeren deskribapena:

Proiektu honetan, robotaren kontrol-arazoa hainbat azpi-portaeratan deskonposatu da. Azpi-portaera bakoitzak robotaren jarduera jakin bat kontrolatzen du, erronka zehatz bati aurre egiteko. Azpi-portaera hauek egoeren bidez definitu dira, egoera bakoitzak baldintza eta erabaki jakin batzuk egingo ditu. Egoera-aldagaiak egoera bakoitzaren aldaketa eta portaera arteko trantsizioak kontrolatzeko erabiltzen dira. Ondoren, erabilitako portaerak eta horien funtzionamendua azalduko dira, sasi-kodearen bidez.

1. Marra jarraitzea:

Robotak lurrian dagoen zinta beltza jarraitu behar du zirkuito bat bezela erabiliz. Zintaren posizioa detektatzeko argi-sentsorea erabiltzen da, eta zinta jarraitzeko robotaren motorrak behar bezala doitzen dira.

Erabilitako sasi kodea hurrengoa izanda:

1. Zinta beltzaren balioak irakurri argi-sentsorearekin.
2. Batez besteko balioa kalkulatu (zintaren posizioa identifikatzeko).
3. Zinta aurkitu (balioa $<$ batez bestekoa denean).
4. Zintaren erdigunearen posizioa kalkulatu.
5. Diferentzia kalkulatu (erdigunearen eta robotaren posizioaren artean).
6. Motorren abiadura eta norabidea diferentziaren arabera egokitu.

Hasiera batean pentsatu genuen batzbestekoa kalkulatu gabe egitea baina funtzionamendua ez zen egokia eta aldatu behar izan genuen.

2. Semaforoen interpretazioa:

Robotak semaforoen kolorea irakurri eta irteera hartu edo ez erabaki behar du. Kolore-sentsorea erabiliz, RGB balioak analizatzen dira eta semaforoen sekuentziaren arabera irten edo ezerabakitzen da.

Erabilitako sasi kodea hurrengoa izanda:

1. Kolore-sentsorearen RGB balioak irakurri.
2. RGB balioen eta aurrez definitutako koloreen arteko diferentzia kalkulatu.
3. Gutxieneko diferentziaren kolorea hautatu.
4. Irteeraren kolorea eta sekuentzia egiaztatu.
5. Baldintzak betetzen badira, 'giraDerecha' aktibatu.
6. Baldintzak ez badira betetzen, ez hartu irteera.

Nahiz eta semaforoen kudeaketarekin arazorik ez izan, egia da egunaren arabera batzuetan koloreak ez zituela ondo detektatzen eta rgb.py erabiliz kalibratu egiten genituen koloreak.

3. Oztopoen detekzioa:

Robotaren aurrean dauden oztupoak detektatu eta geldialdiak egin ahal izateko ultrasoinu-sentsorea erabiltzen da. Honek bere parean gertuen duen objektuarekiko distantzia neurtzen du.

Erabilitako sasi kodea hurrengoa izanda:

1. Ultrasoinu-sentsorearen distantzia irakurri.
2. Distantzia < 30 cm bada, robotaren motorrak gelditu.
3. Distantzia 30 cm baino handiagoa bada, jarraitu aurreko egoeran.

Oztupoekin zalantza eduki genuen zenbat zentimetroko distantziara jarri, txikiegia bazen errobota kurba batean baldin bazegoen izan zitekeelako oztupoa ez ikustea, eta handiegia bazen, goizegi gelditzen zen eta beste atzeko robotak geldiarazten zituen.

Froga desberdinak egin ondoren, 30cm jartzea erabaki genuen eta ondo funtzionatu zigun.

Portaeren arteko koordinazioaren nondik norakoak:

Robotaren portaerak koordinatzeko, kontrol-estrategia orokor bat diseinatu da. Estrategia hau egoera-automataren bidez gauzatzen da, non portaerak modu hierarkikoan eta elkarrekikotasunez antolatuta dauden. Portaera bakoitzak egoera jakin batean jarduten du, eta egoera-aldaketak trantsizio-baldintzen bidez eragiten dira. Koordinazioaren helburu nagusia portaera guztiek helburu komunari –pizzak behar bezala banatzea– modu koherentean laguntzea da.

Portaeren arteko koordinazioa eta kontrol-estrategia orokorra

Robotaren portaera orokorra hiru mailatan antolatzen da:

1. **Marra jarraitzea (Line Following):** Portaera hau beti aktibo dago robotak lurrean dagoen zinta beltza jarraitu behar duenean.
2. **Semaforoen interpretazioa (Semaphore Detection):** Portaera hau aktibatzen da semaforo bat detektatzen denean. Baldintza betetzen bada, zintatik irteera egiteko aukera aktibatzen da.
3. **Oztopoen detekzioa (Obstacle Detection):** Portaera hau gainontzekoen gainetik dago, eta lehentasunez gelditzen du robotaren mugimendua talkak saihesteko.

3- Ondorioak eta hobekuntzak:

Eraikitako sistemaren portaera globalaren analisia eta hobekuntza-proposamenak honako hauek dira:

3.1. Sistemaren portaera globala

Praktika honen helburua zen ibilgailuak marra beltz bati jarraituz, semaforoak identifikatu eta pizza-banaketaren kodeari dagokion ordenan banaketa-puntuetara iristea. Sistemak, gehienetan, portaera egokia erakutsi du egoera normaletan, eta honako funtzionalitateak behar bezala bete ditu:

- Marra beltzari jarraitzea: Ibilgailuak zirkuituari jarraitzea lortu du marra beltza detektatuz.
- Semaforoen identifikazioa: Kolore-sentsorearen bidez semaforoen egoera detektatu da (urdina, berdea edo gorria) eta sekuentzia egokia ezagutu da.
- Bidegurutzeak kudeatzea eta talkak ekiditea: Ultrasoinuaren sentsorea erabiliz talkak saihestu dira, eta bidegurutzetako mugimenduak kudeatu dira.
- Kode dinamikoei egokitzea: Proba bakoitzean esleitutako kode dinamikoari jarraituz, sekuentzia behar bezala exekutatu da.

3.2. Portaera desagokiak eta arazoak

Hala ere, sistemaren portaerak zenbait egoeratan akatsak izan ditu:

1. Kolore identifikazioan arazoak: Batzuetan, argi-baldintzak edo sentsorearen kalibrazio desagokia direla eta, semaforoen koloreak behar bezala ez identifikatu dira. Horrek banaketa-sekuentziaren akatsak eragin ditu.
2. Marratik irtetea: Batzuetan, zirkuituko kurba estuetan edo marraren kolore aldaketak egon diren tokietan, ibilgailuak marra beltza galtzeko joera izan du.
3. Sentsoreen interferentziak: Ultrasoinuaren edo kolore-sentsorearen arteko interferentziak agertu dira noizean behin, bereziki zirkuituaren hasiera-anbiguotasunean.

3.3. Hobekuntzak

Denbora gehiago eta baliabide gehiago izanez gero, sistema hobetu ahal izango litzateke honako modu hauetan:

1. Kolore-sentsorearen kalibrazio dinamikoa: Ingurunearen argi-baldintzetara automatikoki egokitzen den kalibrazio-mekanismo bat gehitu daiteke, koloreen detekzioa fidagarriagoa izan dadin.
2. Marra beltzaren jarraipena optimizatzea: Marraren jarraipen algoritmoa gehiago fintzea, adibidez, PID kontrol sistema bat erabiliz, marratik irtetea saihesteko.
3. Zirkuituaren mapa aurrez kargatzea: Egungo metodoa hobetu liteke aurrez kargatutako mapa edo bidearen modeloa erabiliz, mugimenduen zehaztasuna handitzeko.
4. Talken aurkako hobekuntzak: Ultrasoinuaren sentsorea beste sentsore batzuekin osatzea, adibidez infragorria edo kamerak, talken aurkako erantzun fidagarriagoak lortzeko.
5. Hardwarearen hobekuntza: Ibilgailuaren hardwarea optimizatzea, adibidez, gurpilak eta motorak hobetuz, mugimendua zehatzago eta azkarrago izateko.
6. Kodearen dokumentazioa eta modularitatea: Kode gehiago modularizatuz eta behar bezala dokumentatuz, mantentze-lanak eta hedapenak errazteko.

3.4. Talde lanaren esperientzia

Talde lanari dagokionez, honako ondorioak aipatubehar dira:

1. Lanaren banaketa egokia izan da, eta talde bakoitzak bere ardurak behar bezala bete ditu.
2. Taldekideen arteko komunikazioa funtsezkoa izan da arazoak garaiz identifikatzeko eta konpontzeko.
3. Denboraren kudeaketa hobetzea komenigarria izango litzateke, azken uneko estresa ekiditeko eta hobekuntza-proposamenak praktikan jartzeko denbora gehiago izateko.

Ondorioz, sistemak funtsezko helburuak bete ditu, eta garapen- eta hobekuntza-aukera handiak aurkitu dira etorkizuneko lanetan sistema fidagarriagoa eta eraginkorragoa izan dadin.

4- Bibliografia:

[1] Python-en hari kudeaketa: <https://egela.ehu.eus/mod/resource/view.php?id=7773093>

[2] PID Kudeaketa: <https://egela.ehu.eus/mod/resource/view.php?id=7773026>

[3] Enuntziatua: <https://egela.ehu.eus/mod/resource/view.php?id=7773089>