

Pertsonen jarraipenaren ebaluazioa robot mugikor batean

Karrera Bukaerako Proiektua

2014ko ekainaren 25a

Bingen Gonzalez Oiarbide

Gainbegiralea:

Elena Lazkano Ortega

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

*Teorian ez dago ezberdintasunik
teoria eta praktiraren artean.
Praktikan bai.*

Jatorri ezezaguna.

Eskertza

Lerro hauetan proiektua gauzatzen lagundu didaten guztioi nire esker ona adierazi nahi diet, seguru nago baten bat ahaztuko zaidala.

IK4-TEKNIKER-eko sistema adimendunen unitatekoei, proiektua bertan egiteko aukera eskaintzeagatik, beharrezko material guztia eskura uzteagatik eta eskaintako laguntzagatik.

Ander Ansuateguiri nirekin izandako pazientziagatik eta irakatsi didan guztiagatik.

Carlos, Aitor, Ane eta Anderri egunero Eibarrera egin beharreko bidaia lanerako soilik izan ez delako :)

Eskerrik asko denoi!

Gaien Aurkibidea

1. Proiektuaren Helburu Dokumentua	1
1.1. Proiektuaren irismena	2
1.1.1. Eskakizun minimoak	2
1.1.2. Proiektuaren murriztapenak	2
1.1.3. LDE diagrama	3
1.1.4. Atazen deskribapena	3
1.2. Plangintza	6
1.2.1. Atazen estimazioa	6
1.2.2. Arriskuen analisisa	7
2. Proiektuaren jarraipena	11
2.1. Sarrera	12
2.2. Burututako atazen eta estimazioen alderaketa	12
2.3. Desbideratzeen zergatia eta izandako arazoak	13
3. Robot mugikorren nabigazioa	15
3.1. Sarrera	17
3.2. Robot mugikorren nabigaziorako eskema posible bat	18
3.3. Monte-Carlo metodoa	19
3.3.1. Sarrera	19
3.3.2. Monte-Carloren lokalizazioa	20
3.4. Kalmanen filtroak	24

3.4.1.	Sarrera	24
3.4.2.	Algorimoa	25
3.4.3.	Jarraipena Kalmanen filtroak erabiliz	27
3.5.	Partikula-filtroak	29
3.5.1.	Sarrera	29
3.5.2.	Kalmanen filtroarekin ezberdintasunak	30
3.5.3.	Algoritmoa	31
3.5.4.	Pisuen degenerazioa	33
3.5.5.	Garrantzi funtzioaren aukeraketa	34
3.5.6.	Birlagintzea (<i>Resampling</i>)	35
3.5.7.	Ondorioak	35
3.6.	Mugimenduen planifikatzaileak	37
3.6.1.	Sarrera	37
3.6.2.	Programazio dinamikoa	38
3.6.3.	A*	39
3.6.4.	Konparaketa	42
3.6.5.	Ondorioak	43
3.7.	Robotaren mugimendua	43
3.7.1.	Sarrera	43
3.7.2.	Leuntzea	44
3.7.3.	PID kontrola	44
4.	Pertsonen jarraipena robot mugikor batean	47
4.1.	Sarrera	48
4.2.	Robotaren Egitura	49
4.2.1.	Sentsoreen ezaugarriak	49
4.2.2.	Hanken Detekzioa	52
4.2.3.	Txalekoaren Detekzioa	54
4.2.4.	Detekzio Termikoa	55
4.2.5.	Fusioa Partikula-Filtroa erabiliz	56
4.2.6.	Pertsonen jarraipenerako partikula-filtroa	58
4.2.7.	Kontrol modulua	59

<i>Gaien Aurkibidea</i>	VII
5. Esperimentazioa	61
5.1. Sarrera	62
5.2. Proba 1	62
5.2.1. Helburua	62
5.2.2. Ingurunea	63
5.2.3. Ondorioak	63
5.3. Proba 2	67
5.3.1. Helburua	67
5.3.2. Ingurunea	67
5.3.3. Jasotako datuen azterketa	68
5.3.4. Robotaren abiadura ezberdinekin probak	69
5.3.5. Partikula-Filtroaren partikula kopuru ezberdinekin probak	75
5.3.6. Sentsoreetan pisu ezberdinekin probak	81
5.3.7. Zirkuitu borobil batean probak	90
5.3.8. Ondorioak	95
5.4. Proba 3	97
5.4.1. Helburua	97
5.4.2. Ingurunea	97
5.4.3. Galdetegia	97
5.4.4. Emaitzak	99
5.4.5. Ondorioak	101
6. Ondorioak eta etorkizuneko lana	103
6.1. Sarrera	104
6.2. Nabigazioa robot mugikorretan	104
6.3. Pertsonen jarraipena robot mugikorretan	104
6.4. Etorkizuneko lana	105
Bibliografia	107

Irudien Zerrenda

1.1. Proiektuaren LDE diagrama.	3
3.1. Robot mugikor baten nabigazio eskema.	19
3.2. Monte-Carlo lokalizazioa, hasierako egoera.	21
3.3. Monte-Carlo lokalizazioa, atea ikusi ondorengo egoera.	21
3.4. Monte-Carlo lokalizazioa, robotak mugitu eta geroko egoera.	22
3.5. Monte-Carlo lokalizazioa, bigarren atea detektatu eta geroko egoera.	22
3.6. Monte-Carloren lokalizazioa, robotak bere burua kokatu du.	23
3.7. Monte-Carloren lokalizazioa, algoritmoa.	24
3.8. Kalmanen filtroa, algoritmoa.	27
3.9. Kalmanen filtroak, objektu mugikorraren posizioak.	28
3.10. Kalmanen filtroak, objektu mugikorraren posizioaren estimazioa.	28
3.11. Kalmanen filtroak, objektuaren mugimendua geziekin.	29
3.12. Partikula-filtroak eta birlaginketaren funtzionamendua.	36
3.13. Robot batek egin ditzakeen mugimenduak, gelaxka batetatik ondoan duen beste batetara joateko.	38
3.14. Programazio dinamikoa oztoporik gabeko ingurune batean	42
3.15. Programazio dinamikoa oztopoa dagoen ingurune batean	42
3.16. A* algoritmoa oztoporik gabeko ingurune batean	43
3.17. A* algoritmoa oztopoa dagoen ingurune batean	43
3.18. Leuntze metodoa	44
3.19. Kontrol proportzional integral deribatiboaren bloke-diagrama	46

4.1. Robotaren egitura, hiru datu sarrera eta erabakiak hartzeko partikula-filtroa	50
4.2. Robotaren egitura, Segway RMP-a sentsoreekin	51
4.3. Robotaren egitura, HTPA sentsore termikoa	51
4.4. Robotaren egitura, Kinect sentsorea	52
4.5. Robotaren egitura, Hokuyo laser sentsorea	52
4.6. Hankak detektatzeko erabiltzen den patroia	53
4.7. Laserra hankak detektatzen	53
4.8. Irudia filtratu eta gero lortzen den irudi bitarra	54
4.9. Zentroidea kalkulatu eta gero lortzen den irudia	54
4.10. Irudia filtratu eta gero lortzen den irudi bitarra, beste posizio batean	55
4.11. Zentroidea kalkulatu eta gero lortzen den irudia, beste posizio batean	55
4.12. Pertsonen jarraipena egiteko algoritmoa, partikula-filtroa erabiliz. .	56
4.13. Koordenatu sistema zilindrikoa	59
4.14. Koordenatu sistema polarra	59
5.1. $t=1$	64
5.2. $t=2$	65
5.3. $t=3$	65
5.4. $t=4$	66
5.5. $t=5$	66
5.6. Zirkuitu zuzena	67
5.7. Zirkuitu borobila	67
5.8. Angeluaren konparaketa abiadura 0.35m/s denean.	70
5.9. Angeluaren konparaketa abiadura 0.6m/s denean.	71
5.10. Angeluaren konparaketa abiadura 1.2m/s denean.	71
5.11. 3 abiadurek angeluaren estimazioan izandako errorearen konparaketa.	72
5.12. Distantziaren konparaketa abiadura 0.35m/s denean.	73
5.13. Distantziaren konparaketa abiadura 0.6m/s denean.	73

5.14. Distantziaren konparaketa abiadura 1.2m/s denean.	74
5.15. 3 abiadurek distantziaren estimazioan izandako errorearen konparaketa.	74
5.16. Angeluaren konparaketa partikula kopurua 25 denean.	76
5.17. Angeluaren konparaketa partikula kopurua 50 denean.	77
5.18. Angeluaren konparaketa partikula kopurua 100 denean.	77
5.19. Partikula kopuru ezberdinek angeluaren estimazioan izandako errorearen konparaketa.	78
5.20. Distantziaren konparaketa partikula kopurua 25 denean.	79
5.21. Distantziaren konparaketa partikula kopurua 50 denean.	79
5.22. Distantziaren konparaketa partikula kopurua 100 denean.	80
5.23. Partikula kopuru ezberdinek distantziaren estimazioan izandako errorearen konparaketa.	80
5.24. Angeluaren konparaketa Txalekoa=1, Termopila=0 eta Hanka=0 denean.	83
5.25. Angeluaren konparaketa Txalekoa=0.6, Termopila=0.2 eta Hanka=0.2 denean.	83
5.26. Angeluaren konparaketa Txalekoa=0.2, Termopila=0.6 eta Hanka=0.2 denean.	84
5.27. Angeluaren konparaketa Txalekoa=0.2, Termopila=0.2 eta Hanka=0.6 denean.	84
5.28. Angeluaren konparaketa Txalekoa=0.3, Termopila=0.3 eta Hanka=0.3 denean.	85
5.29. Partikula kopuru ezberdinek angeluaren estimazioan izandako errorearen konparaketa.	85
5.30. Distantziaren konparaketa Txalekoa=1, Termopila=0 eta Hanka=0 denean.	87
5.31. Distantziaren konparaketa Txalekoa=0.6, Termopila=0.2 eta Hanka=0.2 denean.	87
5.32. Distantziaren konparaketa Txalekoa=0.2, Termopila=0.6 eta Hanka=0.2 denean.	88
5.33. Distantziaren konparaketa Txalekoa=0.2, Termopila=0.2 eta Hanka=0.6 denean.	88

5.34. Distantziaren konparaketa Txalekoa=0.3, Termopila=0.3 eta Hanka=0.3 denean.	89
5.35. Partikula kopuru ezberdinek distantziaren estimazioan izandako errorearen konparaketa.	89
5.36. Zirkuitu borobilean kalkulaturako angeluaren eta benetako balioen arteko erlazioa, txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasuan.	92
5.37. Zirkuitu borobilean kalkulaturako angeluaren eta benetako balioen arteko erlazioa, txalekoa=0.3, termopila=0.3 eta hanka=0.3 kasuan.	92
5.38. Zirkuitu borobilean angeluaren estimazioan izandako errorearen konparaketa.	93
5.39. Zirkuitu borobilean kalkulaturako distantziaren eta benetako balioen arteko erlazioa, txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasuan.	94
5.40. Zirkuitu borobilean kalkulaturako distantziaren eta benetako balioen arteko erlazioa, txalekoa=0.3, termopila=0.3 eta hanka=0.3 kasuan.	94
5.41. Zirkuitu borobilean distantziaren estimazioan izandako errorearen konparaketa.	95

Taulen Zerrenda

1.1. Atazen estimazioen taula.	6
1.2. Arriskuen eskala taula.	7
2.1. Proiektuaren jarraipen taula.	12
5.1. Abiadura ezberdinen erroreak gradutan angeluarekiko.	70
5.2. Abiadura ezberdinen erroreak metrotan distantziarekiko.	72
5.3. Partikula kopuru ezberdinekin lortutako erroreak gradutan angeluarekiko.	76
5.4. Partikula kopuru ezberdinen izandako erroreak metrotan distantziarekiko.	78
5.5. Detekzioei balio ezberdinak emanda lortutako erroreak gradutan angeluarekiko.	82
5.6. Sentsoreei balio ezberdinak emanez lortutako erroreak metrotan distantziarekiko.	86
5.7. Zirkuitu borobilean lortutako erroreak gradutan angeluarekiko.	91
5.8. Zirkuitu borobilean lortutako erroreak metrotan distantziarekiko.	93
5.9. Proba egin dutenek erantzundako galdetegia.	98
5.10. Jarraipenaren sentsazioa.	99
5.11. Jarraipenaren erosotasuna.	99
5.12. Robotaren distantzia.	100
5.13. Robotaren abiadura.	100
5.14. Espektatibaren inguruko galdera.	100

Laburpena

Robot mugikorrek gure gizarteari ekar diezaieketen onura eta erosotasunez ohar-turik, ari gara hauek pixkanaka gure egunerokoan txertatzen; etxeko lanetan, lan industrialetan edo erreskate lanetan lagun gaitzaten.

Memoria honetan aurkeztutako lanak robot mugikorren nabigazioaren ikuspegi teorikoa landu nahi du. Robot batek bere burua lokalizatu, eta ondoren, leku batetik beste batera autonomoki mugitzeko erabiltzen diren teknikak deskribatuz.

Honetaz gain, sentso ezberdinetatik jasotako informazioa fusionatuz, pertso-nen jarraipena egiten duen robot baten portaera aztertu eta ebaluatu da, honen errendimendua hobetu asmoz.

1. Kapituluia

Proiektuaren Helburu Dokumentua

Gaien Aurkibidea

1.1. Proiektuaren irismena	2
1.1.1. Eskakizun minimoak	2
1.1.2. Proiektuaren murriztapenak	2
1.1.3. LDE diagrama	3
1.1.4. Atazen deskribapena	3
1.2. Plangintza	6
1.2.1. Atazen estimazioa	6
1.2.2. Arriskuen analisisa	7

1.1. Proiektuaren irismena

Proiektu honen helburua esperimentu ezberdinak egitea da, sentsoreen fusioan oinarritutako pertsonen detekzioa egiten duen plataforma robotiko mugikor baten portaera ebaluatzeko.

Hau egiteko, IK4-TEKNIKER enpresan garatutako robot mugikor bat erabiliko da. Pertsonen detekzio eta jarraipena egiten duen sistema honek informazio iturri ezberdinak erabiltzen ditu: laser sentsore bat, ikusmen sentsore bat eta sentsore termiko bat, hain zuzen ere. Hiru sentsore horiek ematen duten informazioa fusionatzeko partikula-filtro bat erabiltzen da.

Egin beharrekoa, hiru informazio iturri horiek ahalik eta egokien konbinatzea da, robotaren pertsona detekzio eta jarraipen portaera hobetzeko.

1.1.1. Eskakizun minimoak

Proiektua burutzeko jarraian adierazitako lanak egin beharko dira:

- Partikula-filtroen funtzionamenduen inguruko analisi eta ikasketa.
- Esperimentuen definizioa.
- Emaitzen azterketa eta ebaluazioa.

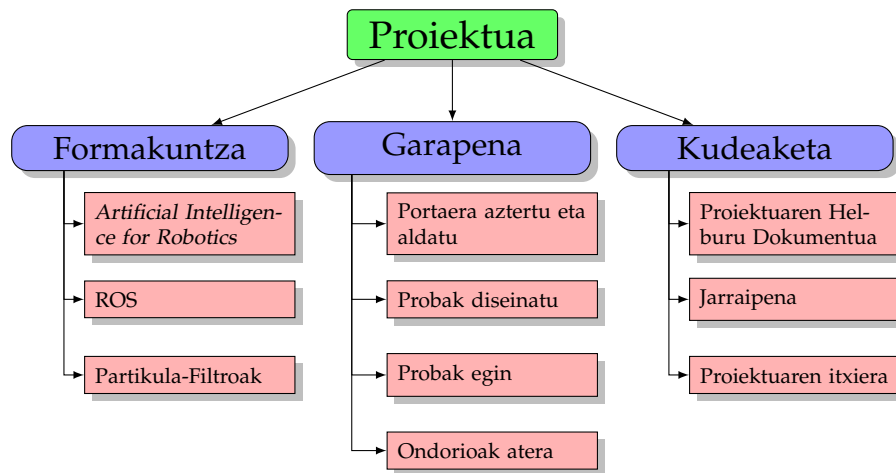
1.1.2. Proiektuaren murriztapenak

Proiektu honetan dauden murriztapenak hauek dira:

- Orduei dagokienez, gehienez 375 ordu erabili ahal izango dira. Hau da, 15 kreditu ditu karrera amaierako proiektuak eta kreditu bakoitza 25 ordutan hitzartuta dagoenez, aipatutako orduak erabil daitezke gehienez.
- Datei dagokienez, berriz, proiektuaren hasiera 2014ko urtarrilaren 12a izango da eta amaiera 2014ko ekainaren 30a.

1.1.3. LDE diagrama

Jarraian dagoen irudian, ikus 1.1. irudia, proiektuaren Lanaren Deskonposaketa Diagrama ikus daiteke. Bertan, adierazten den moduan, proiektua hiru atal nagusitan banatzen da: formakuntza, garapena eta kudeaketa.



1.1. Irudia: Proiektuaren LDE diagrama.

1.1.4. Atazen deskribapena

Jarraian aurreko atalean marrazturiko LDE diagraman finkatu diren ataza desberdinen deskribapena zehaztuko da.

Formakuntza

Ataza honetan proiektuan zehar burutuko den formakuntza biltzen da. Honek proiektuaren zati handi bat bere gain hartuko du, bertan jorratuko diren gaiak ez baitira gehiegi sakondu Ingeniaritza Informatikoa titulazioaren irakasgai ezberdinetan. Hiru azpiatalez osatzen da.

- **Artificial Intelligence for Robotics:** Ataza hau, atazaren izena duen eta www.udacity.com webguneak eskaintzen duen doako ikastaroa burutzean datza. Ikastaroaren arduraduna *Sebastian Thrun Stanford* Unibertsitateko irakaslea eta *Google*-ren auto autonomoaren bultzatzaileetako bat dena da. Ikastaroan gai hauek landuko dira:

- Lokalizazioa Monte-Carlo metodoak erabiliz.

- Detekzioa Kalman Filtroak erabiliz.
 - Lokalizazioa Partikula-Filtroak erabiliz.
 - Planifikazioa eta bilaketa, heuristiko eta A* algoritmoarekin.
 - PID (*Proportional Integral Derivative*) kontrola.
- **ROS:** Ataza honetan ROS-en funtzionamendua eta erabilera ikasiko dira. ROS (*Robotic Operating System*) robotentzako softwarea garatzeko erabiltzen den framework bat da. Proiektuan erabiliko den robotak hau erabiltzen du, beraz robota maneiatu ahal izateko hau menperatzea ezinbestekoa izango da. Ikasketa prozesuan gehienbat wiki.ros.org webgunea erabiliko da, bertan informazio asko baitago. Baina, horrez gain, *ROS by Example* [3] eta *Learning ROS for Robotics Programming* [5] liburuak ere erabiliko dira.
 - **Partikula-Filtroak:** Hemen partikula-filtroen inguruko ezagutza zabalduko da. *Artificial Intelligence for Robotics* ikastaroan lehen urratsak emango dira baina ikastaro horretan partikula-filtroen erabilera lokalizazioari zuzenduta dago. Praktikarako erabiliko den robotean ordea, partikula-filtroak sentso-reen informazioa fusionatzeko erabiltzen da eta, ondorioz, gehiago sakondu beharko da gai honetan sarean dagoen informazioa erabiliz.

Garapena

Ataza honetan proiektuaren zati mardulena biltzen da. IK4-TEKNIKER teknologia zentroak garatutako robot mugikorrarekin lehen kontaktua edukiko da eta, formakuntza ondo barneratu dela aurreikusiz, proiektuaren garapenari ekingo zaio. Lau fasetan ezberdindu da garapena, baina lau faseak oso lotuta daude. Lehenik eta behin **robotaren portaera aztertu eta aldatu** ataza dago, honen atzetik **probak diseinatu**, ondoren, **probak egin** eta, azkenik, **ondorioak atera**. Honek ez du esan nahi ataza bakoitza behin egingo denik, laurak behin eta berriz egin beharko dira. Hau da, robotean egindako aldaketek zein ondorio izan dituzten ikusteko probak diseinatu eta egin beharko dira. Ondorioen arabera aldaketa berriak egin beharko dira robotean eta honek proba berrien diseinu eta exekuzioa dakar berarekin. Ikus daitekeen moduan, atalak era ziklikoan behin eta berriz egin beharko dira ondorio interesgarriak eskuratu bitartean.

Atal honetan egiten den guztia ondo dokumentatu behar da memoriaren zati handi eta garrantzitsuena bertan idatziko baita.

- **Portaera aztertu eta aldatu:** Atal honetan robotarekin lehen harremana edukiko da. Lehenik eta behin, bere kodea aztertu, ulertu eta robotak kode horri nola erantzuten duen aztertuko da, azken finean kode hori izango da maneiatu beharrekoa, beraz oso garrantzitsua da hau pazientziaz egitea. Ondoren, portaera hobetzeko egin beharreko aldaketak egingo dira.

- **Probak diseinatu:** Robotean egindako aldaketen erantzuna aztertzeke egin beharreko probak diseinatuko dira.
- **Probak egin:** Ataza honetan aurreko atazan diseinatutako probak martxan ipiniko dira eta egin diren aldaketek portaeran izandako eragina aztertuko da.
- **Ondorioak atera:** Egindako probek izan duten emaitza aztertuko da eta aurretik egindako probekin alderatu, portaerak hobera edo okerrera egin duen ondorioztatzeko.

Kudeaketa

Ataza honetan proiektuak kudeaketa egokia izateko behar diren atalak bereiztu dira. Hiru atal, hain zuzen ere:

- **Proiektuaren helburu dokumentua:** Ataza honetan proiektuaren eskakizunak aztertuko dira, hauetatik abiatuz irismena finkatzeko. Irismena finkatu eta gero, dauden murriztapenak aztertuko dira eta proiektua ataza ezberdinetan banatuko da LDE diagrama bat eginez. Ondoren, definitu diren atazak burutzeko beharko den denboraren estimazioa egingo da. Proiektuan zehar eduki daitezkeen arriskuen analisisa ere egingo da, arazoan aurrean erantzunak edukitzeko.
- **Jarraipena:** Ataza honetan proiektuaren jarraipena eta kontrola egingo da. Horretarako, ataza bakoitza burutzeko behar izan den denbora kontutan hartuko da eta **atazen estimazioa** atalean estimatu denarekin alderatu, desbideratzeak ikusi eta zergatia aztertzeke, proiektuan zehar izandako arazoak aipatuz.
- **Proiektuaren itxiera:** Ataza honen bitartez proiektuari amaiera emango zaio, entregatu behar den dokumentazioari azken ukituak emango zaizkio, txukundu egingo da eta, azkenik, entregatu. Proiektuaren murriztapenak atalean definitu da noiz den proiektua entregatzeko azken eguna.

1.2. Plangintza

Atal honetan proiektuaren atazen estimazioa eta proiektuan lanean ari garela egon daitezkeen arriskuen analisisa egin da.

1.2.1. Atazen estimazioa

1.1. taulan, aurretik deskribatu diren atazen estimazioa egin da. Taulan ataza bakoitzaren hasiera eta amaiera datak eta ataza hori burutzeko estimatu den esfortzua ordutan ikus daiteke.

Deskribapena	Hasiera	Bukaera	Esfortzua
Formakuntza	2014/1/20	2014/3/17	120 ordu
-Artificial Intelligence for Robotics	2014/1/20	2014/2/17	60 ordu
-ROS	2014/2/17	2014/3/3	30 ordu
-Partikula filtroak	2014/3/3	2014/3/17	30 ordu
Garapena	2014/3/17	2014/6/16	200 ordu
-Portaera aztertu eta aldatu	2014/3/17	2014/6/9	50 ordu
-Probak diseinatu	2014/3/23	2014/6/9	50 ordu
-Probak egin	2014/3/23	2014/6/9	50 ordu
-Ondorioak	2014/4/1	2014/6/16	50 ordu
Kudeaketa	2014/1/12	2014/6/30	30 ordu
-Plangintza	2014/1/12	2014/1/20	10 ordu
-Jarraipena	2014/1/20	2014/6/16	10 ordu
-Proiektuaren itxiera	2014/6/16	2014/6/30	10 ordu
Guztira:	2014/1/12	2014/6/30	350 ordu

1.1. Taula: Atazen estimazioen taula.

Taula begiratuta ikus daiteke guztira 350 ordu estimatu direla proiektua bukatzeko. Baina murriztapenetan adierazi moduan, 375 da erabili daitezkeen ordu kopuru maximoa. **Estimazioan erabili ez diren 25 orduak larrialdi orduak dira.** Proiektuan zehar ataza batzuetan estimatu baino denbora gehiago behar izanez gero, ordu hauek erabiliko dira murriztapenetan ezarritako ordu muga ez gainditzeko.

1.2.2. Arriskuen analisia

Atal honetan proiektu hau aurrera eramateko gerta daitezkeen arriskuak identifikatu dira eta beraiei aurre egiteko plana zehaztu.

Arrisku gisara identifikatu dira, gertatuz gero, proiektuaren irismenean atzerapena eragin dezaketen ekintzak. Ekintza hauen aurkako kontingentzia plan egokia ezinbestekoa da gertatuz gero proiektuan ahalik eta eragin txikiena izateko.

Arriskuak bi taldetan multzokatu dira, arrisku teknikoak eta arrisku pertsonalak. Guztia ulergarriagoa egiteko, gertatzeko aukeren eta larritasun mailaren eskalak definitu dira, ikusi 1.2. taula.

Gertatzeko aukera	Oso txikia	Txikia	Ertaina	Handia	Oso handia
Larritasun maila	Oso txikia	Txikia	Ertaina	Handia	Oso handia

1.2. Taula: Arriskuen eskala taula.

Arrisku teknikoak

- Irismena ez betetzeko arriskua:
 - **Deskribapena:** Irismenean finkatutako helburuak ez betetzea.
 - **Gertatzeko aukera:** Txikia.
 - **Larritasun maila:** Oso handia.
 - **Eragina:** Proiektua bertan behera gera daiteke, ez baita adostutakoa bete.
 - **Aurrezaintza plana:** Irismenean finkatutako puntu guztiak burutzeko modukoak direla konprobatu behar da.
 - **Erantzun ekintza:** Proiektuaren zuzendariarekin hitz egin eta berak hartutako erabakien arabera jokatu da.
- Teknologiaren erabilerarekin arazoak:
 - **Deskribapena:** Proiektuan erabili behar den teknologia ez da ezagutzen eta, formakuntza egin eta gero, ez da espero zen ezagutza maila lortu.
 - **Gertatzeko aukera:** Ertaina.
 - **Larritasun maila:** Handia.

- **Eragina:** Proiektua atzeratu dezake eta honek proiektuaren amaieradarekin arazoak izateko arriskua sortzen du.
- **Aurrezaintza plana:** Formakuntza atalean arduraz jokatu behar da, eta kontzeptuak ondo ulertu direla ziurtatu.
- **Erantzun ekintza:** Formakuntzarako estimatu den denboraz gain, larrialdietarako gorde diren orduak ere formakuntzarako erabiliko dira.
- Informazio edo datu galera:
 - **Deskribapena:** Proiektuaren datuak edo informazioa galtzen da. Arrazoi ezberdinengatik izan daiteke, ordenagailuan arazoak izan direlako edo datuak gordetzeko erabilitako dispositiboan arazoak izan direlako edo beste edozein arazorengatik.
 - **Gertatzeko aukera:** Ertaina.
 - **Larritasun maila:** Oso Handia.
 - **Eragina:** Proiektuaren datuak galduko dira eta, galdutako informazioaren arabera, proiektuan atzerapen handiagoa eragingo du, lan batzuk berriz egin beharko dira, eta honek estimazioetan arazoak sortuko ditu.
 - **Aurrezaintza plana:** Proiektuaren datu eta informazioa leku batean baino gehiagotan gorde behar da eta segurtasun kopiak egin behar dira. Informazioa Ubuntu eskaintzen duen Ubuntu One lainoan gordeko da, hau egiteko nahikoa da Ubuntu makina batetik proiektuaren informazio eta datuak gordetzen dituzten karpetak Ubuntu Onerekin sinkronizatzea. Hau eginez gero, karpeta horietan gordetakoaren kopia automatikoki gordeko da sarean eta, gainera, edozein ordenagailutatik atzigarri egongo da. Ubuntu One erabiltzeaz gain, segurtasun kopiak egingo dira astero USB batean.
 - **Erantzun ekintza:** Informazioa berreskuratu, lehenik Ubuntu Onen begiratu eta, bertatik, lanen azken bertsioak hartu. Edozein arazorengatik Ubuntu Onen gordetako datuak ezin baditugu berreskuratu, azken segurtasun kopia erabiliko da.
- Robotean arazoak:
 - **Deskribapena:** IK4-TEKNIKER enpresan garatutako robotak arazoak ditu, hardware edo software arazoak baina proiektuarekin zer ikusirik ez dutenak.
 - **Gertatzeko aukera:** Txikia.
 - **Larritasun maila:** Handia.
 - **Eragina:** Proiektuan atzerapenak eragingo ditu eta, dituen arazoen arabera, atzerapena handiagoa izango da.

- **Aurrezaintza plana:** Robota ondo zaindu behar da eta proiekturen helburuetarako erabili, ez jolasteko edo proiektutik at dauden gauzak egiteko.
- **Erantzun ekintza:** Lehenik IK4-TEKNIKER enpresako kideekin hitz egin eta konpontzeko beharko den denbora zehaztu, arazoa larria bada proiektuaren zuzendariarekin hitz egin eta, proiektuan aurkitzen garen fasearen arabera, erabaki bat adostu berarekin.

Arrisku pertsonalak

- Estimaturako epeak ez betetzea:
 - **Deskribapena:** Proiektuaren plangintza egitean estimazioak gaizki egin dira. Hau arazo edo abantaila izan daiteke, gerta baitaiteke estimatu baino denbora gehiago erabiltzea ataza bat burutzen edo, bestalde, estimatu baino gutxiago.
 - **Gertatzeko aukera:** Handia.
 - **Larritasun maila:** Ertaina.
 - **Eragina:** Estimatu baino denbora gehiago behar izan bada, eragina negatiboa izango da, eta honek proiektuan atzerapenak eragingo ditu. Estimatu baino denbora gutxiago behar izan bada, berriz, eragina positiboa izango da eta proiektua estimatu baino lehenago amaitzeko aukera izango da.
 - **Aurrezaintza plana:** Plangintza egitean estimazioak ahalik eta zuzenen egin behar dira, ausnartzeko denbora hartuz, eta errealitatean behar izango den denboran pentsatuz.
 - **Erantzun ekintza:** Estimazioa labur egin den kasuetan larrialdietarako denbora erabiliko da ahal den neurrian, denbora horrekin ez bada nahikoa proiektua birplanifikatu beharko da estimazioak berriz eginez. Estimazioa handiegia egin den kasuetan, berriz, soberan geratu den denbora larrialdi denborari gehituko zaio, beharrezkoa bada proiektuaren beste atal batean erabili ahal izateko.
- Behin behineko baja:
 - **Deskribapena:** Arrazoi ezberdinak direla medio egun bat edo gehiagotan proiektua geldo egongo da. Larritasun maila proiektua geldo dagoen denborarekin batera esponentzialki handituko da.
 - **Gertatzeko aukera:** Ertaina.
 - **Larritasun maila:** Ertaina.

- **Eragina:** Proiektuan atzerapenak eragingo ditu.
- **Aurrezaintza plana:** Ez dago aurrezaintza plan garbirik, aurreikusi ezin diren faktore asko baitaude kasu honetan.
- **Erantzun ekintza:** Berriz ere lanari ekitean galdutako denborak proiektuan eragin duen atzerapena aztertu beharko da eta atzerapen horren arabera erabakiak hartu. Atzerapena oso handia izan bada, zuzendariekin hitz egin eta proiektua birplanifikatu beharko da. Atzerapena txikia bada, berriz, egun batzuetan esfortzua handitu beharko da, ordu gehiago lan eginez plangintzarekin egunean ipintzeko.

2. Kapituluia

Proiektuaren jarraipena

Gaien Aurkibidea

2.1. Sarrera	12
2.2. Burututako atazen eta estimazioen alderaketa	12
2.3. Desbideratzeen zergatia eta izandako arazoak	13

2.1. Sarrera

Atal honetan proiektuak bere garapenean izan duen norabidea azaltzen da, eta horretarako egindako estimazioak eta benetan behar izandako denbora alderatzen dira. Alderaketa horretatik sortu den desbideratzea azaltzen da, desbideratzea arrazoituz eta desbideratzea sortu duten arazoak deskribatuz.

2.2. Burututako atazen eta estimazioen alderaketa

2.1. taulan, estimatutako esfortzua eta esfortzu errealak alderatu dira, izandako desbideratzea hobeto ulertzeko. Taulan atazen deskribapena, benetan noiz hasi eta noiz amaitu diren atazak, plangintzan estimatu den denbora, benetan behar izan den denbora, eta desbideratzea ikus daitezke. Koloreak ulermena errazteko erabili dira:

Estimatu baina denbora **gehiago** behar izan da: **Gehiago**

Estimatu baina denbora **gutxiago** behar izan da: **Gutxiago**

Estimatutako denbora behar izan da: **Berdina**

Deskribapena	Hasiera	Bukaera	Esfortzu estimatua	Esfortzu errealak	Desbideratzea
Formakuntza	2014/1/20	2014/4/2	120 ordu	150 ordu	30 ordu
-Artificial Intelligence for Robotics	2014/1/20	2014/2/24	60 ordu	70 ordu	10 ordu
-ROS	2014/2/24	2014/3/20	30 ordu	40 ordu	10 ordu
-Partikula filtroak	2014/3/3	2014/4/2	30 ordu	40 ordu	10 ordu
Garapena	2014/4/3	2014/6/16	200 ordu	180 ordu	20 ordu
-Portaera aztertu eta aldatu	2014/4/3	2014/6/9	50 ordu	40 ordu	10 ordu
-Probak diseinatu	2014/4/21	2014/6/9	50 ordu	30 ordu	20 ordu
-Probak egin	2014/4/21	2014/6/9	50 ordu	60 ordu	10 ordu
-Ondorioak	2014/5/5	2014/6/16	50 ordu	50 ordu	0 ordu
Kudeaketa	2014/1/12	2014/6/30	30 ordu	30 ordu	0 ordu
-Plangintza	2014/1/12	2014/1/20	10 ordu	10 ordu	0 ordu
-Jarraipena	2014/1/20	2014/6/16	10 ordu	10 ordu	0 ordu
-Proiektuaren itxiera	2014/6/16	2014/6/30	10 ordu	10 ordu	0 ordu
Guztira:	2014/1/12	2014/6/30	350 ordu	360 ordu	10 ordu

2.1. Taula: Proiektuaren jarraipen taula.

2.3. Desbideratzeen zergatia eta izandako arazoak

Taulan ikus daitekeen bezala, formakuntza atalak desbiderapena handia izan du, 30 ordukoa hain zuzen ere. Hau gertatzeko arrazoi bat baino gehiago egon dira; lehenik, ikasketa kurtsoa egiteko espero baino denbora gehiago behar izan da, baina hau ez da izan atzerapen handiena eragin duena. Datetan erreparatuz gero, ikus daiteke bi asteko atzerapena egon dela eta hori robotean izandako hardware arazoaren errua izan da. Sentsore batzuekin hardware arazoak egon dira eta, horiek konpontzea gure esku ez zegoenez, proiektuaren garapena atzeratu egin da eta denbora gehiago eman da formakuntzan.

Garapenari dagokionez estimatu baino denbora gutxiago behar izan da, esan bezala, robota martxan jartzeko zenbait arazo egon direlako eta horrek robotarekin lanean aritzeko denbora epea txikitu egin du. Esan behar da proben diseinuan estimatu baino denbora gutxiago behar izan dela, laguntza jaso delako lankideen partetik. Baina, bestalde, probak egiteko estimatu baino denbora dezente gehiago behar izan da.

Kudeaketan estimazioak zuzen egin dira.

Proiektuan izandako arazorik larrienak robotaren hardwarearekin loturiko arazoak izan dira. Kasu batean sentsore batek huts egin du eta konpontzeko kanpoko enpresa batera bidali behar izan da, horrek eragin duen atzerapenarekin.

3. Kapituluia

Robot mugikorren nabigazioa

Gaien Aurkibidea

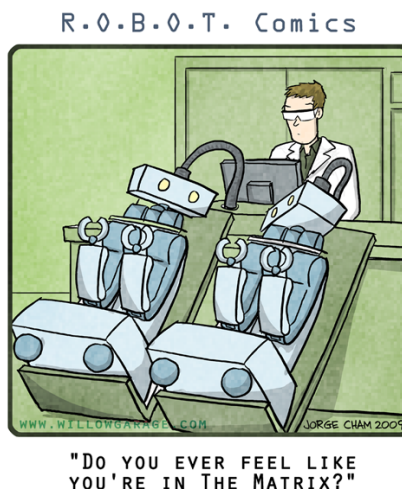
3.1. Sarrera	17
3.2. Robot mugikorren nabigaziorako eskema posible bat	18
3.3. Monte-Carlo metodoa	19
3.3.1. Sarrera	19
3.3.2. Monte-Carloren lokalizazioa	20
3.4. Kalmanen filtroak	24
3.4.1. Sarrera	24
3.4.2. Algorimoa	25
3.4.3. Jarraipena Kalmanen filtroak erabiliz	27
3.5. Partikula-filtroak	29
3.5.1. Sarrera	29
3.5.2. Kalmanen filtroarekin ezberdintasunak	30
3.5.3. Algoritmoa	31
3.5.4. Pisuen degenerazioa	33
3.5.5. Garrantzi funtzioaren aukeraketa	34
3.5.6. Birlagintzea (<i>Resampling</i>)	35
3.5.7. Ondorioak	35
3.6. Mugimenduen planifikatzaileak	37
3.6.1. Sarrera	37
3.6.2. Programazio dinamikoa	38
3.6.3. A*	39
3.6.4. Konparaketa	42

3.6.5. Ondorioak	43
3.7. Robotaren mugimendua	43
3.7.1. Sarrera	43
3.7.2. Leuntzea	44
3.7.3. PID kontrola	44

3.1. Sarrera

Nabigazioa, robot bat, oztopo ezberdinak egon daitezkeen ingurune batean, puntu batetik beste puntu batera eramateko metodologiatzat defini daiteke. Eskema eta metodo ezberdinak egon arren, guztien nahia ibilgailua helburura modu seguru batean eramatea da.

Robot batek nabigazioan egin behar dituen zereginak hauek dira: lehenik, inguruaren hautematea egin behar du, bere burua bertan lokalizatuz eta horretarako dituen sentsoreek eskaintako informazioa baliatuz; ondoren, bide bat planifikatu behar du, bere burua lokalizatu den puntutik helburura iristeko; jarraian, planifikatu den bidetik robota bideratu eta gidatu egin behar da.



Kapitulu honetan, robot mugikor batek modu seguruan nabigatzeko erabiltzen diren teknika batzuk azalduko dira. Lehenik eta behin, robot mugikor baten nabigazio eskema bat (3.2 atala) azalduko da. Ondoren, Monte-Carlo metodoa (3.3 atala) deskribatuko da, metodo honekin robota ingurune batean lokalizatu daiteke. 3.4 atalean, berriz, Kalmanen filtroak deskribatuko dira, objektuen *tracking*-a¹ egiteko erabil daitezkeenak. 3.5 atalean Partikula-Filtroak deskribatuko dira, eta aurretik aipatutako metodo estatistikoekin alderatuko da. Jarraian, 3.6 atalean, plangintza egiteko teknika batzuk azalduko dira: A* eta Programazioa dinamikoa (*Dynaminc Programming*) algoritmoak. 3.7 atalean robotak bidetik eraman eta kontrolatzeko teknikak: PID kontrola robota kontrolatzeko eta *smoothing*² teknika planifikatutako bidea leuntzeko.

¹*tracking* euskarara jarraitze moduan itzuli daiteke eta objektu baten egoera estimatzean datza.

²*smoothing* euskarara leuntze moduan itzuli daiteke eta ibilgailuak egin beharreko birak leuntzean datza.

3.2. Robot mugikorren nabigaziorako eskema posible bat

Robot mugikor batek nabigazio eginkizun bat duenean, bide bat egin behar du hasierako posizio batetatik abiatu eta amaierako posizio batetara heldu arte, tarteko posizio edo azpi-helburu batzuetatik igaroz.

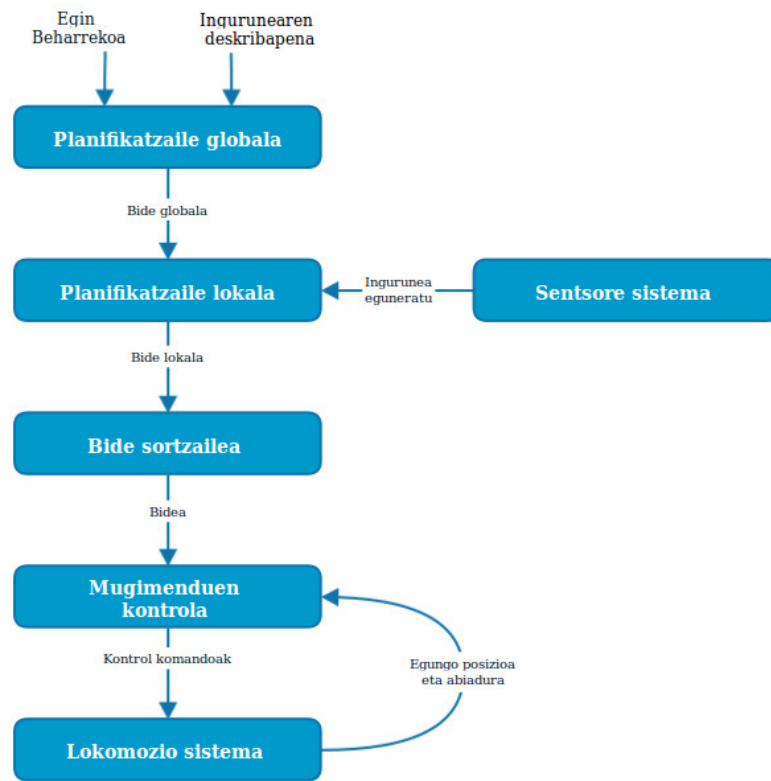
Nabigazioa lau fase hauetan banatzen da:

- **Robotaren lokalizazioa:** Ingurunearen mapa bat badugu, robotak bere burua kokatuko du bertan, sentsoreek jasotzen duten informazioa erabiliz.
- **Bidearen plangintza:** Ibilgailua helmugara iristeko jarraitu beharreko jomuga edo azpi-helburu sekuentzia ordenatu bat sortuko da. Hau sortzeko, ingurunearen mapa eta estrategia jakin bat erabiliko dira, estrategia ezberdinak daude.
- **Bidea sortu:** Hemen, bidearen plangintzan sortutako sekuentziatik funtzio jarraitu bat sortuko da, ondoren diskretizatu eta bidea sortzeko.³
- **Bideari jarraitu:** Ibilgailua gidatu eta kontrolatzen da, sortu den bidetik eramateko.

3.1. irudian ikus daitekeen eskema, robot mugikor baten nabigazio eskema posible bat da. Eskeman ikus daitekeen moduan, hasierako informazioa ingurunearen mapa eta robotak izan dezakeen betebeharrak edo ataza dira.

Lehenengo, robotak bere burua mapan kokatu behar du, hori egin eta gero, helburua planifikatzen da eta bertara iristeko bide seguru bat sortzen da planifikatzaile globalean. Ingurunea perfektua balitz, kalkulaturako bidea, zuzenean bide sortzaileari pasako litzaioke, baina, mapa perfektua kasu errealean existitzen ez denez eta ingurunean egon daitezkeen oztopo edo objektuak mugitu daitezkeenez, ibilgailuak ingurunean oztopoekin talka egiteko arriskua du. Horregatik, planifikatzaile lokalak sentsoreen informazioa jasotzen du etengabe eta, datuak aztertuta, planifikaturako bidean ea aldaketarik egin behar den erabakitzen du.

³Ataza hau aurrekoarekin oso lotuta dago, baina dokumentu honetan ezberdintzea erabaki da.



3.1. Irudia: Robot mugikor baten nabigazio eskema.

Ondoren, bidea sortzen da eta robotaren kontrol-sistemak, robotak erabakitako bidetik ahalik eta hobekien gidatzeko hartu beharreko erabakiak hartuko ditu, uneko posizioa eta abiadura kontuan edukita.

3.3. Monte-Carlo metodoa

3.3.1. Sarrera

Monte-Carlo metodoa, metodo numeriko bat da. Zorizko aldagaien simulazio bat eginez, problema fisiko zein matematikoei irtenbidea bilatzea ahalbidetzen duena. Monte-Carlo izena, metodoak kasinoetako ruleta jokuekin duen parekotasunetik dator.

Gaur egun Monte-Carlo metodoa, metodo numeriko edo analitikoak erabiliz konponbide zaila duten, baina zorizko faktoreekiko menpeko diren edo modelo

probabilistiko artifizial bati lotu daitezkeen arazoak konpontzeko erabiltzen da. Hala nola, aldagai askoko integralak, funtzioen minimizazioa, etab.

Monte-Carlo metodoak arazo hauei irtenbidea aurkitzen die, ordenagailu batean experimentuak eginez lagin estatistikoekin. Metodoa edozein problematan erabil daiteke, bai estokastiko zein deterministetan.

3.3.2. Monte-Carloren lokalizazioa

Sarrera

Robotika mugikorrean Monte-Carlo metodoak roboten lokalizazioa egiteko erabil daitezke.

Lokalizazioa, makina batek bere burua koordenatu sistema batean duen posizioa ezagutzeko duen ahalmena da. Ataza honetan arazo horri aurre egiteko erabil daitezkeen teknika bat azalduko da, Monte-Carlo lokalizazioa, hain zuzen ere.

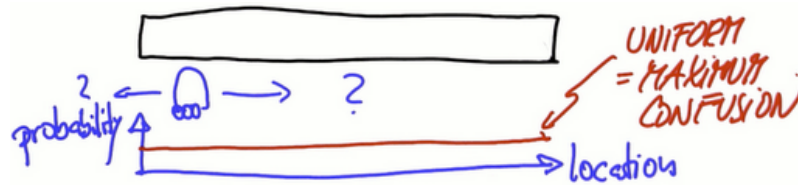
Hasiera batean, pentsa daiteke zenbait kasutan GPS sistema batekin konpondu daitekeela arazoa, baina ez da horrela. Sistema horien arazoa zehaztasun falta da, 1 eta 10 metro arteko erroreak izan ditzakete eta hori onartezina da, robot batek zehaztasun falta horrekin talkak edukiko baititu.

Monte-Carloren lokalizazio metodoa, roboten lokalizazioa egiteko metodo bat da. Ingurunearen mapa bat edukita, algoritmoak robotaren posizioa eta orientazioa estimatzen ditu, robota ingurunean mugitu ahala, sentsoreen bitartez informazioa jasoko du eta horrekin lokalizatuko du robotak bere burua mapan. Robotak hasiera batean ez daki non dagoen eta edozein lekutan egoteko probabilitatea berdina dela pentsatuko du. Robota mugitu ahala, eta sentsoreetatik ingurunearen informazioa jaso ahala, bere uneko posizioaren predikzioa egingo du.

Adibidea

Jarraian, adibide baten bidez azalduko da Monte-Carloren lokalizaziorako teknika, histograma filtro izenarekin ere ezagutzen dena. Adibidea *Artificial Intelligence for Robotics* kurtsoetik hartua da.

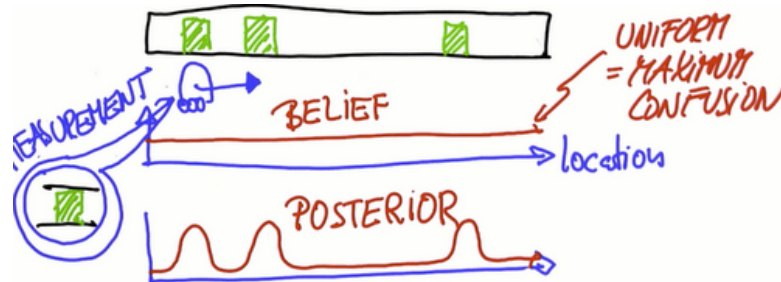
Demagun dimentsio bakarreko mundu bat dagoela eta robotak munduaren mapa duela. Robotak ez daki non dagoen kokatuta munduan eta, beraz, mapakoo edozein posiziotan egoteko probabilitatea berdina da berarentzat, matematikoki bere probabilitate funtzioa uniforme da esaten da. Egoera hori ziurgabetasun



3.2. Irudia: Monte-Carlo lokalizazioa, hasierako egoera.

handieneko egoera (*maximum confusion*) izango da. Adierazitakoa, 3.2. irudian ikus daiteke.

Munduan marka batzuk daude, kasu honetan 3 ate izango dira, eta robotak ate horiek identifika ditzake. Robotak sentsoreen bidez ate baten ondoan dagoela ikusi du, ondorioz, berak kokapenari buruz duen ustea aldatu egingo da. 3.3. Irudian ikus daiteke bere uste berria errepresentatzen duen funtzioa, *posterior*⁴ izenez identifikatzen dena. Aipatutako irudian ikus daitekeen moduan, probabilitate funtzioa aldatu egin da eta, orain, ateen ingurunekeo posizioetan egoteko probabilitatea handiagoa da, ate guztiek probabilitate bera izan arren.



3.3. Irudia: Monte-Carlo lokalizazioa, atea ikusi ondorengo egoera.

Orain, robota eskuinetara mugitzen dela suposatuko da eta honek bere ustea aldatuko du. Aldaketa mugimenduaren arabera da, eta 3.4. irudian ikus daiteke funtzioan duen eragina. Ikus daiteke robotaren sinismen funtzioa eskuinera mugitu dela eta gainera berdindu egin dela, tontorrak leunduz. Funtzioak zehaztasuna galdu du, hau da, ziurgabetasuna handitu da.

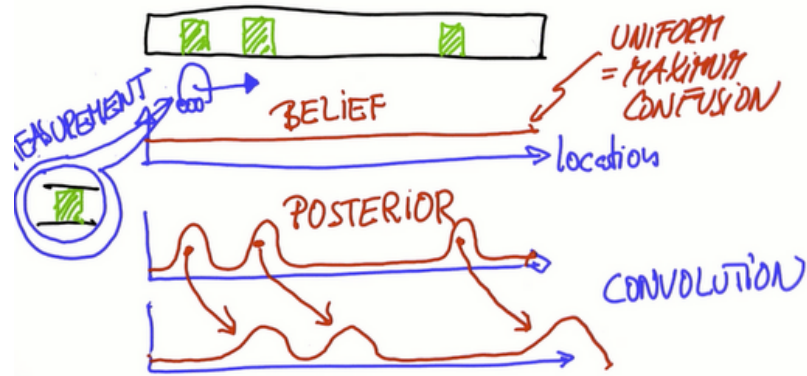
Funtzioa berdindu eta tontorrak leuntzeari, konboluzioa⁵ esaten zaio.

⁴Honela izendatu da neurketa bat eta gero sortu delako.

⁵Konboluzioa metodo matematiko bat da eta honela definitzen da: bi funtzioen biderketen arteko integrala, bietako bat η distantzia bat mugitu denean. f eta g funtzioen arteko konboluzioa:

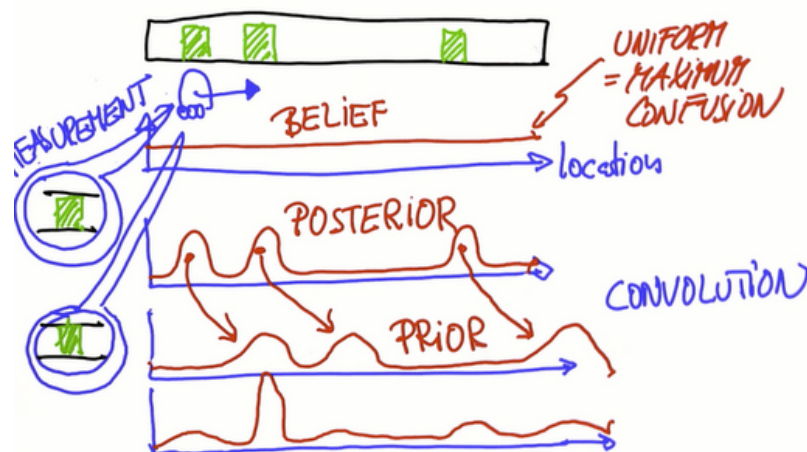
$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\eta)g(t - \eta)d\eta \quad (3.1)$$

Kasu honetan hauek dira konboluzioan erabili diren funtzioak: lehenengo funtzioa, robotaren sinismen funtzioa da, irudian *posterior* moduan adierazita dagoena. Bigarrena, robotak egindako distantzia definitzen duen funtzioa da, hau ez dago irudian. Emaitza, berriz, irudian ikus daitekeen leundutako funtzioa da.



3.4. Irudia: Monte-Carlo lokalizazioa, robota mugitu eta geroko egoera.

Orain, robotak beste ate bat ikusi duela suposatuko da, beraz badaki beste ate baten ondoan dagoela. Honek, lehenengo atearen kasuan bezala, probabilitate funtzioaren ziurtasuna handituko du. Kasu honetan, ordea, robota ez dago ziurgabetasun egoera batean, baditu bere kokapenaren inguruko susmo batzuk. Eta berak dituen susmoak bigarren atearen detekzioarekin konbinatuta probabilitate banaketa berri bat lortuko du, 3.5. irudian ikus daitekeena.

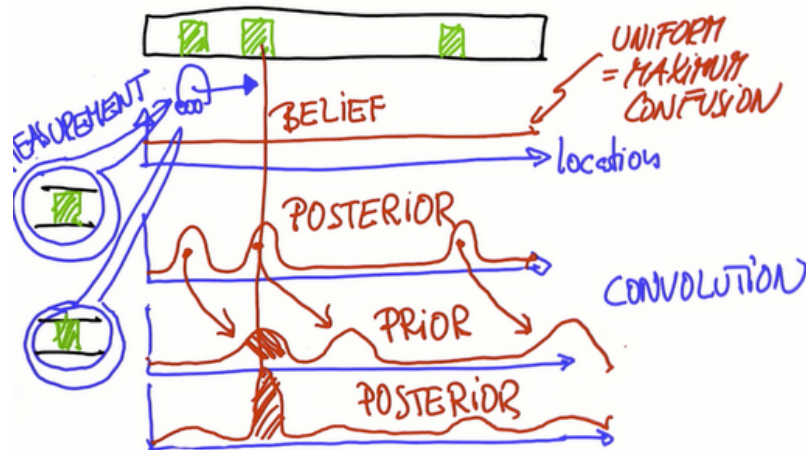


3.5. Irudia: Monte-Carlo lokalizazioa, bigarren atea detektatu eta geroko egoera.

Funtzio berrian tontor garbi bat identifika daiteke beste kokxa guztien gainetik.

Tontor hori bigarren ateari dagokio. Gertatu dena zera da: Lehenengo atea ikusi duenean, robotak bazekien ate baten ondoan zegoela, baina ez zekien zehazki zein atearen ondoan. Bigarrena ikustean, bi ate jarraian ikusi ditu, eta bi ate jarraian dauden posizioen inguruan egoteko probabilitate handia dagoela konturatu da.

Ikusi 3.6. irudia.



3.6. Irudia: Monte-Carloren lokalizazioa, robotak bere burua kokatu du.

Esan behar da robotak ez duela ziurtasun osoa bere kokapenarekiko, baina geroz eta gehiago mugitu, orduan eta ziurrago egongo da.

Algoritmoa

Jarraian, Monte-Carloren lokalizazioaren algoritmoa deskribatuko da, ikusi 3.7. irudia. Ingurunearen mapa bat emanda, robota bertan lokalizatzea da algoritmoaren helburua.

Algoritmok aurreko ataleko adibidean deskribaturikoa egiten du. Lehen urratsa, robotaren mugimendua eguneratu eta, bertatik, sinismen funtzio berri bat lortzea izango da; ondoren, bere sentsoreetatik jasotzen duen informazioa eguneratuko du eta beste probabilitate funtzio bat sortuko du; bi pausu horien ondoren, lortu dituen bi funtzioak konbinatuko ditu eta, amaitzeko, funtzio berria marraztuko du.

Algoritmoak t unero sarrera hauek edukiko ditu:

- Aurreko sinismena: $X_{t-1} = \{x_{t-1}^{[1]}, x_{t-1}^{[2]}, \dots, x_{t-1}^{[M]}\}$.

- Ekintza agindu bat: u_t .
- Sentsoreetatik jasotako informazioa: z_t .

Algoritmoak itzuliko duena hau izango da:

- Sinismen berri bat: X_t .

Algorithm 1: Monte-Carlo Lokalizazioa

Sarrera : X_{t-1}, u_t, z_t
Irteera : X_t

- 1 $\bar{X}_t = X_t = \emptyset$;
- 2 **for** $m = 1$ **to** M **do**
- 3 $x_t^{[m]} = \text{mugimenduakEguneratu}(u_t, x_{t-1}^{[m]})$;
- 4 $w_t^{[m]} = \text{sensoreakEguneratu}(z_t, x_t^{[m]})$;
- 5 $\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$;
- 6 **for** $m = 1$ **to** M **do**
- 7 draw $x_t^{[i]}$ from \bar{X}_t with probability $\propto w_t^{[i]}$;
- 8 $X_t = X_t + x_t^{[i]}$;
- 9 **return** X_t

3.7. Irudia: Monte-Carloren lokalizazioa, algoritmoa.

3.4. Kalmanen filtroak

3.4.1. Sarrera

Kalmanen filtroak sistema baten egoera ezkutua (neuritu ezin dena) estimatzeko teknika batzuk dira. Kalmanen filtroek egoera jarraitu bat estimatzen dute eta modal bakarreko banaketa bat itzultzen dute. Gainera, sistemak zarata zuria ⁶ duenean ere erabil daitezke.

Kalmanen filtroak denboran zehar aztertutako neurketen serieak erabiltzen ditu, zarata eta beste zehazgabetasun batzuk kontuan edukita, ezagutzen ez ditugun

⁶Zarata zuria ezaugarri hauek dituen prozesu estokastikoa da: bere itxaropen matematikoa 0 da une guztietarako, bariantza mugatua eta konstantea du, eta une ezberdinetako balioak korrelazio gabeak dira.

aldagaien estimazioak egiteko. Formalago esanda, Kalmanen filtroak sarrerako datuen gainean lanean aritzen dira modu errekurtsiboan, sistema baten egoeraren estimazio estatistiko ahalik eta zehatzena egiteko [8].

Filtro hauen asmatzailea Rudolf E. Kalman izan zen 1960. urtean eta, geroztik, ikerketa asko egin dira eta arlo ezberdinetan aplikatu dira. Esaterako: nabigazio autonomo zein laguntzazkoan, misilen rastrea egiteko edo ekonomian.

3.4.2. Algorimoa

Kalmanen filtroak ekuazio matematiko multzo batean oinarritzen dira eta hauek emaitza errekurtsibo hoberena itzultzen dute, karratu txikiaren erregresioa⁷ erabiliz. Helburua estimatzaile lineal bat sortzea da, t sistema batentzat $t - 1$ -en genuen informazioan oinarrituta eta t -n dagoen informazioarekin eguneratuz.

Filtroa erabiltzeko, sistema, modelo estokastiko⁸ lineal bat bezala deskriba daitekeela suposatzen da, non gehitzen den sistemari zein beste edozein informaziori loturiko erroreak, banaketa normala, media zero eta zehaztutako bariantza duen.

Emaitza optimoa da, filtroak atzemandako informazio guztia eta sistemaren portaeraren inguruko aurre ezagutza erabiltzen baititu, errorea estatistikoki minimizatuz. Errekurtsibo terminoarekin, sistemak neurri berri bat edo behaketa berri bat duenero emaitza ber kalkulatzeko delako adierazi nahi da.

Kalmanen filtroa algoritmo errekurtsibo bat da, non x_t egoera ausazko aldagai Gaussiartzat hartzen den. Errekurtsibo terminoarekin, sistemak neurri berri bat edo behaketa berri bat duenero emaitza ber kalkulatzeko delako adierazi nahi da. Kalmanen filtroa bi urratsetan deskribatzen da: **iragarpena** eta **zuzenketa**. Lehenengoak, egoeraren iragarpena eta t uneko egoeraren erroreakin kobariantza $t - 1$ unean oinarritzat hartuta kalkulatzeko arduratzen dira. Bigarren ekuazio multzoa, berriz, atzeraelikaduraren arduraduna da, hau da, aurreko *a priori* estimazioan informazio berria gehitzen dute eta honela egoeraren *a posteriori* estimazio hobea lortzen da. Jarraian daude ekuazioak:

Iragarpena

- *A priori* estimazioa:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{\Phi}_t \mathbf{x}_{t-1|t-1} \quad (3.2)$$

⁷Karratu txikiaren erregresioa, aldagai anitzeko datu multzo batera lerro bat doitzeko metodo bat da, hondar karratuaren batura minimotzen duena lerroaren parametroak zehazteko.

⁸estokastiko, zorizko aldagaien bilduma bat da eta hauek beste aldagai baten arabera aldatzen dira, gehienetan denbora.

- *A priori* estimazioari loturiko errorearen kobariantza:

$$\mathbf{P}_{t|t-1} = \mathbf{\Phi}_t \mathbf{P}_{t-1|t-1} \mathbf{\Phi}_t^T + \mathbf{Q}_t \quad (3.3)$$

Zuzenketa

- Neurketaren eguneraketa:

$$\tilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1} \quad (3.4)$$

- Kalmanen irabazia:

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^T (\mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t)^{-1} \quad (3.5)$$

- *A posteriori* estimazioa:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \tilde{\mathbf{y}}_t \quad (3.6)$$

- *A posteriori* estimazioari loturiko errorearen kobariantza:

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1} \quad (3.7)$$

non:

- $\mathbf{\Phi}_t$: Egoera trantsizio matrizea. $\mathbf{x}_{t|t-1}$ eta $\mathbf{x}_{t-1|t-1}$ erlazionatzen dituen matrizea den, denboraren menpe dauden funtzioak egon ezean.
- $\mathbf{x}_{t|t-1}$: Egoera bektorearen *a priori* estimazioa den.
- $\mathbf{P}_{t|t-1}$: *A priori* estimazioari loturiko errorearen kobariantza den.
- \mathbf{z}_t : t uneko neurketa bektorea den.
- \mathbf{H}_t : Neurketen eta t uneko egoera bektorearen arteko erlazioa adierazten duen matrizea den, neurketetan zaratarik ez dagoen egoera ideal batean.
- \mathbf{R}_t : Neurketen zaraten kobariantza matrizea den.

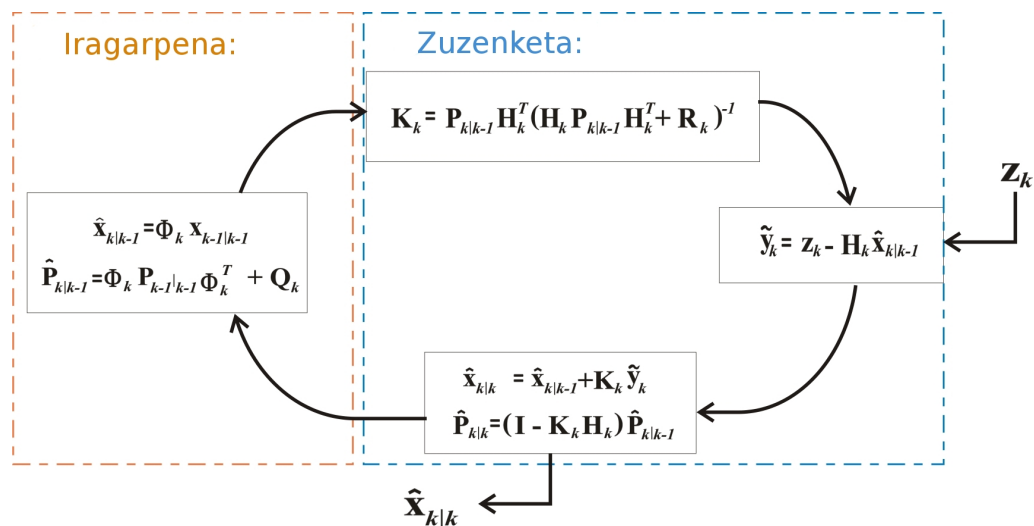
3.8. Irudian Kalmanen filtroaren algoritmo errekurtsiboa ikus daiteke. Ikusi iragapen ekuazioek egoeraren estimazioa ($\hat{\mathbf{x}}_{t|t-1}$) eta errorearen kobariantzaren estimazioa ($\mathbf{P}_{t|t-1}$) egiten dutela aurrerantz, $t - 1$ -etik t -ra. $\mathbf{\Phi}_t$ matrizeak $t - 1$

aurreko egoera, uneko egoerarekin t erlazionatzen du. \mathbf{Q}_t -k egoera estimatu nahi duen prozesuaren ausazko perturbazioaren kobariantza adierazten du.

Zuzenketa egin behar den lehenengo gauza Kalmanen irabazia \mathbf{K}_t kalkulatzeko da. Honek egoeraren estimazio berriaren errorearen kobariantza txikituko du. Ondorengo urratsa z_t neurketa egitea da eta, honekin, estimazio $\hat{\mathbf{x}}_{t|t}$ berri bat egitea. Azken urratsa estimazioari loturiko errorearen kobariantza $\hat{\mathbf{P}}_{t|t}$ hobeto bat lortzea da. Prozesua errepikatu egingo da, estimazio eta errorearen kobariantza berriarekin.

Algorithm 2: Kalmanen Filtroa

1



3.8. Irudia: Kalmanen filtroa, algoritmoa.

3.4.3. Jarraipena Kalmanen filtroak erabiliz

Sarrera

Atal honetan robotika mugikorren filtro honek izan dezakeen erabilera bat deskribatuko da, objektuen jarraipena egiteko erabilela hain zuzen. Hau da, ingurunean dauden objektuei jarraitzea egitea, talkak gerta ez daitezken.

Adibidea

Jarraian adibide batekin azalduko da objektu baten jarraipena zer den eta Kalmanen filtroa zertarako erabil daitezkeen. Adibidea *Artificial Intelligence for Robotics*

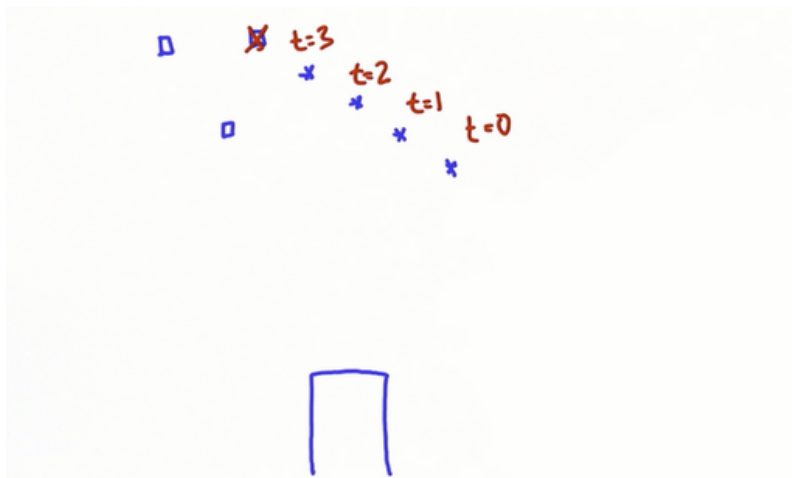
kurtsotik hartua da.

Demagun robot mugikor bat dagoela, 3.9. irudian laukizuzen batekin marraztuta dagoena. Honek dituen sentsoreek objektu bat atzeman dute ingurunean. Objektua * batekin identifikatu da irudian eta denboran zehar ikus daiteke : $t=0$, $t=1$, $t=2$, $t=3$, mugitzen ari dela.



3.9. Irudia: Kalmanen filtroak, objektu mugikorraren posizioak.

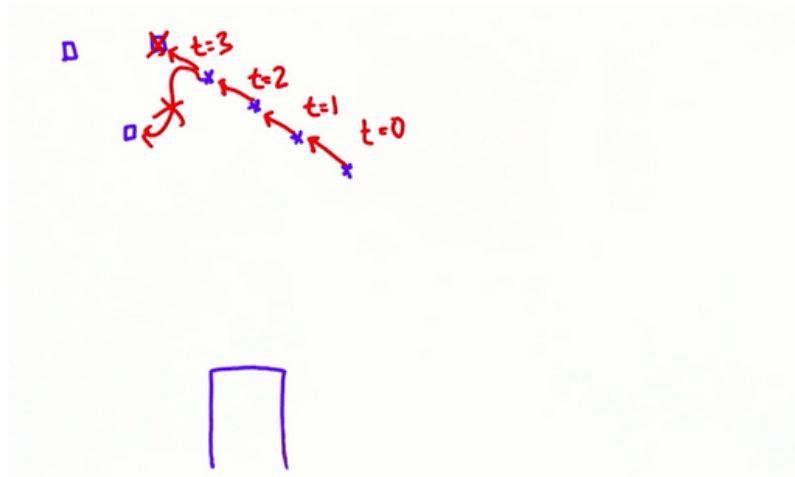
Non egongo da objektua $t=4$ denean? Objektuaren abiaduran aldaketa handiegirik gertatu ez bada, intuizioa erabiliz, erraz asma daiteke non egongo den objektua. 3.10. irudian x batekin markatu da posizioa, intuizioz, inongo eragiketa matematikorik egin gabe estimatu den posizioa.



3.10. Irudia: Kalmanen filtroak, objektu mugikorraren posizioaren estimazioa.

Kalmanen filtroa hori bezalako estimazioak egiteko erabil daitezke, ziurtasun osoa ez eduki arren. Gainera, objektuaren abiadura ere identifika dezake.

3.11. irudian objektuaren ibilbidea adierazten da geziekin, Kalmanen filtroak estimatuko lukeena.



3.11. Irudia: Kalmanen filtroak, objektuaren mugimendua geziekin.

Esaterako Google-ren auto autonomoak honelako teknikak erabiltzen ditu trafikoa ulertzeko, bere laser eta radar sentsoreek biltzen duten informazioa erabiliz.

3.5. Partikula-filtroak

3.5.1. Sarrera

Partikula-filtroak, denboran zehar aldatzen doan sistema baten egoera estimatzeko erabiltzen den metodoa da. 1993. urtean N. Gordon, D.Salmond eta A.Smith-ek proposatu zuten *bootstrap* filtro bezala, bayesiar filtro errekurtsiboak inplementatzeko Monte-Carlo metodoak erabiliz [2]. Algoritmo hau SIS (*Sequential Importance Sampling*) algoritmoa bezala ezagutzen da.

Oinarrian, partikula-filtroa, lagin multzo batez (partikulak) eta partikula bakoitzak erlazonaturik duen balio batzuek (pisua) osatzen da. Partikulak, prozesu baten espazioan puntu moduan adierazten diren prozesuaren egoera posibleak dira. Eta pisua puntu horietako bakoitzak duen balio bat da.

Lau fase nagusi ditu:

- Hasieraketa
 - Partikula-filtroak zoriz puntu multzo bat ipintzen du, jarraipena egiteko (adibidez, mapa baten gainean, lokalizazioa egiteko edo irudi batean tracking-erako). Hasieraketa fasean, partikula multzoa zoriz sortu daiteke, edo *a priori* informazio motaren bat erabil daiteke (objetuaren tamaina, gutxi gora beherako posizioa...)
- Eguneraketa
 - Helburu egoera eta partikula bakoitzaren egoera kontuan edukita pisu bat esleituko zaio partikula bakoitzari.
- Estimazioa
 - Esleituta dueten pisu horietan oinarrituz, partikula multzo berri bat sortuko da. Multzo berri hau, *a priori* egoerak urrengo denbora unean izango duen egoeraren estimazioa izango da. Horretarako, birlagintze (*resampling*) metodo estatistikoak erabili ohi dira, hauetan partikula bakoitzaren *a posteriori* probabilitatea kontuan edukiz partikula berriak sortuko dira, probabilitate handiena duten partikuletan oinarrituz. Sortzen diren partikula guztiek pisu bera izango dute.
- Iragarpena
 - Denbora une berrirako partikula multzo berria sortu eta gero, bakoitzaren egoeran aldaketa txiki bat egiten da, zarata motaren bat gehituz sistenan aldaketak sortu eta, urrengo unean objektuak izango duen egoera estimatu asmoz.

Behin iragarpen fasea amaituta, partikula multzo berri bat lortzen da, eta, berriro, eguneraketa fasean egongo da algoritmoa, begizta hori errepikatuz datu sekuentzia amaitu arte. Kasu honetan, hasieraketa fasera itzuliko da.

3.5.2. Kalmanen filtroarekin ezberdintasunak

Kalmanen filtroak duen murriztapen bat, egoeraren aldagaiak Gaussiar moduan banatzen direla da. Horregatik Kalmanen filtroen estimazioak ez dira oso zuzenak banaketa hori jarraitzen ez duten egoera aldagaientzat. Murriztapen hau gainditu daiteke partikula-filtroak erabiliz. Partikula-filtroetan, t unean baldintzazko egoerak duen dentsitatea, pisua (lagintze probabilitatea) esleituta duten partikula multzo batzuek errepresentatzen dute. Pisuek partikularen garrantzia definitzen dute. Konplexutasun konputazionala gutxitzeko asmoz, tupla⁹ bakoitzarentzat,

⁹n-koptea edo tupla, n objetuen zerrenda ordenatua da, eta objektu horiek bere elementuak dira

metatzen joango den pisu bat gordeko da c_n , non $c_N = 1$. t uneari dagozkion lagin berriak, aurreko urratsean marraztu dira $t - 1$ unean eta horretarako laginketa eskema ezberdinak erabil daitezke. Laginketa eskema ohikoena garrantzizko laginketa (*importance sampling*) da, urrengo atalean definitzen dena.

3.5.3. Algoritmoa

Partikula-filtroen ideia, probabilitateen dentsitate-funtzioa (pdf) errepresentatzea da, ondoren pisua esleituta duten partikula multzo batek erabiliko duena, sisteman azken egoeraren estimazioa egiteko, lagin eta pisu horietan oinarrituz. Lagin kopurua handitu ahala ($N \rightarrow \infty$), Monte-Carlo metodoaren ondorioz, probabilitate dentsitate-funtzioaren hurbilketa zehatzagoa da eta ondorioz partikula filtroak estimazio Bayestar zuzenagoa lortuko du.

Algoritmoaren xehetasunak garatzeko, egoera espazioan lagindutako puntuak honela adieraziko dira: $\{x_{0:t}^i\}_{i=1}^{N_s}$, eta beraien pisuak: $\{w_t^i\}_{i=1}^{N_s}$. Pisu hauek normalizatuta daude, $\sum_t w_t^i = 1$. Horrela, *a posteriori* t unean, $p(x_t|z_t)$ probabilitateen dentsitate-funtzioaren, hurbilketa bat lor daiteke:

$$p(x_t|z_t) \approx \sum_{i=1}^{N_s} w_t^i \delta(x_t - x_t^i) \quad (3.8)$$

Horrela, probabilitateen dentsitate-funtzioaren (pdf) *a posteriori* $p(x_{0:t}|z_{1:t})$ hurbilpen haztatu diskretu bat lortuko da.

Pisuak garrantzizko laginketa printzipioa erabiliz erabakitzen dira. Printzipio hau ondorengoan oinarritzen da: suposatuz, $p(x) \propto \pi(x)$ probabilitateen dentsitate-funtzio bat dela, berarentzat laginak marraztea zaila izan harren, $\pi(x)$ ebaluatu daiteke. Gainera, $x^i \sim q(x)$, $i = 1, \dots, N_s$ laginak, proposaturiko $q(\cdot)$ funtzio batetik sortu dira, garrantzi funtzioa deiturikoa. Beraz, $p(\cdot)$ -ren pdf hurbilketa haztatua honela izango litzateke:

$$p(x) \approx \sum_{i=1}^{N_s} w^i \delta(x - x^i) \quad (3.9)$$

non:

$$w^i \propto \frac{\pi(x^i)}{q(x^i)} \quad (3.10)$$

i -garren partikularen pisu normalizatua den.

Bayes-en teorema aplikatuta, errekurtsiboki, $p(x_{0:t-1}|z_{1:t-1})$ oinarritutako *a posteriori*, $p(x_{0:t}|z_{1:t})$ probabilitate dentsitate-funtzioa eskuratzeko funtzioa honela deribatzen da:

$$\begin{aligned}
p(x_{0:t}|z_{1:t}) &= \frac{p(z_t|x_{0:t}|z_{1:t-1})p(x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\
&= \frac{p(z_t|x_{0:t}, z_{1:t-1})p(x_t|x_{0:t}, z_{1:t-1})p(x_{0:t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\
&= \frac{p(z_t|x_t)p(x_t|x_{t-1})}{p(z_t|z_{1:t-1})}p(x_{0:t-1}|z_{1:t-1}) \quad (3.11)
\end{aligned}$$

non $p(z_t|z_{1:t-1})$ proportzionalitate konstante bat den, ondorioz:

$$p(x_{0:t}|z_{1:t}) \propto p(z_t|x_t)p(x_t|x_{t-1})p(x_{0:t-1}|z_{1:t-1}) \quad (3.12)$$

Kasu sekuentzialera itzuliz, iterazio bakoitzean, $p(x_{0:t-1}|z_{1:t-1})$ -era hurbilpen bat osatzen duten laginak lortzen dira. Helburua pdf-a *a posteriori* $p(x_{0:t}|z_{1:t})$ -ra hurbiltzea da. Horretarako algoritmoak partikula berriak sortuko ditu x_t^i eta dagoeneko existitzen direnei ($x_{0:t-1}^i$) batu, egoera berria sortu ($x_{0:t}^i$) eta pisuak, w_t^i eguneratzeko. Honela, lortu nahi dugun emaitzara gerturatuko gara.

Garrantzi funtzioa, faktorizatzeko aukeratu bada, honela:

$$q(x_{0:t}|z_{1:t}) = q(x_t|x_{0:t-1}, z_{1:t})q(x_{0:t-1}|z_{1:t-1}) \quad (3.13)$$

orduan, eskuratutako bigarren faktorea, garrantzi faktoreari dagokio aurreko denbora unerarte eta lehenengo faktorea, uneko denbora funtzioa da. Honela, laginak eskuratu daitezke $x_{0:t}^i \sim q(x_{0:t}|z_{1:t})$ ihadanik existitzen direnei esleituz $x_{0:t-1}^i \sim q(x_{0:t-1}|z_{1:t-1})$, egoera berriarekin:

$$x_{0:t}^i \sim q(x_t|x_{0:t-1}, z_{1:t}) \quad (3.14)$$

pisuak w_t^i eguneratzeko unean, 3.10 espresioan oinarrituz:

$$w_t^i \propto \frac{p(x_{0:t}^i|z_{1:t})}{q(x_{0:t}^i|z_{1:t})} \quad (3.15)$$

Probabilitate dentsitate-funtzioaren *a posteriori* normalizazio faktorea ezezaguna denez, pisuak era proportzionalan soilik zehaztu daitezke. pisuak eskuratzeko, ekuazioak ordezkatzuz gero, erlazio hau geratuko litzateke:

$$w_t^i \propto \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)p(x_{0:t-1}^i|z_{1:t-1})}{q(x_t^i|x_{0:t-1}^i, z_{1:t})q(x_{0:t-1}^i|z_{1:t-1})} = w_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, z_t)} \quad (3.16)$$

Estimazio errekurtsibo Bayestarrean, une bakoitzaren *a posteriori* pdf-a kalkulatzeko interesatzen zaigu, $p(x_t^i|z_{1:t})$. Garrantzi funtzioa hau dela kontuan hartuta: $q(x_t|x_{0:t-1}, z_{1:t}) = q(x_t|x_{t-1}, z_t)$, soilik aurreko unearen egoeraren eta uneko behaketaren menpe dago. Hau oso erabilgarria da, denbora une bakoitzean $p(x_t|z_{1:t})$ -ren iragazitako estimazioa soilik eskatzen den kasuetan. Hemendik aurrera kasu horretan gaudela suposatuko da, Markov-en lehen ordeneko iragazkia izanik, ez baita informazioa galtzen. Egoera horretan, soilik x_t^i gorde behar da; honela, aurreko multzoa $x_{0:t-1}^i$ eta egindako behaketak $z_{1:t-1}$ alde batera utz daitezke. Horrela, partikulak eraldatutako garrantzi funtzio honetatik sortuko dira:

$$x_t^i \sim q(x_t|x_{t-1}, z_t) \quad (3.17)$$

eta pisuak, berriz:

$$w_t^i \propto w_{t-1}^i \frac{p(z_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, z_t)} \quad (3.18)$$

Partikulak 3.17 erlazioarekin sortu badira eta beraien pisuak 3.18 erlazioarekin, jarraian ikus daiteke *a posteriori* probabilitate dentsitate-funtzioaren hurbilketa:

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^{N_s} w_t^i \delta(x_t - x_t^i) = p_N(x_t|z_{1:t}) \quad (3.19)$$

Froga daiteke, lagin kopurua handitu ahala ($N \rightarrow \infty$) 3.19 ekuazioak benetako *a posteriori* probabilitate dentsitate funtziora $p(x_t|z_{1:t})$ konbergitzen duela.

3.5.4. Pisuen degenerazioa

SIS (*Sequential Importance Sampling*) algoritmoan urrats kopuru jakin bat egin eta gero, arazo bat sortzen da: partikula kopuru txiki batek soilik duela pisu ez arbuigarria. Arazo hau, pisuen degenerazio moduan ezagutzen da. Aurreko atalean aurkeztutako algoritmoan ez da irtenbiderik aurreikusi arazo honentzat, baina arazo honen estimazioa egin daiteke daiteke N_{eff} lagin tamaina eraginkor bat erabiliz:

$$N_{eff} \approx \hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (w_t^i)^2} \quad (3.20)$$

Non, w_t^i pisu normalizatuak diren (3.16 ekuazioa). Partikula guztiek pisu bera badute, hau da, $w_t^i = \frac{1}{N}$ non $i = 1, \dots, N$, orduan partikulen efektibitatea maximoa da, $N_{eff} = N$. Bestalde, partikula guztiek 0 pisua badute batek izan ezik, efektibitatea minimoa da, $N_{eff} = 1$. Horrek esan nahi du partikula kopurua gutxitzean algoritmoa degeneratu egingo dela. Aurre egiteko soluzio bat, partikula kopuru

N oso altua erabiltzea izan daiteke, baina ez da praktikoa, karga konputazional oso handia suposatuko bailuke. Beste soluzio posible bat, garrantzi funtzio egoki bat erabiltzea da partikulak aukeratzeko. Hala eta guztiz, lortuko dena arazoa beranduago azaltzea izango da.

3.5.5. Garrantzi funtzioaren aukeraketa

$q(x_t^i|x_{t-1}^i, z_t)$ Garrantzi funtzioaren aukeraketa partikula-filtroaren diseinuaren atal kritikoetako bat da. Funtzioaren dentsitate optimoa bilatzeko unean, kon-tuan edukitzen da SIS algoritmoaren degenerazioa mugatzeko modu bat garrantzi funtzioarekin pisuen bariantza minimizatzea dela eta, ondorioz, N_{eff} maximizatzen dela. Haukeraketa egokiena, Doucet et al.[1] frogatzen duenez: $q(x_t^i|x_{t-1}^i, z_t) = p(x_t^i|x_{t-1}^i, z_t)$ da, kasu honetan pisuen bariantza zero baita. Honela idatz daiteke:

$$q(x_t^i|x_{t-1}^i, z_t) = p(x_t^i|x_{t-1}^i, z_t) = \frac{p(z_t|x_t, x_{t-1}^i)p(x_t|x_{t-1}^i)}{p(z_t|x_{t-1}^i)} \quad (3.21)$$

Beraz, 3.16 ekuazioan ordezkaturaz, pisuen kalkulua horrela da:

$$w_t^i \propto w_{t-1}^i p(z_t|x_{t-1}^i) = w_{t-1}^i \int p(z_t|x_t)p(x_t|x_{t-1}^i)dx_t \quad (3.22)$$

Ikus daiteke nola lor daitezkeen t uneko pisuak partikulak ugaltu baino lehenago. Normalean, ezin daiteke $p(x_t|x_{t-1}^i, z_t)$ probabilitateen dentsitate-funtzioa lagindu eta integralak (3.22) ez du forma analitikorik. Hala eta guztiz ere, kasu batzuetan, egoera Gaussiar batean esaterako, posible da ebaluazio analitiko bat egitea.

Bestalde, egokiena ez izan arren, aukeraketa egoki eta erraza, garrantzi funtzio bezala *a priori* funtzioa erabiltzea litzateke:

$$p(x_t|x_{t-1}^i, z_t) = P(x_t|x_{t-1}^i) \quad (3.23)$$

3.23 ekuazioa 3.16 erlazioan ordezkaturaz gero, pisuak denbora une bakoitzean eguneratuko dira:

$$w_t^i \propto w_{t-1}^i p(z_t|x_t^i) \quad (3.24)$$

Orain ezinezkoa da pisu berriak lortzea aurretik partikukulak ugaltu gabe.

3.5.6. Birlagintzea (*Resampling*)

Algoritmoan birlagintzea erabiltzeak pisuen degenerazio arazoa konpontzen du, [4]. Birlagintzeak pisu txikieneko partikulak ezabatzen ditu eta pisu haundiene-koen kopiak sortu, laginak duen partikula kopuru eraginkorra N_{eff} , definitutako atalasearen N_T , azpitik dagoenean.

Birlagintze teknika ezberdin asko daude, jarraian teknika errazenetako bat aurkez-tuko da, birlagintze multinomiala moduan ezagutzen dena hain zuzen ere.

t unean egindako zorizko neurketa $\{x_t^i, w_t^i\}$, birlagintzea egiten ari den bitartean multzo berrian eraldatuko da $\{x_t^{*i}, \frac{1}{N}\}$, non partikula guztiek pisu bera duten. Partikula berri bakoitza $\{x_t^{*i}\}$, gutxi gorabeherako *a posteriori* funtzioa birlagintzen sortzen da:

$$p(x_t|z_t) \approx \sum_{i=1}^N w_t^i \delta(x_t - x_t^i) \quad (3.25)$$

Bere pisuari dagokion probabilitetearekin:

$$Pr(x_t^{*i} = x_t^j) = w_t^j \quad (3.26)$$

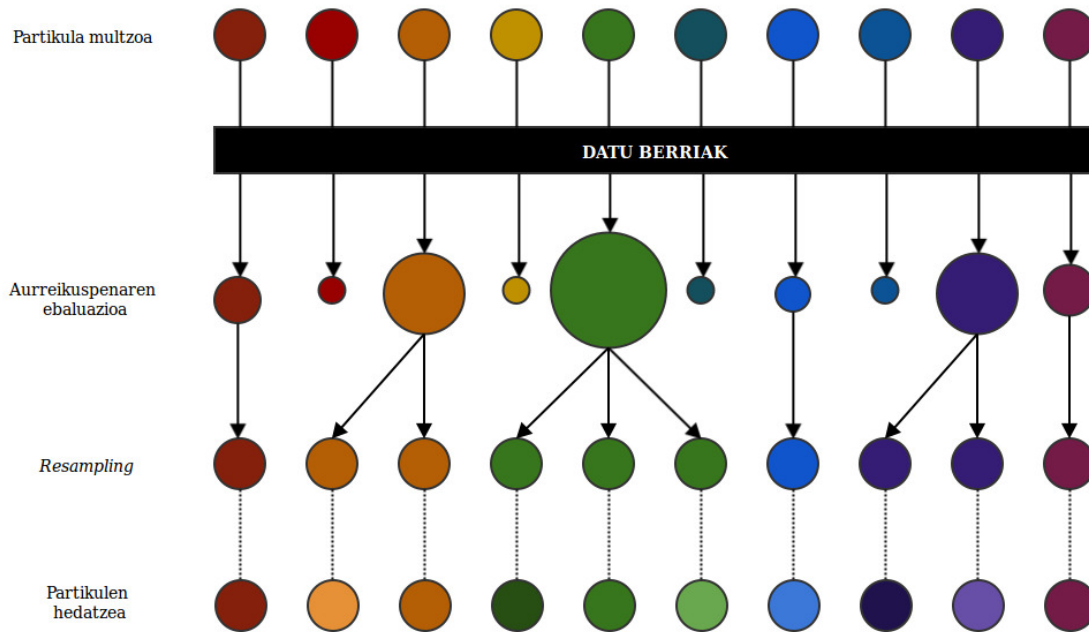
3.26 emaitza, probabilitateen dentsitate-funtzio diskretuaren i.i.d (*independent and identically distributed*) lagin bat da.

3.12. irudian, partikula-filtroaren eta birlagintzearen funtzionamendua ikus dai-teke. Hasieran partikula multzo bat dago, eta ingurunearen datuak jasotzen dituenean, hasieran dauden partikulei pisua esleituko zaie, irudian zenbat eta biribil handiagoa hainbat eta pisu handiagoa. Ikus daitekeen moduan birlagintzea egitean pisu txikieneko partikulak ezabatu egiten dira eta pisu handienekoetatik partikula gehiago sortzen dira, hasierako partikula kopuru bera lortuz. Ondoren, egoera aldaketaren ondorioz, partikulak hedatzen dira. Hau ziklikoki errepikatu-ko da.

Birlagintze metodoa, pisuaren degenerazioaren arazoa gutxitzeko erabiltzen den metodo bat da, baina, bestalde, pisu oso handiko partikulak aukeratzeak, birlagin-tze urrats bakoitzaren ondorengo laginak lagin errepikatuak edukitzea eragiten du. Horrek aniztasuna galtzea eragiten du eta, honi, lagin txirotzea deritzo.

3.5.7. Ondorioak

Aurreko atazetan ikusi da, SIS deskribatutako algoritmoak arazo bat zuela, pisuen degenerazio arazoa. Arazo hori konpontzeko birlagintzearen teknika deskribatu



3.12. Irudia: Partikula-filtroak eta birlaginketaren funtzionamendua.

da. Normalean erabiltzen den algoritmoa SIR Garrantziako Birlagintze Sekuentziala (*Sequential Importance Resampling*) da, zuzenean birlagintze atala algoritmoan bertan eginez. Hau izango da atal praktikoan erabiliko den partikula-filtro algoritmoa.

Esan behar da, partikula-filtroak robotaren lokalizaziorako edota objektuen jarraipena egiteko erabil daitezkeela. Beraz, aurreko ataletan deskribatu diren Montecarlo filtroak (Histograma filtroak), Kalmanen filroak eta Partikula-filtroak, hirurek gauza antzekoak egiteko balio dute. Orduan noiz erabili behar da filtro bakoitza?

- Partikula-filtroak hiruren artean errezenak dira inplementatzeko orduan, baina, arazoaren dimentsio kopurua handitu ahala, partikula-filtroen konplexutasuna esponentzialki haundituko da.
- Kalmanen filtroak dira esponentzialki handitzen ez diren bakarrak. Beraien arazoa da *unimodal* motakoak direla, beraz ezin dute hipotesi askorekin lan egin. Nahiz eta badauden Kalman filtroen hedadura batzuk arazo horri aurre egiteko.

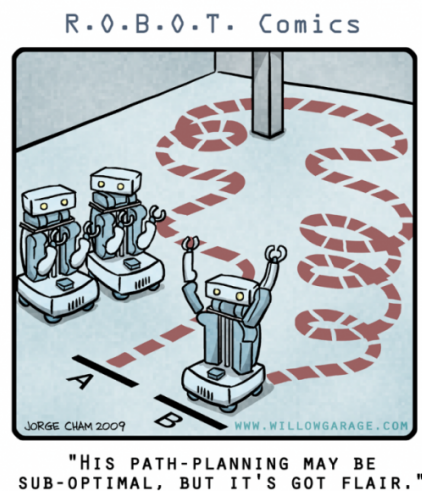
- Histograma filtroak, partikula-filtroen egoera antzekoetan erabili daitezke, ziurgabetasun egoera global batean.

Hau kontuan izanda, banaketa *multimodal*-etan Partikula-Filtroa da erabilgarriena, posible bada eta *unimodal* banaketa duten espazio jarraituetan, berriz, Kalmanen Filtroak erabiltzea gomendatzen da.

3.6. Mugimenduen planifikatzaileak

3.6.1. Sarrera

Atal honetan, robot mugikor batek leku batetik bestera mugitu behar duenean bide egokia aukeratzeko erabil daitezkeen bi algoritmo deskribatuko dira: A* eta Programazio Dinamiko *Dynamic programming* eta, ondoren, bien konparaketa egingo da. Atal honetan deskribatuko dena, *Artificial Intelligence for Robotics* kurtsoan ikasitakoan eta *Un estudio de dos métodos de planificación de trayectorias en entornos estáticos* [6] ikerketan oinarrituta dago.



Jarraian azalduko diren planifikazio metodoetan, datu hauek ezagunak direla aurreikusten da: ingurunearen mapa bat, hasierako puntu bat, helburu puntu bat eta kostu bat. Helburua, berriz, kostu txikieneko bidea aurkitzea izango da.

Planifikazio metodoak, hein handi batean, lan egingo duten ingurunearen menpe daude. Ingurune hauek estatikoak edo dinamikoak izan daitezke. Ingurune estati-koetan, ingurunearen ezagutza osoa edukitzen da eta bere mapa ez da denboran

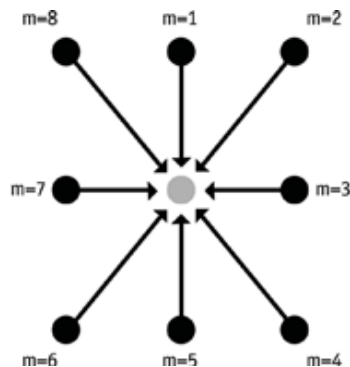
zehir aldatuko. Ingurune dinamikoetan, berriz, ez da informazio osoa edukitzen, oztopo eta objektuak denboran zehar alda baitaitezke.

3.6.2. Programazio dinamikoa

Ingurune konplexuetan, robot mugikorrek helburura joateko egin beharreko ibilbideak planifikatzeko erabiltzen da algoritmo hau. Horretarako, alderdi formal batzuk definitzen dira, esaterako inugurunearen matrize bat:

$$M(i,j) = \begin{cases} 0, & \text{oztopoa dagoen gelaxka bada} \\ 1, & \text{libre dagoen gelaxka bada} \end{cases} \quad (3.27)$$

eta emandako egoera batean robot batek egin ditzakeen mugimenduak, 3.13. irudian daudenak dira.



3.13. Irudia: Robot batek egin ditzakeen mugimenduak, gelaxka batetik ondoan duen beste batetara joateko.

Algoritmo honetan, egoera bakoitza, robotak okupatu dezakeen posizio bat da, hau da, inguruneke gelaxka bat. Gainera, egoera bakoitzak kostu bat du esleituta, ekuazio honekin kalkulatu daena:

$$C_{k+1,i,j} = \min(F(i,j,m) + C) \quad (3.28)$$

non,

- $C_{k+1,i,j}(i,j)$ gelaskaren kostua den, $k + 1$ iterazioan.
- $F(i,j,m)$ helburu funtzioa den, kasu honetan (i,j) egoeraren eta egindako mugimenduaren arteko distantzia bezala definitu da.

- C aurreko egoeraren kostua dem, hau da, aurreko egoeraren kostua.

Egoera baten kostuak -1 balioa badu, egoera hori robotarentzat galerazita dagoela esan nahi du.

Jarraian, algoritmoaren sasikodea deskribatzen da.

Programazio dinamikoa algoritmoaren sasikodea:

- Amaierako egoeratik hasi bilaketa, bere kostuaren balioa 0 ipiniz eta bere mugimendu mota 0.
- Beste egoera guztiak -1 kostuarekin hasieratu.
- Ondorengoa errepikatu (hasierako egoeraren kostua = -1 den bitartean, edo -1 kostua duten egoera kopuruak eta aurreko egoeran -1 kostua duten egoera kopuruak berdinak diren bitartean):
 - Egoera matrizearen uneko egoera bakoitzarentzat:
 - * Ikusi zein egoeretatik irits daitekeen uneko egoerara. Kalkulatu bide posible bakoitzaren kostua eta kostu txikiena duena aukeratu. Gorde bide horretarako erabiltzen den mugimendu mota.
 - * Egoera batek -1 kostua edukitzeari uzten badio, -1 balioa duten uneko egoera kopurua eguneratu.
 - * Uneko egoera hasierako egoera bada eta kostua -1 en desberdina bada, lehen emaitza aurkitu da.
 - Aurreko kostuen matrizea eguneratu.
 - Hurrengo iterazioan aztertuko diren egoerak adierazi.
- Algoritmoaren amaiera.

3.6.3. A*

Algoritmo honek, aurreko atalean deskribatu den Programazio dinamikoak ez bezala, funtzio heuristikoko bat erabiltzen du. Algoritmo hau asko erabiltzen da, ingurune ezagun edo estatikoetan, ibilbide optimoak bilatzeko.

Algoritmo honek bi zerrenda edukiko ditu: **Irekiak** izeneko zerrenda bat, sortu diren nodoak baina oraindik beren ondorengoak sortu ez dituztenak gordetzeko; eta, **Itxiak** izeneko zerrenda bat, iadanik aztertu diren nodoak gordetzeko. Nodo bakoitzarentzat, kostu funtzio bat definitzen da:

$$f(n) = g(n) + h(n) \quad (3.29)$$

non,

- $g(n)$ iturburutik n nodorarteko bide optimoaren kostua den.
- $h(n)$ heuristiko bat den, n nodotik helburura iristeko kostuaren estimazio bat egiten duena.

Jarraian, algoritmoaren sasikodea deskribatzen da.

A* algoritmoaren sasikodea

- **Irekiak** zerrenda hasieratu hasiera nodoa ipiniz. Nodo horren g balioan 0 ipini eta dagokion h balioa, $f = h$. **Itxiak** zerrenda hutsik hasieratu.
- Hau errepikatu helburu nodoa aurkitu arte:
 - **Irekiak** zerrendan ez badago nodorik, ERROREA.
 - Beste kasuetan, **Irekiak** zerrendako f balio hoberena (balio txikiena) duen balio aukeratu eta **NodoHoberena** moduan izendatu.
 - **NodoHoberena** **Irekiak** zerrendatik ezabatu eta **Itxiak** zerrendan gorde.
 - Begiratu ea **NodoHoberena** helburu nodoa den, horrela bada AMAITU, soluzioa aurkitu da.
 - Bestela, **NodoHoberena** nodoaren ondorengo nodoak sortu, baina **NodoHoberena** beraiei apuntatzen ipini gabe, lehenago hauetakoren bat aurretik sortua izan den begiratu behar da.
 - Ondorengo bakoitzarentzat OndorengoTratatu() funtzioari deitu.
- Algoritmoaren amaiera

OndorengoaTratatu() funtzioaren sasikodea:

- **Ondorengoa**, **NodoHoberenari** apuntatzen jarri. Atzeraka eginiko lotura hauek ahalbidetzen dute emaitza bilatu eta gero, erabaki den bidea errekuperatzea.
- Kalkulatu: $g(\text{ondorengoa}) = g(\text{NodoHoberena}) + \text{NodoHoberenatik ondorengora joateko bidearen kostua}$.
- Begiratu **ondorengoa** ea **Irekiak** zerrendan dagoen. Baiezkoa bada, **Zahar** moduan izendatu **Irekiak** zerrendan dagoen nodoa. Nodo hori grafoan existitzen denez, baztertu egingo da eta **Zaharra** **NodoHoberenaren** ondorengo zerrendan gorde. Erabaki **Zaharraren** gurasoa loturak **NodoHoberenari** apuntatu behar dion. Horrela izan beharko litzateke **ondorengorarte** bilatu den bideak, **Zaharrerarte** aurkitu zen bide hoberenak baina kostu gutxiago badu, ondorengoa eta **Zaharra** nodo bera baitira. Egiaztatzeko gutxiago kostatzen dela **Zaharrera** iristea bere uneko gurasotik edo ondorengora **NodoHoberenetik**, beraien g balioak konparatzen dira. **Zaharrak** kostu txikiagoa edo bera badu ez da ezer egin behar. **Ondorengoak** kostu txikiagoa badu, orduan **Zaharraren** guraso lotura **NodoHoberenari** apuntatzen ipini, $g(\text{Zaharra})$ -n bide optimo berria gorde eta $f(\text{Zaharra})$ eguneratu. **Ondorengoa** existitzen denez, sortutako nodo berria ezabatu.
- Begiratu ondorengoa **Itxiak** zerrendan dagoen. Horrela bada, zerrendako nodoa **Zaharra** bezala izendatu. Erabaki, **Zaharraren** guraso loturak **NodoHoberenari** apuntatu behar dion. Horretarako g balioak konparatzen dira. Ondorengoak kostu hobe badu, **Zaharraren** guraso lotura **NodoHobererena** apuntatzen ipini, bide optimo berria $g(\text{Zaharra})$ -n gorde eta $f(\text{Zaharra})$ eguneratu. Zaharraren ondorengoetara g hedatu eta ondorengo nodoa ezabatu.
- Ondorengoa ez badago **Irikiak** zerrendan ezta **Itxiak** zerrendan ere, **Irekiaken** gorde eta **NodoHoberenaren** ondorengo zerrendan ipini. Kalkulatu $f(\text{ondorengoa}) = g(\text{ondorengoa}) + h(\text{ondorengoa})$
- Amaitu da **OndorengoaTratatu()**.

A* algoritmoak zabalerako bilaketa egiten du eta soluzio bat existitzen bada, beti aurkituko du. Grafoko n nodo guztientzat $g(n) = 0$ betetzen bada bilaketa handia egin beharko da, kostu gutxieneko nodoetan sakonerako bilaketetatik abiatuta. Grafoko n nodo guztientzat $h(n) = 0$ bada, kostu uniforme edukiko du eta programazio dinamikoaren antzeko kasu batean egongo da.

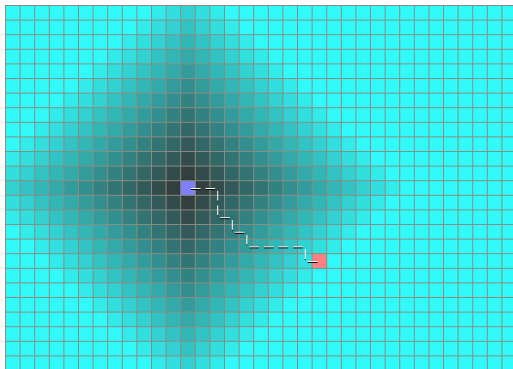
Algoritmoaren optimotasuna bermatzeko, $h(n)$ funtzioak optimista izan behar du, hau da, funtzio honek ezingo ditu benetan helburu nodora iristeko dagoen kostua baina estimazio handiagoak egin.

3.6.4. Konparaketa

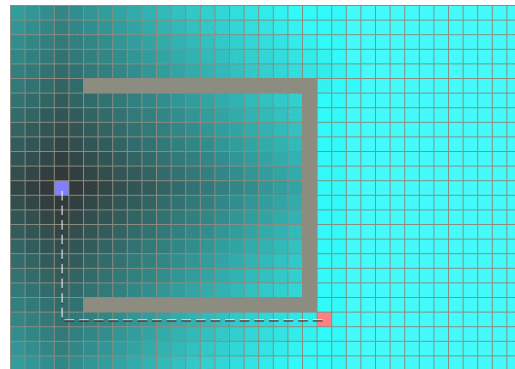
Bi algoritmoak alderatuta ikus daiteke konputazionalki oso kostu ezberdina dutela bi algoritmoek.

Programazio dinamikoak ingurunearen gelaxka guztiak aztertzen ditu, gelaxka bakoitzean egin beharreko mugimendu egokiena erabakiz. Algoritmoak soluzio egokiena aurkituko du, baina kostu konputazional handia edukiko du. Esan daiteke indar hutsa erabiltzen duela soluzioa aurkitzeko. Duen abantaila da ez duela heuristikorik erabiltzen eta, ondorioz, heuristikorik ez badaukagu erabilgarria da.

Jarraian dauden irudietan (3.14. eta 3.15.) ikus daiteke, algoritmoak nola lauki guztiak aztertzen dituen eta azkenean helbururako bide egokiena aurkitzen duen. Koloretuta daude aztertuko lituzkeen gelaxkak, ilunagoetatik hasi eta argiagoen-tara; gogoratu algoritmoa helburu puntutik hasten dela lanean. Hasierako puntua kolore gorrixka da eta helburua, morea.

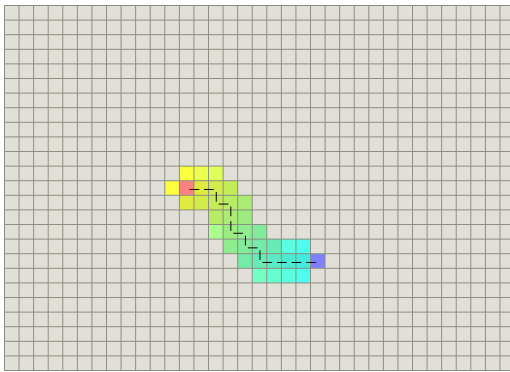


3.14. Irudia: Programazio dinamikoaren oztoporik gabeko ingurune batean

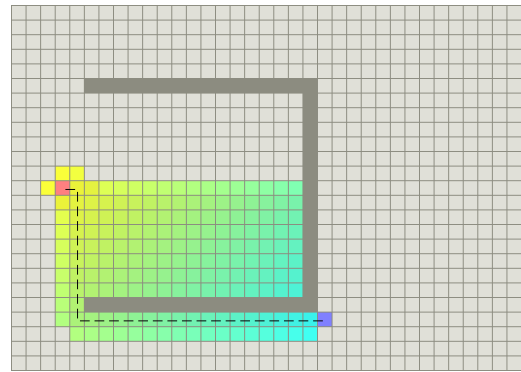


3.15. Irudia: Programazio dinamikoaren oztopo dagoen ingurune batean

A* algoritmoak, ez ditu inguruneko gelaxka guztiak aztertzen, ondorioz, konputazionalki azkarragoa da eta kostu gutxiago edukiko du. Gainera, erabilitako heuristikoa onargarria bada, emaitza optimoa edukiko dugu. Jarraian (3.16. eta 3.17.) irudietan ikus daiteke A* algoritmoak aztertuko lituzkeen gelaxkak aurreko ingurune beran, ikusi hasiera eta helburua lekuz aldatuta daudela adibide honetan, justu alderantziz.



3.16. Irudia: A* algoritmoa oztoporik gabeko ingurune batean



3.17. Irudia: A* algoritmoa oztopoa dagoen ingurune batean

3.6.5. Ondorioak

Bi adibideak ikusita, antzeman daiteke bi algoritmoak bidea aurkitzeko gai izan arren, A* dela algoritmo eraginkorrena. Bide optimoa azkarrago aurkituko baitu. Hori bai, zuzen funtzionatzeko heuristikoa onargarria behar du, heuristikoa ez bada onargarria, biek emaitza antzekoak edukiko dituzte.

Heuristikoa eskuratzea ez da beti erraza, batzuetan oso garestia izan daiteke ingurune batentzat heuristikoa aurkitzea, kasu horietan Programazio dinamikoa erabiltzea da aukera bat.

3.7. Robotaren mugimendua

3.7.1. Sarrera

Aurreko atalean, robot batek puntu batetik beste batera joateko egin beharreko ibil-bidea kalkulatzeko erabil daitezkeen teknika batzuk aurkeztu dira. Hemen, bidea kalkulatu eta gero, bide hori jarraitzeko erabil daitezkeen teknika bat deskribatuko da. Bidea erabaki eta gero, lehenengo gauza bide hori leuntzea da, horretarako 3.7.1 azpiatalean deskribatzen den leuntze (*smoothing*) teknika erabiltzen da. Leuntze honen helburua robotarentzat bidea erraztea da, alperrikako norabide aldaketa bortitzak eta alperrikako distantziak ekidinez.

Behin egin beharreko bidea leunduta, robota bertatik ahalik eta ondoen joateko PID kontrola erabili daiteke, 3.7.2 atalean deskribatu dena.

- $r(t)$ erreferentzia sarrera.
- $y(t)$ kontrolatutako aldagaia.

Kontroladore mota honek errore seinalea handitu ahala kontrol seinalea handitzen du, eta errore seinalea txikia denean kontrol seinalea txikitzen du. Kontroladore hau bakarrik erabiltzearen desabantaila, egoera gerakorreko errorea sortzen duela da, eta, ondorioz, kontrolaturiko aldagaia ez da sarrera erreferentziaren berdina izatera iritsiko. Honek 0 ez den errore seinalea sortuko du.

Kontrol Proporzional Integrala (PI)

Integralaren ekintzarekin, kontrol seinalea, errore seinalearen integralarekiko proporzionala da, 3.31 ekuazioan ikus daitekeen moduan. K_I integralaren irabazia da.

$$u(t) = K_p e(t) + K_I \int_0^t e(t) dt \quad (3.31)$$

P kontroladorearen arazoa egoera gerakorreko errorea da eta honi integratzaile bat gehituta kontrol sisteman, sistemak kontrolatutako aldagaiaren balioa handitu edo gutxitu dezake seinalearen errorea 0 izan arte. Errorea ezabatzean integralaren zatiak martxan egoteari uzten dio. Errorea ezeztatzeko denbora behar da eta honek egoera gerakorreko iristeko denbora luzatuko du, erantzun egokia atzeratuz.

Kontrol Proporzional Deribatiboa (PD)

Deribatibo ezaugarria, errore seinalea deribatuz lortzen da eta hau K_D konstante deribatiboarekin biderkatzen da, 3.32 ekuazioan ikus daitekeen bezala.

$$u(t) = K_p e(t) + K_D \frac{de(t)}{dt} \quad (3.32)$$

Deribatibo atalak sistema egonkorragoa egiten du. Errore seinalearen aldaketa abiaduran eragiten du eta errore seinalea oso handia egin baino lehenago, zuzenketak egin ditzake.

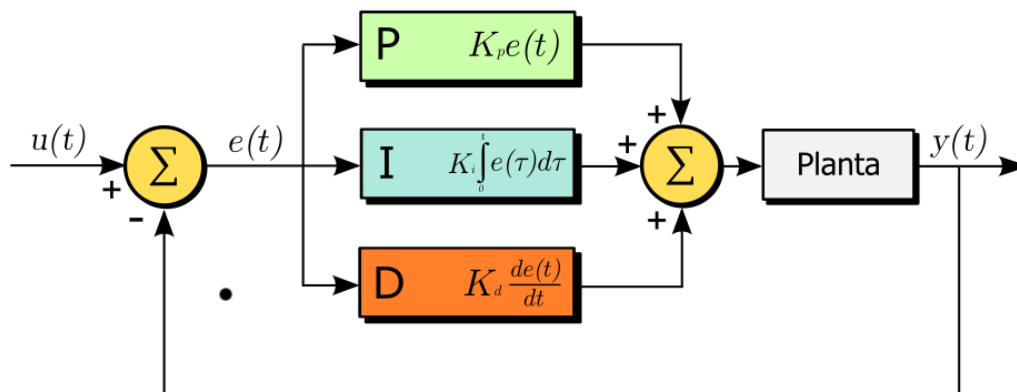
Kontrol Proporzional Integral Deribatiboa (PID)

Kontroladore honen aurretik aipatutako hiru kontrolatzaileen abantailak biltzen ditu. Diseinuan ez da bakarrik erreferentziaren sarrera kontuan eduki behar, ahal den neurrian perturbazioa eta zarata ekidin behar dira. Hiru kontroladoreek

proportzionalak, integralak eta deribatiboak, errore seinalearen gainean lan egiten dute. 3.33 ekuazioan deskribatzen da kontroladorea.

$$u(t) = K_p e(t) + K_I \int_0^t e(t) dt + \frac{de(t)}{dt} \quad (3.33)$$

3.19. irudian kontrolatzaile proportzional integral deribatiboaren bloke-diagrama ikusi daiteke.



3.19. Irudia: Kontrol proportzional integral deribatiboaren bloke-diagrama

4. Kapituluia

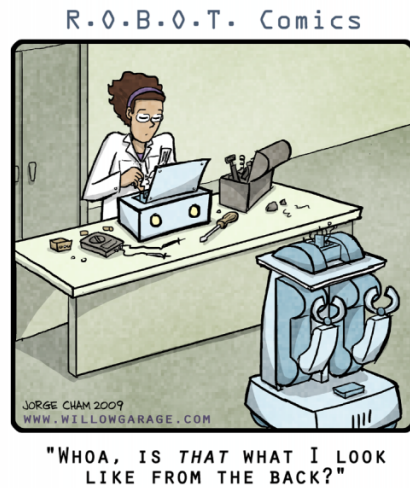
Pertsonen jarraipena robot mugikor batean

Gaien Aurkibidea

4.1. Sarrera	48
4.2. Robotaren Egitura	49
4.2.1. Sentsoreen ezaugarriak	49
4.2.2. Hanken Detekzioa	52
4.2.3. Txalekoaren Detekzioa	54
4.2.4. Detekzio Termikoa	55
4.2.5. Fusioa Partikula-Filtroa erabiliz	56
4.2.6. Pertsonen jarraipenerako partikula-filtroa	58
4.2.7. Kontrol modulua	59

4.1. Sarrera

Atal honetan proiektuaren alderdi praktikoa egiteko erabili den robotaren egitura adieraziko da: sentsoreak eta robotaren portaera inplementatzeko erabili diren teknikak [7].



Mundu errealean, gizakiekin hartu-emana edukiko duen robot mugikor bat garatu nahi denean, garatuko den robotak eduki behar duen ahalmen nagusia, gizakien detekzio eta jarraipena modu seguru batean egiteko gaitasuna da. Jarraian dauden atazetan, **pertsonen jarraipena** portaera seguru bat garatzeko izan daitezkeen arazoak eta arazo horiek gainditzeko har daitezkeen erabaki batzuk azalduko dira. Robotak helburu bezala duen pertsona detektatu duenean, zuzenean bera dagoen lekura joango da. Hori lortzeko robotak, sentsore ezberdinetatik jasotako informazioa erabiltzen du, datu horiek interpretatu eta zuzenean erantzun bat emateko.

Proiektu honetan erabiliko den robotaren gako nagusietako bat, kostu gutxiko sentsoreak erabiltzea da, gizakien jarraipen sistema bat garatzeko. Azken aldian, Microsoft-ek zenbait ikerketa egin ditu gizakien posizio eta mugimendu detekziorako sentsore eta softwarearen inguruan eta ikerketa horiek emaitza onak izan dituzte. Kinect, Microsoft-ek garatutako mugimendu sentsoreak, ikusmen eta sakonera datuak eskaintzen ditu eta bere errendimendurako kostu nahiko baxua duela esan daiteke. Kinect-a robotika mundurako berrikuntza ona izan arren, murriztapen batzuk ere baditu. Batetik, sakonera mapa sentsoretik 80cm baina hurrunago dauden objektuetarako soilik da erabilgarria. Bestetik, Kinect-ak IR kamera¹ erabiltzen du eta, ondorioz, eguzki argiak arazoak sor ditzake, kontuan edukiz eguzki izpiak IR espektroan daudela. Bestalde, aipatu behar da Kinect-ak

¹(IR, *Infrared*, euskaraz kamera infragorria)

gizakien detekziorako erabiltzen dituen algoritmoak kamera estatiko batentzat diseinatuak izan direla eta, ondorioz, ez dira eraginkorrak robot mugikorretan. Robot mugikorretan sentsorea robotean txertatuta egoten da eta hau normalean mugimenduan dagoenez, robotak dinamikoki aldaketak jasango dituen ingurune batean lan egin beharko du, eta honek arazoak sortuko dizkio Kinect kameraren algoritmoari.

Hori guztiari aurre egiteko, proiektuan erabiliko den robotak sentsore ezberdinak konbinatzen ditu: Kinect-a, sentsore termiko bat (Heimman HTPA *thermal sensor*) eta Hokuyo laser sentsore bat; sentsore horiek konbinatuko dira, gizakien detekzio sistemaren sendotasuna hobetu daitekeen ustearekin. Sentsore termikoak ikusmen sentsoreak izan ditzakeen arazoak konpontzen lagunduko du, hala nola, gizaki forma izan dezaketen objektuak gizaki ez kontsideratzea ahalbidetuko du, objektu horien tenperatura aztertuta. Sentsore honen beste abantaila bat, tenperatura datuak ez daudela argiaren menpe da eta honek iluntasunean gizakiak detektatzea ahalbidetzen du. Bestalde, detekzio faltsuak egin ditzake, gizakien tenperatura antzekoa duten makinekin edo berogailuekin.

4.2. Robotaren Egitura

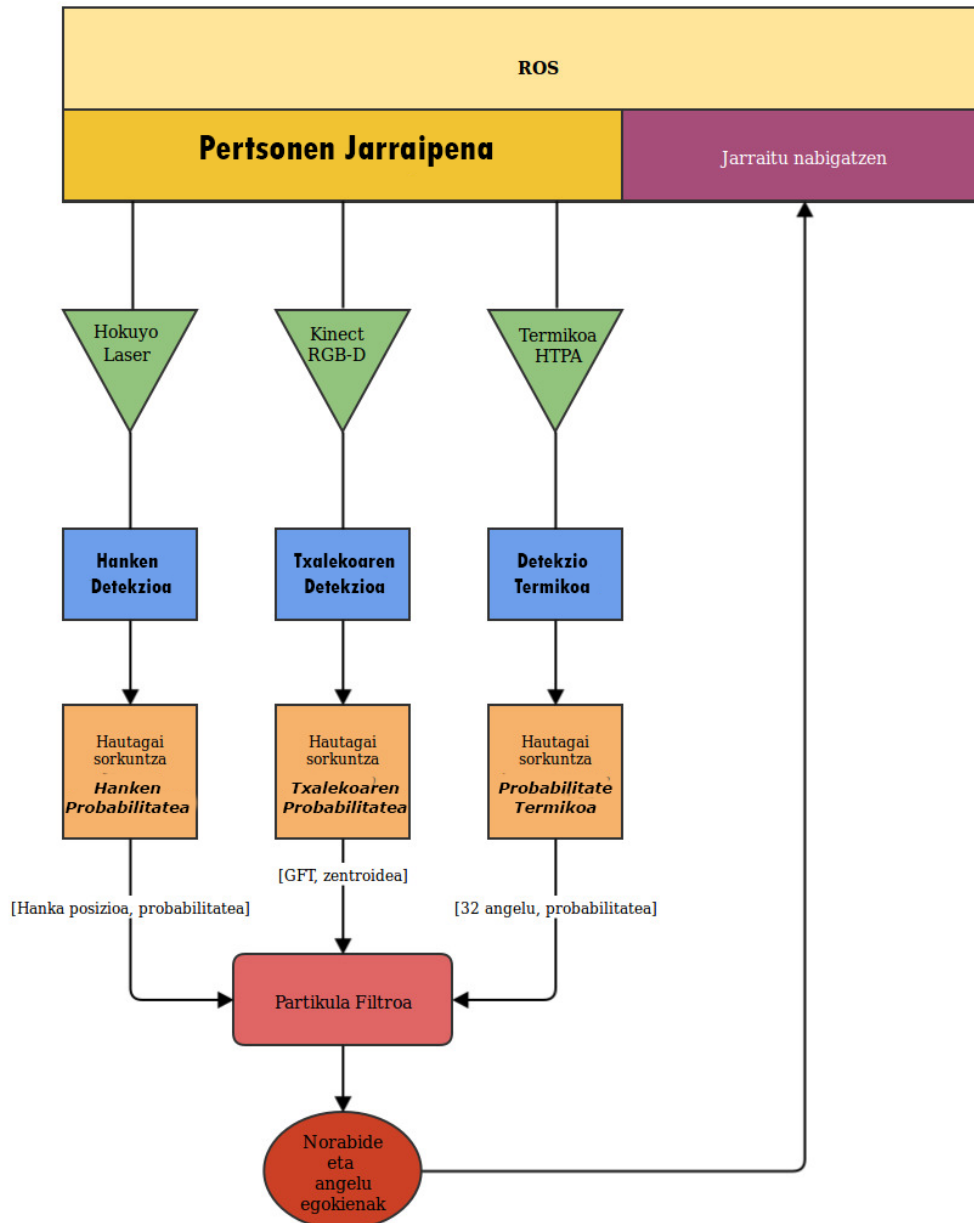
Robotak ikuspuntu multimodal bat edukiko du eta bere ezaugarriak, prozesamendu paraleloa eta sentsoreen bidez jasotako datuak filtratzea izango dira, ikusi [4.1.](#) irudia. Robota ROS sisteman oinarritzen da.

Pertsonen jarraipen sistema, Hokuyo laser sentsore batean, Heimannek garatutako HTPA sentsore termiko batean eta Kinect sentsore batean oinarritzen da, guztia RMP Segway plataforma mugikor batean muntatu delarik, ikusi [4.2.](#) irudia.

Sentsore bakoitzak dagokion detekzio unitatearentzat datuak biltzen ditu, hiru detekzio unitate daude inplementatuta: **Hanken Detekzioa**, **Txalekaren Detekzioa** eta **Detekzio Termikoa**. Detekzio unitate bakoitzak helburu-hautagaiak bidaltzen dizkio partikula-filtroari eta, bertan, jasotako informazioa fusionatzen da. Informazioa fusionatuta, partikula guztietatik hautagai hoberena aukeratzen da, robotaren helburu puntua izango dena, robota puntu horretara abiatu eta pertsonaren jarraipena egin dezan.

4.2.1. Sentsoreen ezaugarriak

- Sentsore termikoa:



4.1. Irudia: Robotaren egitura, hiru datu sarrera eta erabakiak hartzeko partikula-filtroa

- HTPA sentsore termikoak inguruneko tenperaturaren neurketa ahalbidetzen du. Neurketa hauetan bereizmen handiak ez dira beharrezkoak, adibidez: pertsonen detekzioan, gainazal kritikoen temperatura neurketan, suteen detekzio-puntuetan, energia-mantenuan edo segurtasun aplikazioetan. *Array* termofiloak radiazio infragorria detektatzen du eta, ondoren, informazio hori irudi bihurtuko da. Irudiaren pixel bakoitzak tenperatura balio bat izango du. Sentsoreak 32x31 irudi bat soilik



4.2. Irudia: Robotaren egitura, Segway RMP-a sentsoreekin

eskaintzen du, beraz, inguruneko gutxi gorabeherako tenperaturaren irudi bat lortzen da. Teknologia honen abantailetako bat energiaren kontsumo txikia da; beste bat, sistemaren sentsibilitate handia izan daiteke.

Ikusi [4.3.](#) irudia, bertan sentsore termikoa ikus daiteke.

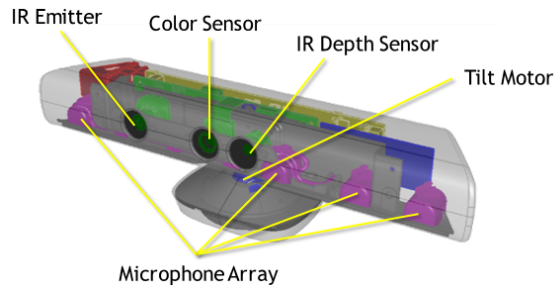


4.3. Irudia: Robotaren egitura, HTPA sentsore termikoa

- **Kamera:**

- Kinect-ak, irudi arruntaz gain, sakoneraren informazioa ere eskaintzen du. Jasotako sakoneraren informazioarekin sakonera irudi bat sortzen da, ondoren, argi infragorria erabiliz sujetua argiztatzeko eta bi IR sentsoreek jasotako informazioaren desberdintasuna neurtzen da. Kinect-ak 640x480 distantzia (sakonera) mapa bat itzultzen du denbora errealean (30fps). Sakoneraz gain, esan bezela, Kinect-ak 640x480 RGB irudia ere eskaintzen du.

Ikusi 4.4. irudia, bertan Kinect kamera baten atalak ikus daitezke eta dituen sentsoreak.



4.4. Irudia: Robotaren egitura, Kinect sentsorea

- **Laserra:**

- Hanken detekzioa egiteko Hokuyo UTM-30LX laser bat erabiltzen da. Laser honek 270 gradutako atzipen angelua dauka, 0.1 eta 30 metro arteko sakonera neurtzen du eta 0.25 gradoko angelu bereizmena dauka (1080 irakurketa ekorketa bakoitzean).

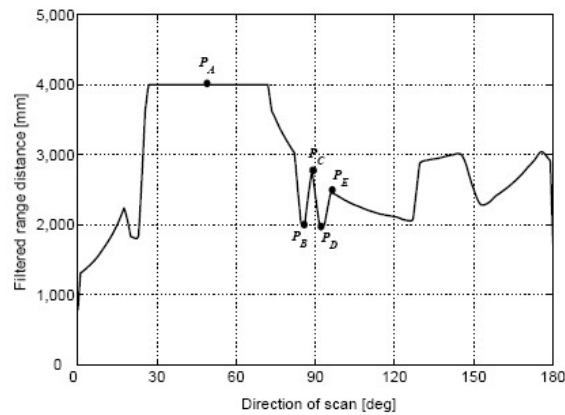
4.5. irudian laserraren argazkia ikus daiteke.



4.5. Irudia: Robotaren egitura, Hokuyo laser sentsorea

4.2.2. Hanken Detekzioa

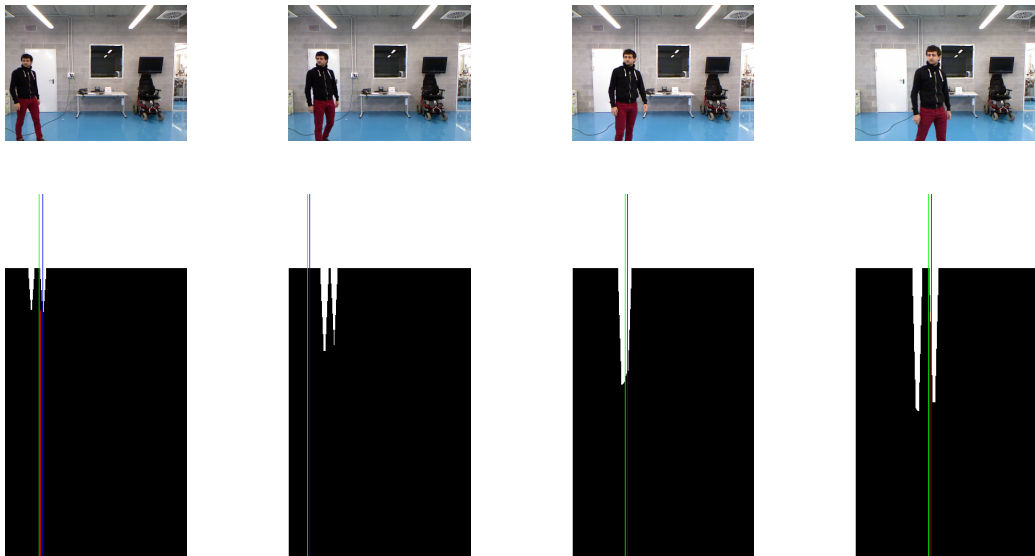
Robotean erabiltzen den **Hanken Detekzio** sistemak $max > min > max > min > max$ sekuentzia bat erabiltzen du, robota laserraren irakurketetan sekuentzia hori bilatzen saiatzen da eta, aurkituz gero, hanka detektatzen du, beti ere parametro batzuen menpe. 4.6. irudian ikus daiteke aipatutako sekuentzia definitzen duen patroia, bertan 5 punturekin markatuta daude aipatutako puntuak: P_A, P_B, P_C, P_D



4.6. Irudia: Hankak detektatzeko erabiltzen den patroia

eta P_E . Puntu horietan oinarrituta neurri ezberdinak daude definituta, esaterako: hanken arteko distantzia edota hanka eta atzealdearen artekoa.

4.7. irudi segidan ikus daiteke, laserrak nola detektatzen dituen hankak eta nola egiten dien jarraipena, aurretik adierazitako patroia eta Kalmanen filtroa erabiliz.



4.7. Irudia: Laserra hankak detektatzen

4.2.3. Txalekoaren Detekzioa

Aurretik esan bezala, RGB irudiak erabiltzen dira helburu posibleak arakatu eta partikula-filtroari informazioa bidaltzeko. Helburuaren posizioa **Txalekoaren Detekzioak** estimatuko du eta informazio hau edukiko du: robotaren zentrorra dagoen distantzia, robotaren paretik dagokion angelua, eta bere altuera. Erabili den partikula-filtroaren egitura kontuan edukita, soilik distantzia eta angelua erabiliko dira kalkuluetan.

Txalekoaren Detekzioaren metodoa, larrialdietako txalekoa jantzita daramaten pertsonekin erabiltzeko diseinatua da. Metodo honetan, RGB filtro bat erabiltzen da irudian agertzen de larrialdi txalekoaren kolore horian zentratzeko, filtroak hasierako irudi koloretsutik irudi bitar bat lortzen du, non pixel txuriek helburu kolorea, kasu honetan txalekoaren kolore horia, errepresentatzen duten. Ondoren, irudi bitarra filtratzen da, zonalde txuri txikiak ezabatuz, dilatazio morfologikoak eginez eta higadura eragiketa batzuk aplikatuz. Emaitzak 4.8. eta 4.10. irudietan ikus daitezke. Zonalde zuri bat baino gehiago badaude irudian, haundiena aukeratzen da. Aukeratutako zonaldea, irudian zenbait ezaugarri lortzeko erabiltzen da, ertzak eta beste balio batzuk, lortutako irudian jarraipena egiteko. Fluxu optikoa² kalkulatzeko, detektatutako ertzak eta Lucas-Kanade metodoa erabiltzen dira. Fluxu optikoa kalkulatu eta gero lortzen den ertzen arteko zentroidea izango da helburuaren estimatutako posizioa.

Ikusi 4.9. eta 4.11. irudiak, hauetan zentroidea agertzen da marraztuta. Jarraian dauden bi irudi multzoetan ikus daiteke bat datozela txalekoa, irudi bitarra eta kalkulaturako zentroidea.



4.8. Irudia: Irudia filtratu eta gero lortzen den irudi bitarra



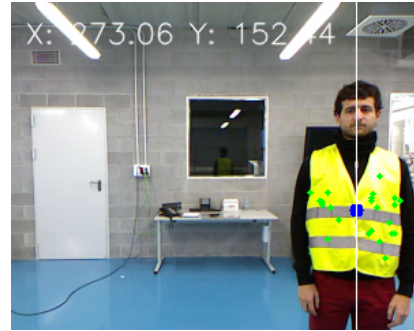
4.9. Irudia: Zentroidea kalkulatu eta gero lortzen den irudia

Tracking-aren segurtasuna handitzeko erroreen aurrean, bereziki detektaturiko pertsonak biratu edo bere ikuspuntua aldatzen duelako, irudiaren ezaugarriak birkalkulatu dira, distantziak aldatu edo puntu batzuk desagertzen direnean.

²Fluxu optikoa behatzaile baten (begi edo kamera) eta eszenaren arteko mugimendu erlatiboak eragindako objektu, gainazal eta ertzen itxurazko mugimenduen patroia da



4.10. Irudia: Irudia filtratu eta gero lortzen den irudi bitarra, beste posizio batean



4.11. Irudia: Zentroidea kalkulatu eta gero lortzen den irudia, beste posizio batean

4.2.4. Detekzio Termikoa

Pertsona batek, bere ingurunearekin alderatuz gero, perfil termiko ezberdina du. Detektatu den pertsona bati dagokion pixel batek, normalean, 37 grado Celsius inguruko balioa izango du, gehienetan pixka bat baxuagoa, azalaren gainean duen erroparen ondorioz.

Robotak inplementatuta duen prozedura batek, irudi termikoa hartuta 32 koma higikorreko zenbakiz osatutako bektorea kalkulatu du. Bektoreko n . elementua, pertsona n zutabean (31 elementuz osatutako zutabea) egotearen probabilitatearen estimazioari dagokio.

Kalkulu hori hiru urratsetan egiten da:

- Lehenengo, termopilaren irudiaren pixel bakoitzari pertsona bat izateko probabilitatea esleitzen zaio. Probabilitate hau esleitzeko kontutan edukitzen da pertsona baten tenperatura normalean μ batezbestekoa eta σ desbideratze estandarra dituen banaketa bat bezala adierazten dela. Batezbestekoa eta desbideratze estandarra kalkulatzeko, jendea sentsoare termikoaren parean ipinita hartu diren zenbait irudi erabili dira. Balio ezberdin asko probatu eta gero, emaitza egokienak batezbestekoa=36 eta deribatze estandarra=2 balioekin lortzen dira.
- Ondoren, probabilitate matrizea leuntzeko konboluzioa aplikatzen zaio matrizeari 5 pixeleko zabalera duen kernel Gaussiar batekin.
- Amaitzeko, n zutabeko maximoa, aurretik aipatutako probabilitate bektorearen n . elementua bezala esleitzen da.

4.2.5. Fusioa Partikula-Filtroa erabiliz

Behin **Hanken Detekzioa**, **Txalekoaren Detekzioa** eta **Detekzio Termikoa** definituta, beharrezkoa da informazio hori fusionatzea, pertsonen *tracking*-a burutzeko. Horretarako *Sequential Importance Resampling (SIR)* partikula-filtro bat erabiltzen da, hau bezalako ez-lineal ez-Gaussian arazoetan erabili ohi dena.

Arazoari begiratuta, t uneko egoera, x eta y posizio parearekin definitu daiteke, $X(t) = [x(t), y(t)]$. Ezin denez helburuaren mugimendu patroia definitu *a priori*, honela definitzen da trantsizio egoera:

$$X(t) = X(t-1) + V(t-1) \quad (4.1)$$

non $X(t-1)$ aurreko egoera bektorea den eta $V(t-1)$ prozesuaren zarata.

Behaketa, bestalde, **Hanken Detekzioak**, **Txalekoaren Detekzioak** eta **Detekzio Termikoak** ematen duten informazioarekin egiten da. Beraz, hiru informazio iturri horietan oinarrituz, partikula bakoitza **hanken probabilitate**, **txalekoaren probabilitate** eta **probabilitate termiko** batekin definituko da.

Azkenean, partikula-filtroen jarraipen prozedura, 4.12. irudiko algoritmoan algoritmoan deskribatzen da.

Algorithm 3: Pertsonen jarraipena egiteko algoritmoa

```

Eskatu :  $\{x_0^i, w_0^i\}_{i=1}^N, z_1$ 
Bermatu:  $\{x_k^i, w_k^i\}_{i=1}^N$  for each  $k$ 
1 forall the  $i=1:N$  do
2    $w_0^i = \frac{1}{N}$ ;
3 while jarraitu jarraitzen do
4   if  $N_{eff} < N_{threshold}$  then
5     Sortu  $N$  partikula aukeraketa ordezkapenarekin (selection with replacement) eginez
6   Aurreikuspena:  $x_{k+1}^i = x_k^i + v_k$ ;
7   Pisuak eguneratu:  $w_{k+1}^i = w_k^i P(x_{k+1}^i | z_{k+1})$ ;
8   Pisuak normalizatu:  $w_k^i = \frac{w_k^i}{\sum_{j=1}^N w_k^j}$ ;
9    $k = k + 1$ 

```

4.12. Irudia: Pertsonen jarraipena egiteko algoritmoa, partikula-filtroa erabiliz.

Hanken Probabilitatea

Aurretik aipatu bazala, **Hanka Detekzioak** laserraren irakurketak analizatuko ditu eta *array* bat itzuliko du, ezarritako atalasea (*threshold*) baina probabilitate handiagoa duten hanken angelu eta sakonerarekin.

Egoera bakoitzarentzat edo partikula bakoitzarentzat hanken probabilitatea kalkulatzeko, gertuen dagoen hanka detektatzen da **Hanken Detekzio** unitateak ematen duen informazioa erabiliz. Hanken posizioan oinarrituta honela kalkulatu da hankaren probabilitatea:

$$dist = \sqrt{r_{laser}^2 + r_t^2 - 2r_{laser}r_t \cos(\theta_{laser} - \theta_t)} \quad (4.2)$$

$$P_{hanka}(X_t|Z_t) = \Phi(r_{laser})e^{-dist} \quad (4.3)$$

non *dist* partikularen eta laserrak detektatutako hankaren arteko distantzia erradiala den. P_{hanka} 4.2.2 atalean deskribatu den hanken patroien probabilitatearen eta funtzio esponentzial baten arteko biderkaketaren bidez eginez kalkulatu da, balio txikiagoak itzuliz distantzia handitu ahala.

Txalekoaren Probabilitatea

Txalekoaren Detekzioak eskaintzen duen informazioa, kameraren probabilitatea kalkulatzeko ere erabiltzen da. Probabilitate hau irudiaren ezaugarriak eta fluxu optikoaren bidez estimatzen den angelua erabiliz kalkulatu da. Honela:

$$dist = (\theta_{txalekoa} - \theta_t) \quad (4.4)$$

$$P_{txalekoa}(X_t|Z_t) = e^{-dist} \quad (4.5)$$

non *dist* partikularen angeluaren eta detektatu den txalekoaren zentroidearen arteko distantzia den, eta $P_{txalekoa}$ distantziaren funtzio esponentzial moduan kalkulatu den.

Probabilitate termikoa

Probabilitate termikoa, **Detekzio Termikoak** ematen duen informazioa erabiliz kalkulatu da. Horretarako, partikula bakoitzarentzat, ebidentzia *array*-ean

atalasea gainditzen duen balio gertukoena detektatzen da eta, angelu horretan oinarrituz, probabilitatea horrela kalkulatzen da:

$$dist = (\theta_{termikoa} - \theta_t) \quad (4.6)$$

$$P_{termikoa}(X_t|Z_t) = \theta(r_{termikoa})e^{-dist} \quad (4.7)$$

non $dist$ partikula eta sentzore termikoak detektatutako pertsonaren arteko distantzia den eta, $P_{termikoa}$, pertsona baten profil termikoaren probabilitatearen eta funtzio esponentzial baten arteko biderkaketa den.

Azken Probabilitatea

Probabilitate guztiak kalkulatu eta gero, emandako egoera baten azken probabilitatea honela kalkulatzen da:

$$P(X_t|Z_t) = \alpha P_{hanka}(X_t|Z_t) + \beta P_{txalekoa}(X_t|Z_t) + \gamma P_{termikoa}(X_t|Z_t) \quad (4.8)$$

non,

- $P_{hanka}(X_t|Z_t)$ hanka probabilitatea den.
- $P_{txalekoa}(X_t|Z_t)$ txaleakoaren probabilitatea.
- $P_{termikoa}(X_t|Z_t)$ probabilitate termikoa
- α , β eta γ probabilitate bakoitzari ematen zaion pisua dira, sentzore horri eman nahi zaion garrantziaren arabera alda daitezke. Zenbat eta balio txikiagoa eduki koefizienteak orduan eta garrantzi gutxiago edukiko du azken emaitzan sentzore horrek jasotako informazioak, balioa handiagoa ipiniz gero, berriz, handiagoa. Balio hauekin jokatu da probak egiterako orduan, sistemaren portaera egoki baten bila.

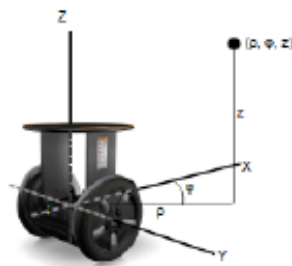
4.2.6. Pertsonen jarraipenerako partikula-filtroa

Pertsonen jarraipena partikula-filtro batekin egin ahal izateko, lehenengo, sistemaren egoera definitu behar da, planteatutako arazoa eta eskuragarri dauden sentzoreak kontuan edukita. Hasiara batean, robotaren biraketa ardatzean finkatutako koordinatu sistema zilindriko bat definitu da, 4.13. irudian ikusen dena.

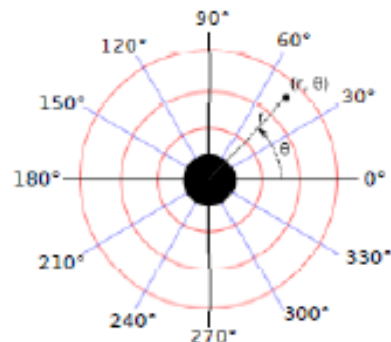
Koordenatu sistema honetan pertsona baten posizioa (ρ, φ, z) balioen bidez definitzen da. Non ρ robotaren zentroarekiko distantzia den, φ robotaren aurre direkzioarekiko angelua eta z altuera. Robotaren mugimenduak XY planoan izango direnez, z koordenatu bertikalak ez du jarraipena egiteko informazio interesgarririk eskaintzen, beraz ez da erabiliko.

Ondorioz, eta sentsore moduluak eskaintzen duten informazioa ikusita, angelua eta distantzia hartuko dira kontuan. Pertsona bat espazioan kokatzeko modurik onena koordenatu polarreko sistema bat erabiltzea dela iritzi da. Sistema mota hauetan planoko puntu bakoitza koordenatu bikote bati dagokio (r, θ) , non r puntua eta jatorriaren arteko distantzia den, eta θ ardatz polarrarekiko angelua, ikusi 4.14. irudia.

Erabilitako robotean, pertsonen jarraipena egiteko partikula-filtro bat erabiltzen da, non partikula bakoitzak r eta θ balioak dituen. Iterazio bakoitzean partikularen pisua aldatuko da sentsoreek jasotako informazioaren arabera.



4.13. Irudia: Koordenatu sistema zilindrikoa



4.14. Irudia: Koordenatu sistema polarra

4.2.7. Kontrol modulua

Jarraian robotaren kontrol modulua deskribatuko da. Robotak erabiltzen duen estrategia, helburu puntua beti zentratu edukitzea da. Hau da, robota beti saiartzten da helburu puntua parez pare edukitzen. Kontrol moduluak akzio/erreakzio moduko sistema bat erabiltzen du. Sarrera bezala partikula-filtroak pertsonaren kokapenaren inguruan egindako estimazioa erabiltzen du eta, horren arabera, abiadura komandoak sortuko ditu. Robota pertsonarengana zuzenean joan eta beraren aurka talkarik gertatuko ez dela ziurtatzeko (v, ω) abiadura linear eta abiadura angeluarrez osatutako komandoak erabiltzen dira. Abiadura horiek

kalkulatzeko, jarraian dauden funtzioak definitu dira:

$$v = \begin{cases} 0 & \text{baldin } dist \leq dist_{min} \\ \frac{1+e^{-k*(dist-dist_{min})}}{1-e^{-k*(dist-dist_{min})}} * v_{max} & \text{baldin } dist_{min} < dist < dist_{max} \\ v_{max} & \text{baldin } dist \geq dist_{max} \end{cases} \quad (4.9)$$

$$|\omega| = \begin{cases} 0 & \text{baldin } ang \leq ang_{min} \\ \frac{1+e^{-k*(|ang|-ang_{min})}}{1-e^{-k*(|ang|-ang_{min})}} * \omega_{max} & \text{baldin } ang_{min} < ang < ang_{max} \\ \omega_{max} & \text{baldin } ang \geq ang_{max} \end{cases} \quad (4.10)$$

Abiadura linealaren kalkulurako $dist_{min}$ atalasea definitu da, bertan robotaren eta pertsonaren arteko distantzia minimoa adierazten da eta, ondorioz, pertsona atalase hori baina gertuago badago, robota ez da linealki mugituko. Distantzia minimo horrekin batera, distantzia maximo bat ere definitzen da $dist_{max}$ eta, distantzia hori gaindituz gero, robota abiadura maximoan v_{max} mugituko da.

Pertsona eta robotaren arteko distantzia $dist_{min}$ eta $dist_{max}$ artean dagoenean funtzio sigmoide bat erabiltzen da. Funtzio honek abiadura lineala $dist$ distantziarekiko "S" forman handitzea eragiten du eta k parametroak hazkundearen malda doitzea ahalbidetzen du. k -ren balioa geroz eta handiagoa izan, malda ere handiagoa izango da.

Abiadura angeluarra kalkulatzeko antzeko funtzio bat erabiltzen da, 4.10 ekuaizioan ikus daitekeen moduan. Kasu honetan, ang_{min} eta ang_{max} atalaseak definitu dira. Robotaren eta pertsonaren arteko angelua ang_{min} baina txikiagoa bada, robota ez du bere orientazioa aldatuko. Angeluak ang_{max} atalasea gainditzen badu, robota abiadura angeluar maximoarekin ω_{max} mugituko da. Angelua ang_{min} eta ang_{max} tartean dagoenean, aurretik aurkeztutako funtzio sigmoidea erabiliko da.

5. Kapituluia

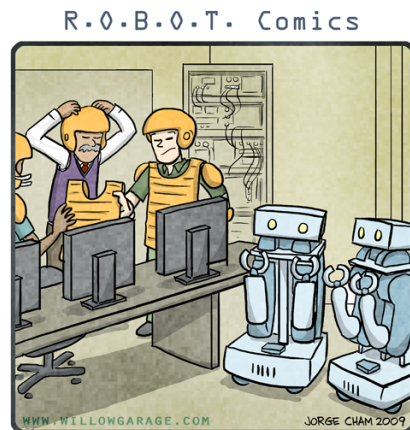
Esperimentazioa

Gaien Aurkibidea

5.1. Sarrera	62
5.2. Proba 1	62
5.2.1. Helburua	62
5.2.2. Ingurunea	63
5.2.3. Ondorioak	63
5.3. Proba 2	67
5.3.1. Helburua	67
5.3.2. Ingurunea	67
5.3.3. Jasotako datuen azterketa	68
5.3.4. Robotaren abiadura ezberdinekin probak	69
5.3.5. Partikula-Filtroaren partikula kopuru ezberdinekin probak	75
5.3.6. Sentsoreetan pisu ezberdinekin probak	81
5.3.7. Zirkuitu borobil batean probak	90
5.3.8. Ondorioak	95
5.4. Proba 3	97
5.4.1. Helburua	97
5.4.2. Ingurunea	97
5.4.3. Galdetegia	97
5.4.4. Emaitzak	99
5.4.5. Ondorioak	101

5.1. Sarrera

Atal honetan proiektuan zehar egin diren proba ezberdinak deskribatuko dira: proba bakoitzaren helburua, zein ingurunetan egin den eta proba horretatik ateratako ondorioak.



"I HAVE A BAD FEELING
ABOUT THIS DEMO."

5.2. Proba 1

Proba hau ikasketa prozesua amaitu eta egin den lehena izan da.

5.2.1. Helburua

Proba honek helburu ezberdinak ditu, baina helburu nagusia teorian ikasitako zenbait gauza praktikan jartzea izan da, hala nola:

- ROS sistema batekin lehen kontaktua izatea, teorian ikasitako zenbait gauzak praktikan nola funtzionatzen duten ikustea.
- Sentsoreekin lan egitea, sentsoreak martxan nola ipintzen diren ikusi, zein datu itzultzen dituzten eta datu horiek ROS sistema batean nola maneiatu daitezkeen aztertzea.
- Robotaren egitura atalean (4.2 atalean) deskribatutako zenbait kontzeptu ulertzea.

- Robotaren kodea begiratu eta ulertzea, erabiliko den ingurunean erabili ahal izateko aldaketaren bat behar badu, egin ahal izateko.
- Pertsonen jarraipena nola egin daitekeen praktikan ikustea, erabiliko diren sentsoarekin eta partikula-filtroarekin pertsonen jarraipena martxan jartzea.

5.2.2. Ingurunea

Proba hau egiteko Kinect kamera bat eta Hokuyo laser bat erabili dira. Laserra ez da 4. atalean deskribatutakoa izan eta, ondorioz, hanken detekzioan aldaketa txiki batzuk egin behar izan dira.

Kinect kamera eta laserra, egitura finko batean ipini dira, eta beraiekin batera hanken detekzioa (4.2.2 atalean deskribatua) eta txalekoaren detekzioa (4.2.3 atalean deskribatua) probatu dira zuzen funtzionatzen zutela ikusteko. Hau eginda partikula-filtroarekin bien datuak fusionatu dira eta proba ezberdinak egin dira, pertsonen jarraipenak portaera egokiena izateko sentsoe bakoitzari eman beharreko pisua ikusteko.

5.2.3. Ondorioak

Proba honetatik atera daitekeen lehen ondorioa hau da: sentsoeen portaera asko aldatzen dela ingurunearen arabehera. Hau da, argiak eragin oso handia duela Kinect kameran, eta laserrak ere portaera ezberdina duela jarraitu behar duen pertsonak jazten dituen kolorearen arabehera.

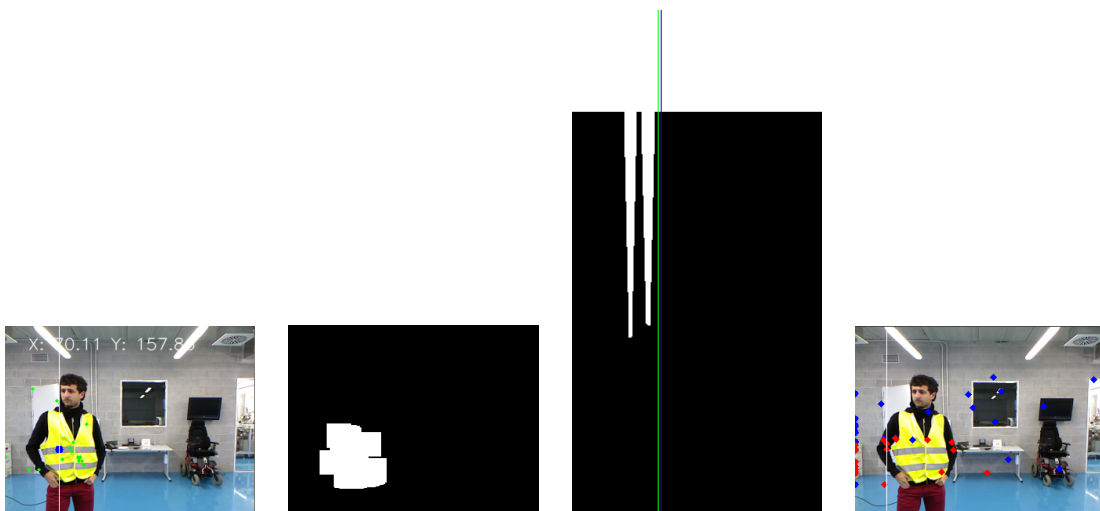
Pertsonen detekzioari dagokionez, proba ezberdinak egin eta gero, ondorioztatu da txalekoaren detekzioa eraginkorragoa eta fidagarriagoa dela hanken detekzioa baino. Hau ikusita eta 4.2.5 azpiatalean adierazitako funtzioak (4.8 ekuazioak) kontuan edukita, txalekoaren detekzioari partikula-filtroan pisu handiagoa ematea erabaki da, kontuan eduki behar da proba honetan ez dela detekzio termikoa erabili eta, ondorioz, bere pisua 0 izango da. Hau ondorioztatzeko hanken eta txalekoaren detekzioak bakarka probatu dira eta ikusi da, hanken detekzioak emaitza kaxkarragoak ematen zituela. Ondoren, proba ezberdinak egin dira pisu ezberdinak emanez detekzio bakoitzari, eta balio hauek erabiltzea erabaki da, emaitza onak eskaintzen zituztela ikusita:

- Hanken detekzioa: 0.2
- Txalekoaren detekzioa: 0.8
- Detekzio termikoa: 0

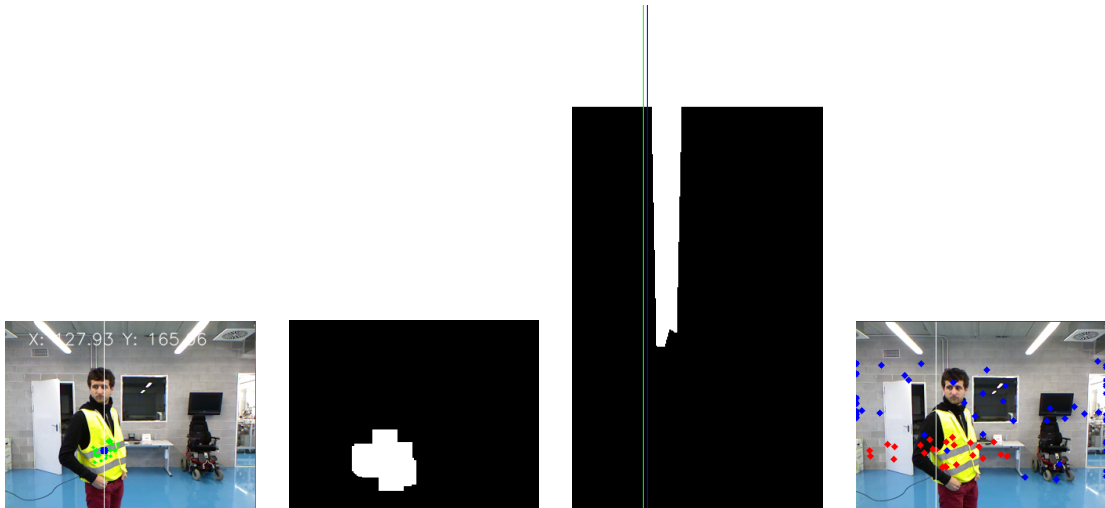
Jarraian dagoen irudi segidan (5.1,5.2,5.3,5.4,5.5.), eginiko probetako batean hartutako irudiak ikus daitezke. Irudi hauek denbota-sekuentzia jarrai batean jasotakoak dira. Ikus daiteke 4 irudi daudela ilara bakoitzean. Lehenengo irudian, txaleko detekziorako erabiltzen den zentroidea ikus daiteke. Bigarrenean, txalekoaren detekziorako erabiltzen den irudi bitarra ikus daiteke, txalekoa detektatu duen tokian kolore txuria duelarik. Hirugarren irudian, hanken detekzioa ikus daiteke, laserraren irakurketen bidez egiten dena. Laugarrenean, berriz, partikula-filtroa funtzionamenduan ikusten da, puntu gorriek pertsona detektatu dutela adierazten dute eta ikus dezakegu, pertsona mugitu ahala nola jarraitzen dioten puntu gorriek.

Irudietan ikus daiteke, denborak aurrera egin ahala, partikula-filtroaren emaitza hobea dela, hau da, puntu gorriak kontzentratuago daude helburuan. Hau partikula-filtroek duten propietate bat da, algoritmoaren iterazio kopuruak gora egin ahala, emaitza hobekia lortu ohi dira.

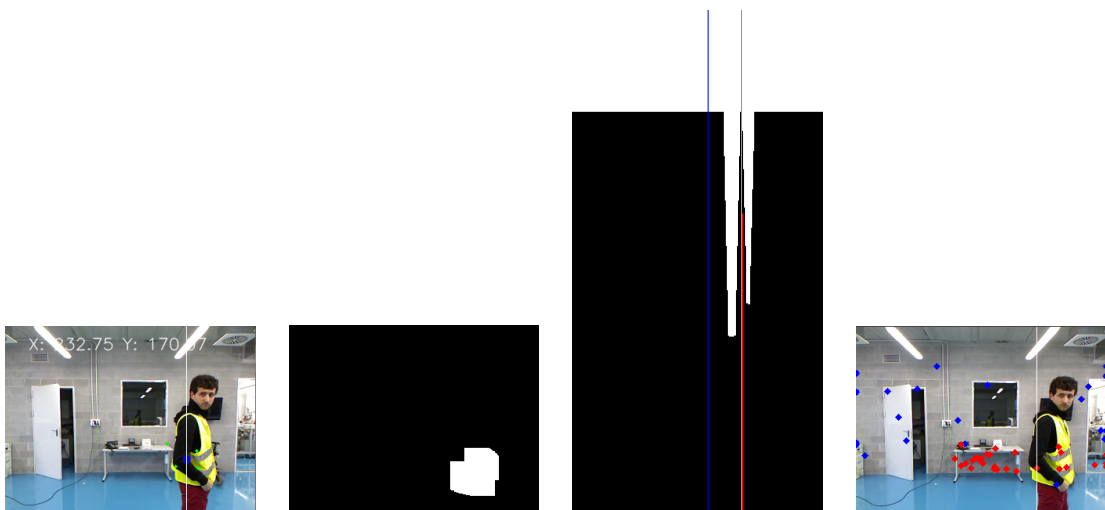
Bestalde, irudietan ere ikus daiteke, hanken detekzioak huts egiten duela une batzuetan, honek helburuaren posizioarekin du zerikusia, mugimenduan dagoenean, hanken detekzioa bilatzen ari den patroia galtzeko aukera baitago hanken posizioaren arabera. Hanken detekzioa galtzen duenean, gordeta geratzen da denbora batean, non zegoen helburuaren posizioa. Hau egiteko Kalmanen filtro bat erabiltzen da, hanken hurrengo posizioa zein izango den aurreikusteko eta jarraipena egiteko. Hanken posizioa galtzen denean, eta ezarritako denboran ez badu berriz aurkitu, kalmanen filtroa berrabiarazi egiten da.



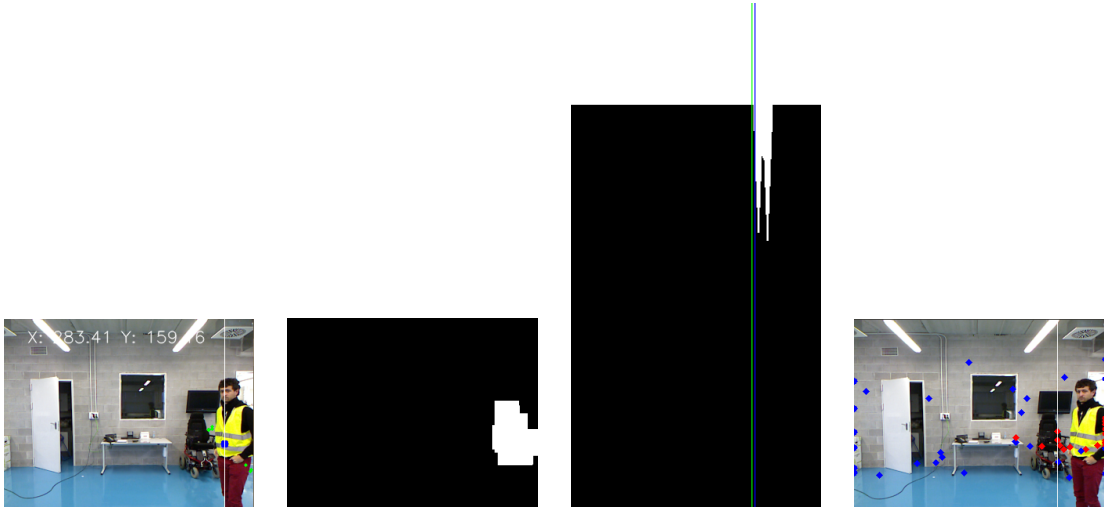
5.1. Irudia: t=1



5.2. Irudia: t=2



5.3. Irudia: t=3



5.4. Irudia: t=4



5.5. Irudia: t=5

5.3. Proba 2

Atal honetan Ktbot robotarekin egindako proba ezberdinak bildu dira, 4. kapituluari deskribatu den robotarekin hain zuzen ere.

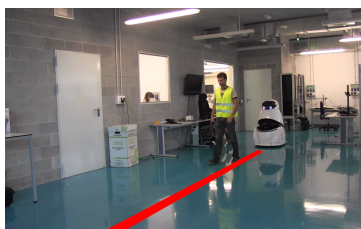
5.3.1. Helburua

Esperimentazio honen helburua robotaren portaera aztertzea izan da. Robotak dituen sentsoreetatik informazioa jaso eta pertsonen jarraipena nola egiten duen ebaluatu da. Horretarako, proba ezberdinak egin dira sentsoreei pisu ezberdinak esleituz, robotari pertsonen jarraipen segurua eta egokiena egitea ahalbidetuko dion konbinazioa aurkitu nahian.

Sentsore bakoitzari esleitu beharreko balioaz gain, beste aldagai batzuk ere aztertu dira, hala nola: robotaren abiadura lineala eta partikula-filtroak dituen partikula kopurua.

5.3.2. Ingurunea

Proba guztiak IK4-TEKNIKER eko Robotika laborategian egin dira bertan marraztutako zirkuitu zuzen bat 5.6. eta zirkuitu borobil 5.7. bat erabiliz. Probatetik datuak jaso eta hauek aztertu behar zirenez, lehenengo, proba guztiak zirkuitu zuzen batean egitea erabaki da eta, ondoren, emaitza esanguratsuenak izan dituzten balioekin proba berriak egin dira zirkuitu borobil batean.



5.6. Irudia: Zirkuitu zuzena



5.7. Irudia: Zirkuitu borobila

Proben datuak nahi bezala bildu eta, ondoren, interpretatu ahal izateko kodean aldaketa batzuk egin behar izan dira. ROS-en exekutatzen den kodean guri interesatzen zaizkigun datuak: termopilaren matrizearen balioak; kinektak eskaintzen duen irudia eta sakonerako irudiaren datuak; laserraren irakurketak; eta partikula-filtroak helburu puntu bezala aukeratu duen partikularen koordinatuak ROS-eko topic batetan publikatuko dituen nodo bat sortu da. Robota martxan dagoen bitartean eta pertsonen jarraipen sistema exekuzioan, garatutako nodoak aipatutako informazio guztia publikatuko du. Proba hauetan egin dena, nodo hori

publikatzen ari zen bitartean datu horiek guztiak grabatzea izan da, ondoren beste ordenagailu batean aztertu ahal izateko. Hau egiteko, ROS-ek eskaintzen duen **roslab** tresna erabili da, honek adierazten zaion topic batean argitaratzen den guztia grabatzen du .bag fitxategietan eta, ondoren, fitxategi horiek exekuta daitezke tresna berdina erabiliz datuen analisia egiteko.

5.3.3. Jasotako datuen azterketa

Esan bezala Ktbot robotarekin zenbait .bag fitxategi grabatu dira ondoren hauen azterketa egiteko. Esperimentazioan robotaren partikula-filtroak egiten duen pertsonaren posizioaren estimazioa eta pertsonaren benetako posizioa konparatu dira ondoren errorea kalkulatzeko.

Partikula-filtroaren estimazioan bi balio lortu dira: partikula hoberenaren angelua robotarekiko, eta sakonera, hau da, helburu pertsonara dagoen distantzia. (Hau 4.2.6 partikula-filtroa pertsonen jarraipenerako atalean azalduta dago).

Konparaketa egiteko bi balio behar dira, partikula-filtroaren estimazioa zuzenean eskuratu daiteke .bag fitxategietatik, baina benetako posizioa eskuratzea lan zailagoa izan da eta horretarako programa ezberdinak garatu behar izan dira. Lehenik eta behin, robotean grabatutako .bag fitxategiak martxan jartzean exekutatu beharreko programa bat garatu da, programa hau grabatutako fitxategien ROS-eko topic-etara harpidetzen da, hauek string moduan argitaratzen duten informazioa karpeta eta fitxategi ezberdinetan antolatzeko. Besteak beste, laser irakurketak, termopilaren balioak, kinectaren sakonera informazioa eta irudiak gordetzen ditu. Irudi hauek fotograma moduan gorde dira, horretarako irudien informazioa tratatu behar izan da. Fotograma hauek helburu pertsonaren benetako posizioa kalkulatzeko erabili dira. Garatutako script batekin pertsona kokatutako pixelen kalkulua automatizatu da eta, 5.1 ekuazio matematikoa erabiliz angeluak kalkulatu dira:

$$\alpha = \arcsin\left(\frac{\sin\left(\frac{FOV_camera}{2}\right)}{zutabe_zentral} * (-(x - zutabe_zentral))\right) \quad (5.1)$$

non:

- α = angelua
- x = posizioa pixeletan
- FOV_camera = Kinectaren ikusmen eremua 57,8°
- zutabe_zentral = 160 px

Kalkulu honekin angelua radianetan kalkulatzen da ondoren gradutara pasatzeko. Helburu pertsonak uneoro robotarekiko duen angelu erreala kalkulatuta, uneoro robota eta helburu pertsonaren arteko distantzia erreala kalkulatu behar izan da. Horretarako laser irakurketak erabiltzea erabaki da. Laserrak 90° tako irakurketa eremua du -45° -tatik 45° -tara helburu pertsonaren kokalekuaren arabera, robotaren parean gradua 0 izanik. Laserrak 90° horietan 360 irakurketa egiten ditu, hau da 4 irakurketa graduko. Kalkulu hauek automatizatzeko c++ eko programa bat garatu da. Programa honek pertsona une oro kokatuta dagoen angelu zerrenda bat jaso eta une oro angelu horri dagokion laser irakurketa itzultzen du. Kontuan edukita hanken detekziorako patroia, gerta daiteke pertsona kokatuta dagoen angelua, laserraren irakurketetan hanka tartean egotea eta, ondorioz, distantzia ez izatea jarraitzen ari den pertsona eta robotaren artekoa, baizik eta askoz handiagoa. Hau ekiditeko laser irakurketak 10° ko tarte batean egitea erabaki da, balio maximo eta minimoak ezabatu eta batezbesteko bat kalkulatu da, hori izango da robota eta pertsonaren arteko benetako distantzia.

Esperimentazioan, esan bezala, partikula-filtroaren estimazioaren errorea kalkulatu da. Konparaketa egiteko errore maximoa, errore minimoa, batez bestekoa eta errorearen desbiderapen estandarra kalkulatu dira. Grafikoki ere errepresentatu dira erroreak, proba kasu bakoitzarentzat angeluaren estimazioan eta benetako angelua konparatzen dituen grafiko bat egin da, baita estimatutako distantzia eta distantzia erreala konparatzen dituen ere.

Proba bakoitzarekin ebaluazio kuantitatibo bat, ebaluazio kualitatibo bat eta emaitzen interpretazio bat egin dira. Ebaluazio kuantitiboan proben zenbakizko emaitzak adierazi dira; kualitatiboan, robotak jarraipena egitean izandako sententzioa eta jarraipena ea behar bezala burutu den; eta, interpretazioan, izandako emaitzak arrazoitu dira eta emaitzetatik ondorioak atera.

5.3.4. Robotaren abiadura ezberdinekin probak

Proba honetan robotaren abiadura lineala ebaluatu nahi izan da, partikula-filtroaren estimazioekin duen erlazioa aztertzeko. Horretarako, robotari hiru abiadura lineal ezberdin eman eta ebaluazioa egin da, hauek izan dira abiaduraren balioak:

- 0.35 m/s
- 0.6 m/s
- 1.2 m/s

Proba hauek egiteko erabili den partikula kopurua 50 izan da eta sentsoreetan ipini diren balioak hauek izan dira: hanken detekzioa 0.2, txalekoaren detekzioa 0.6 eta detekzio termikoa 0.2.

Ebaluazio kuantitatiboa

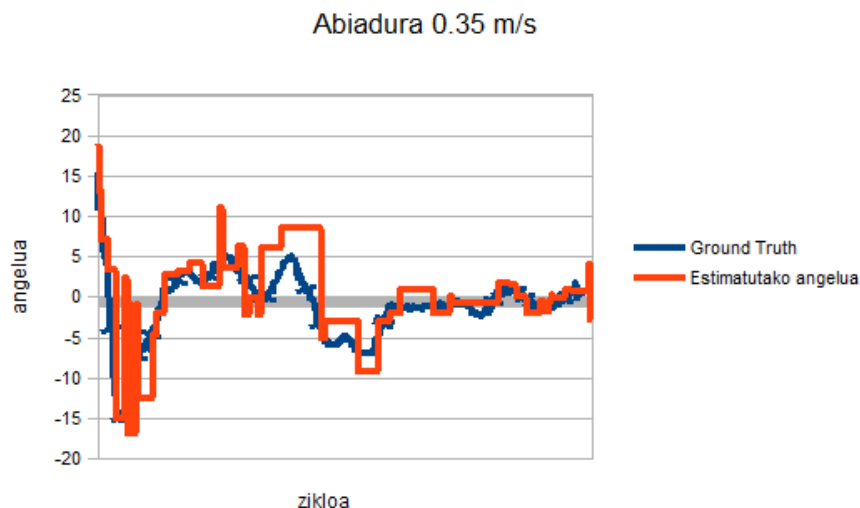
5.1. taulan, abiadura bakoitzarekin jarraipena egitean partikula-filtroaren estimazioak angeluaren kalkuluan izandako erroreen estatistikak errepresentatu dira.

Abiadura	Errore Maximoa	Errore Minimoa	Batezbesteko errorea	Desbiderapen estandarra
0.35	17.44139	0.00335	2.45981	2.72374
0.6	11.67003	0.01805	2.38145	2.30696
1.2	6.65822	0.12457	2.65172	2.02641

5.1. Taula: Abiadura ezberdinen erroreak gradutan angeluarekiko.

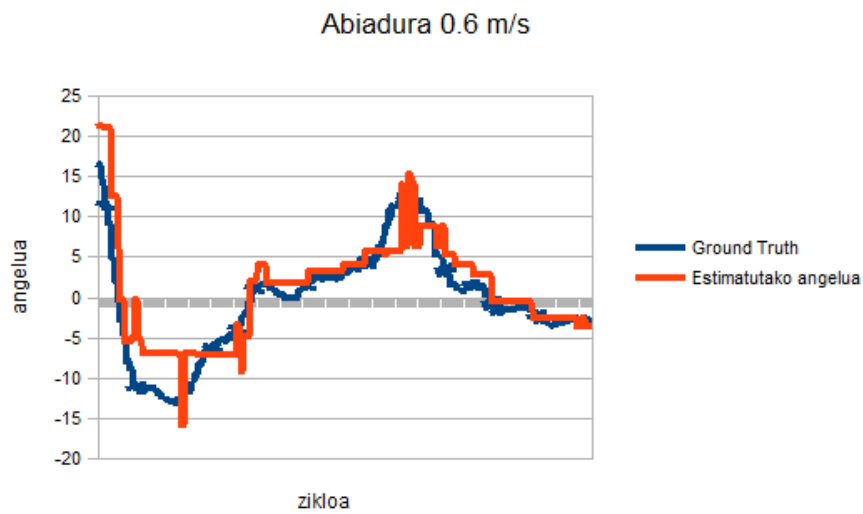
Taulan ikus daitekeen bezala, estatistiketan ez dago alde handiegirik. Errore maximoa eta minimoa ez dira oso balio esanguratsuak, maximoa partikula-filtroak estimatzean egindako huts baten ondorioa izan da, baina errore puntuala da ongorengo grafiketan antzeman daitekeen moduan. Batezbesteko errorea 0.6 m/s-ko abiaduraren kasuan da txikiena, baina balioak nahikoa antzekoak dira hiru kasuetan. Desbiderapen estandar txikiena 1.2 m/s-ko abiadurarekin eduki da.

Ondoren grafikoki errepresentatu dira errore hauek 5.8., 5.9. eta 5.10. irudietako grafikoetan. Grafika bakoitzean partikula filtroaren angeluaren estimazioa eta ground truth (benetako balioak) aren konparaketa ikus daiteke.

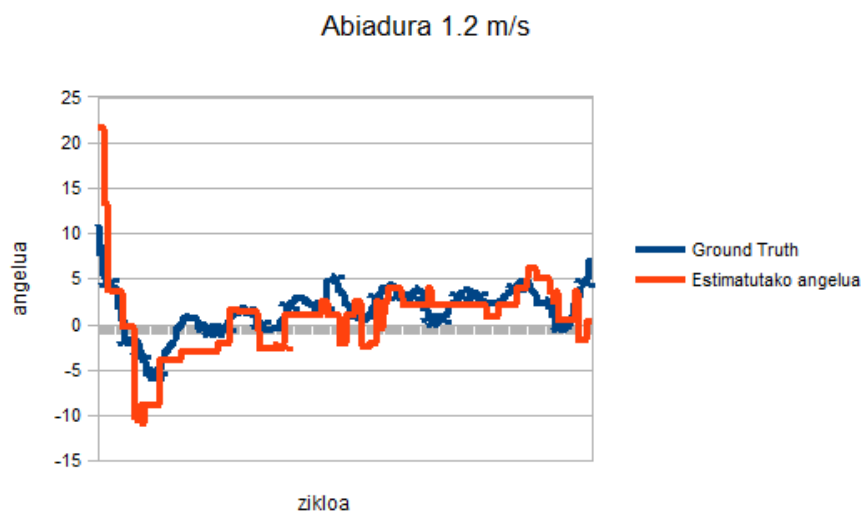


5.8. Irudia: Angeluaren konparaketa abiadura 0.35m/s denean.

Grafikoetan ikus daiteke partikula-filtroaren estimazioa nahikoa zuzena izan dela 3 kasuetarako, nahiz eta une batzuetan erroreak eduki, batez ere hasieran.



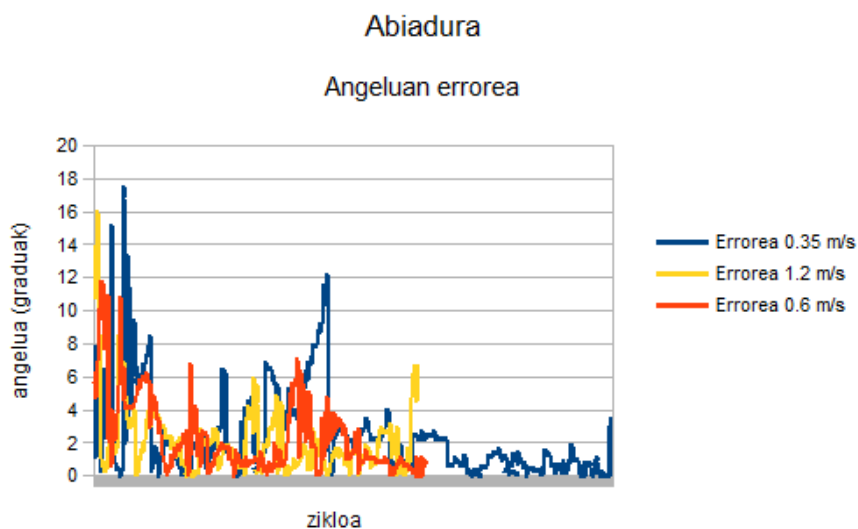
5.9. Irudia: Angeluaren konparaketa abiadura 0.6m/s denean.



5.10. Irudia: Angeluaren konparaketa abiadura 1.2m/s denean.

5.11. irudiko grafikoan hiru erroreak errepresentatu dira konparaketa bat egiteko, hirurek luzera ezberdina edukitzearen arrazoiak robotak abiadura motelenarekin zirkuitua burutzeko denbora gehiago behar izan duela da eta ondorioz ziklo kopurua handiagoa izan da.

Hiru erroreak konparatuta, ikus daiteke hiru kasuetan hasieran kosta egiten zaiola robotari estimazio zuzenak egitea, baina denborak aurrera egin ahala estimazioa zuzenagoa izan da, batez ere 0.6 m/s-ko abiaduraren kasuan. 3 kasuetan ikus dai-



5.11. Irudia: 3 abiadurek angeluaren estimazioan izandako errorearen konparaketa.

teke gorabehera handiak daudela grafikan, hori estimazioan egiten diren akatsen ondorioa da, baina estimazio asko egiten dituzenez hauen eragina ez da antzematen pertsonaren jarraipena egitean. Abiadura 0.35 m/s koa den kasuan normalagoa da angeluaren kalkuluaren errorea handiagoa izatea, robota eta pertsonaren arteko distantzia handiagoa izan baita robotaren abiadura motelaren ondorioz.

5.2. taulan, partikula-filtroak robotaren eta helburu pertsonaren arteko distantziaren estimazioan egindako erroreen estatistikak daude.

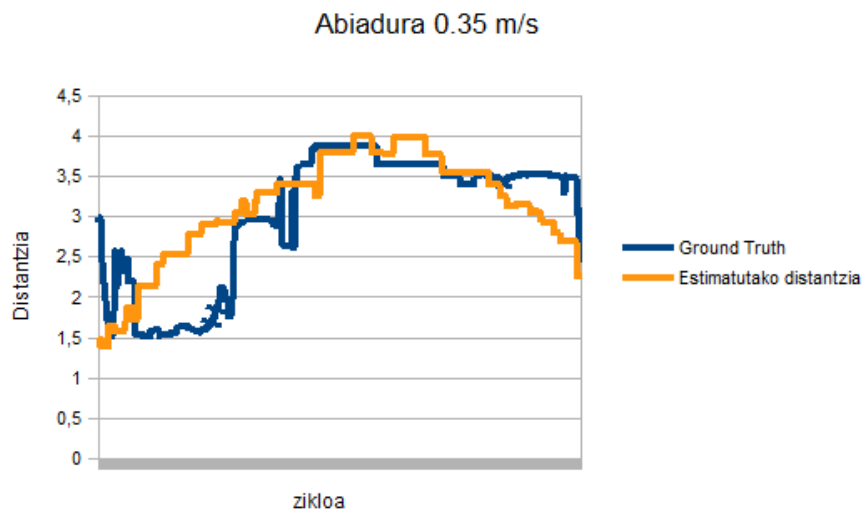
Abiadura	Errore Maximoa	Errore Minimoa	Batazbesteko errorea	Desbiderapen estandarra
0.35	1.60098	0.00601	0.43937	0.36984
0.6	1.35757	0.00249	0.34495	0.24207
1.2	0.72157	0.01376	0.23945	0.19835

5.2. Taula: Abiadura ezberdinen erroreak metroan distantziarekiko.

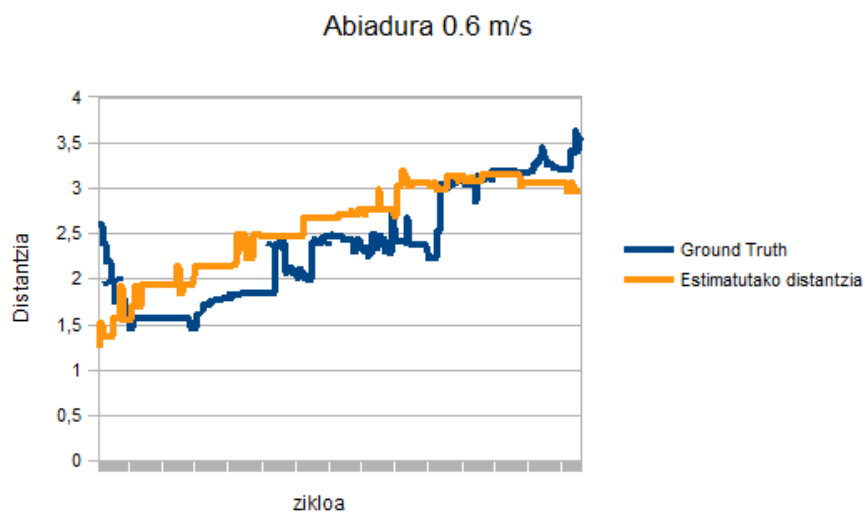
Taulan ikusi daiteke 0.35 m/s-ko kasuak dituela estatistika kaxkarrenak eta hobe-
renak berriz 1.2 m/s-ko kasuak.

Errore hauen errepresentazio grafikoa ikus daiteke jarraian, 5.12., 5.13. eta 5.14. irudietako grafikoetan. Grafiko bakoitzean, zikloetan zehar, partikula-filtroaren distantziaren estimazioa eta benetako balioa ikusi daitezke.

Grafikoak ikusita distantziaren estimazioa ez dela angeluarena bezain ona ondoriozta daiteke.



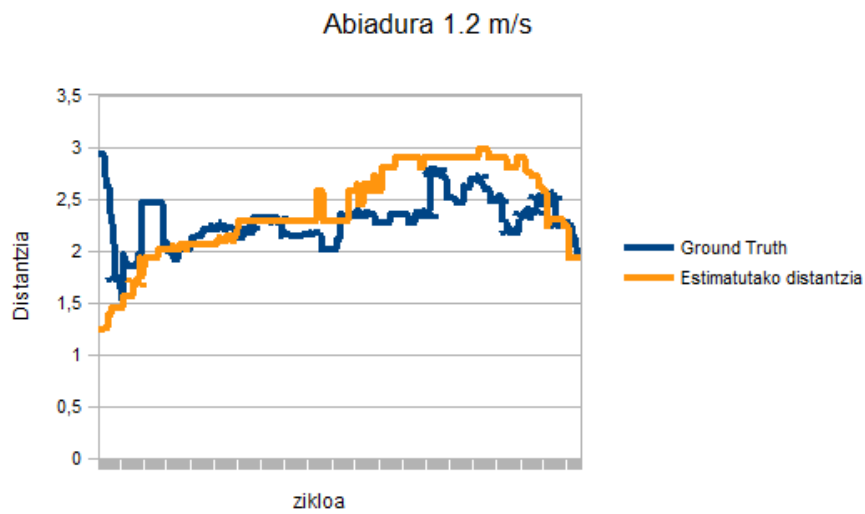
5.12. Irudia: Distantziaren konparaketa abiadura 0.35m/s denean.



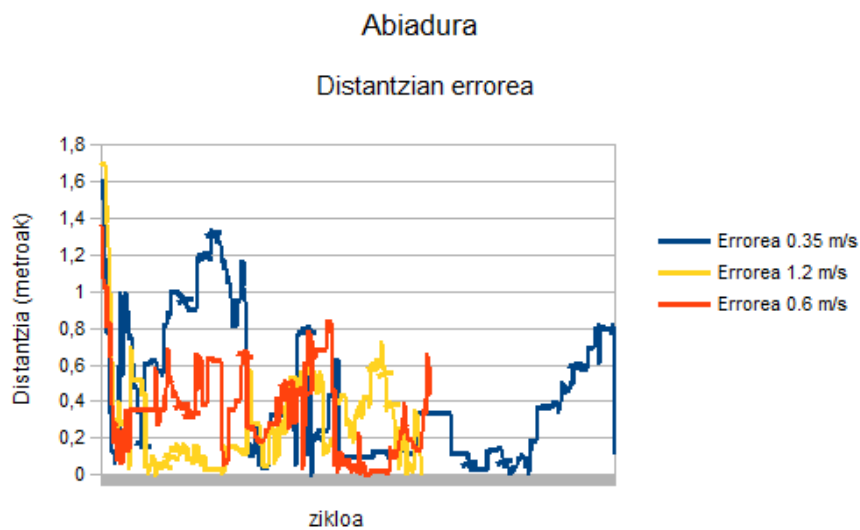
5.13. Irudia: Distantziaren konparaketa abiadura 0.6m/s denean.

5.15. irudian dagoen grafikoan, hiru erroreak errepresentatu dira konparatzeko. Ez dute luzera berdina, esan bezala, abiadura motelean eginiko proba burutzeko ziklo gehiago behar izan direlako.

Kalkulatutako 3 erroreak begiratuta, garbi ikusten da 0.35 m/s-ko abiaduraren kasuan gorabehera handiagoak izan dituela partikula-filtroak estimazioak egi-teko orduan. Hori robota eta gizakiaren arteko distantzia beste kasuetan baino handiagoa izan delako gertatu daiteke.



5.14. Irudia: Distantziaren konparaketa abiadura 1.2m/s denean.



5.15. Irudia: 3 abiadurek distantziaren estimazioan izandako errorearen konparaketa.

Ebaluazio kualitatiboa

Esan behar da 3 kasuetan robota zirkuitua burutzeko gai izan dela, baina abiadura 0.35 m/s koa izan denean robota oso motel joan da eta oso atzera geratu da pertsonarekiko une batzuetan, emaitza estatistikoetan alde handiegirik egon ez arren abiadu hau txikiegia da jarraipena egiteko.

Beste bi kasuak alderatuta esan daiteke jarraipenean ez dela alde haundiegirik izan, abiadura hori maximoa da eta jarraitzen ari den pertsona ere azkarrago joan behar da abiadura hori aprobetxatzeko, hau abiadura 1.2 m/s-koa den kasuan gertatzen da.

Proben ondorioak

Proba hauetatik ondorioztatu da, 0.35 oso abiadura txikia dela robotarentzat. 1.2-ko abiadura ere ez da gehiegi aprobetxatzen, abiadura hori maximoa baita eta horretarako pertsonak ere azkarrago joan behar du. Probetan ikusi da estimazioak errore batzuk izaten dituela eta horien ondorio dira grafikoetako balio maximoak. Grafikoei begiratuta orokorrean angeluen estimazioa hobea izan dela esan daiteke, distantziaren kasuan eragin handiagoa eduki du abiadurak.

Ebaluazio kuantitatiboa eta kualitatiboa kontuan edukita robotarentzat 0.6 m/s ko abiadura ipintzea egokia dela ondorioztatu da proba hauetatik.

5.3.5. Partikula-Filtroaren partikula kopuru ezberdinekin probak

Proba honetan partikula-filtroak dituen partikula kopuruak bere estimazioan duen eragina aztertu nahi izan da. Teoriak dioen bezala zenbat eta partikula gehiago eduki partikula-filtroak, estimazioa zuzenagoa izango da. Partikula gehiago erabiltzeak ordea kostu konputazional handiagoa dauka eta horrek robotean bestea arazo batzuk sor ditzake.

Proba egiteko partikula kopuru ezberdin hauek erabili dira:

- 25
- 50
- 100

Bestalde, probak egiteko 0.6 m/s-ko abiadura erabili du robotak eta sentsoreetan ipini diren balioak hauek izan dira: hanken detekzioa 0.2, txalekoaren detekzioa 0.6 eta detekzio termikoa 0.2.

Ebaluazio kuantitatiboa

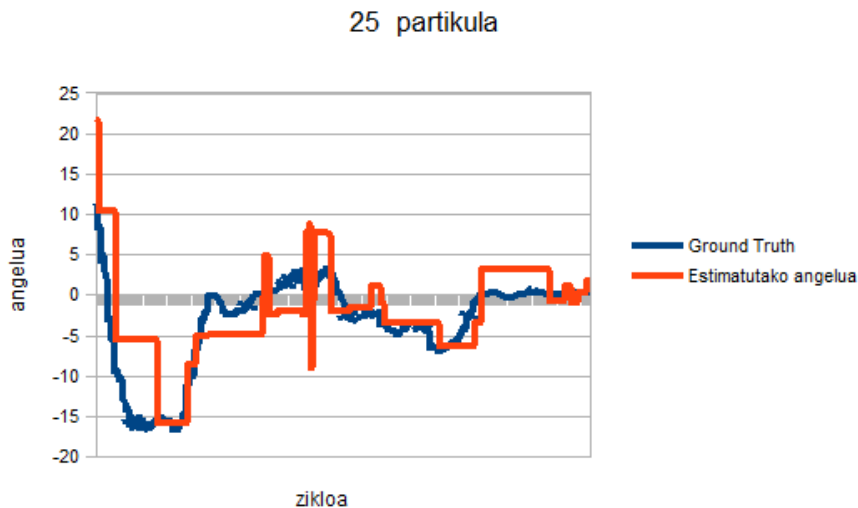
5.3. taulan, partikula kopururu ezberdinekin jarraipena egitean partikula-filtroaren estimazioak angeluaren kalkuluan izandako erroreen estatistikak errepresentatu dira.

Partikula kopurua	Errore Maximoa	Errore Minimoa	Batazbesteko errorea	Desbiderapen estandarra
25	19.82753	0.00063	3.41207	3.19208
50	6.96662	0.00673	2.07927	1.46956
100	2.50161	0.18708	1.07507	0.79224

5.3. Taula: Partikula kopuru ezberdinekin lortutako erroreak gradutan angeluarekiko.

Aurreko taulan ikus daiteke partikula-filtroaren partikula kopuruak eragin zuzena duela estatistiketan zenbat eta partikula gehiago emaitza hobekiago lortu dira.

Ondoren grafikoki errepresentatu dira erroreak hauek 5.16, 5.17. eta 5.18. irudietako grafikoetan. Grafika bakoitzean, partikula-filtroak egindako angeluaren estimazioaren eta ground truth (benetako balioak)-aren bilakaera ikus daiteke zikloek aurrera egin ahala.

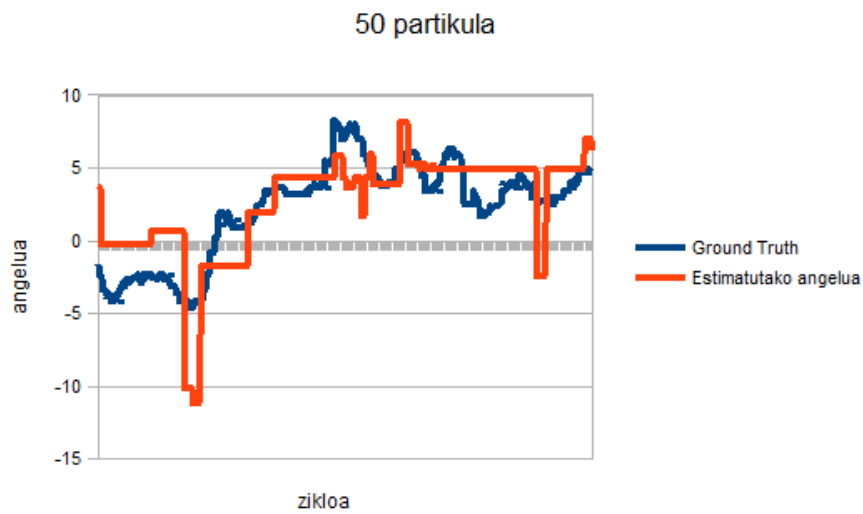


5.16. Irudia: Angeluaren konparaketa partikula kopurua 25 denean.

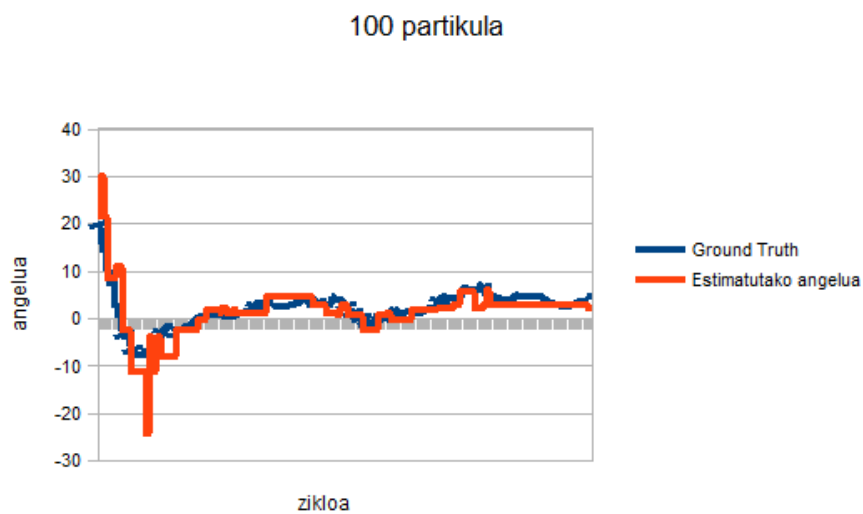
Grafikoei begiratuta garbi ikus daiteke partikula kopurua 100 den kasuak estimazio hoberena egin duela. Gainera, ikus daiteke helburu pertsona parez pare eraman duela angelua askotan 0 inguruan egon baita. Beste bi kasuetan angelua 0tik aldendu egin da. Hori estimazioan izandako erroreen ondorioa da.

5.19. irudiko grafikoan, hiru erroreak errepresentatu dira konparaketa egiteko.

Grafiko honetan ikusten da estimazioan errore handienak hasieran gertatu direla. Robota gelditu dagoenez martxan jarri ondoren estabilitatea lortu arte balantza txiki batzuk egiten ditu eta horrek estimazioan eragina du, baina ez da antzematen jarraipena egiteko unean.



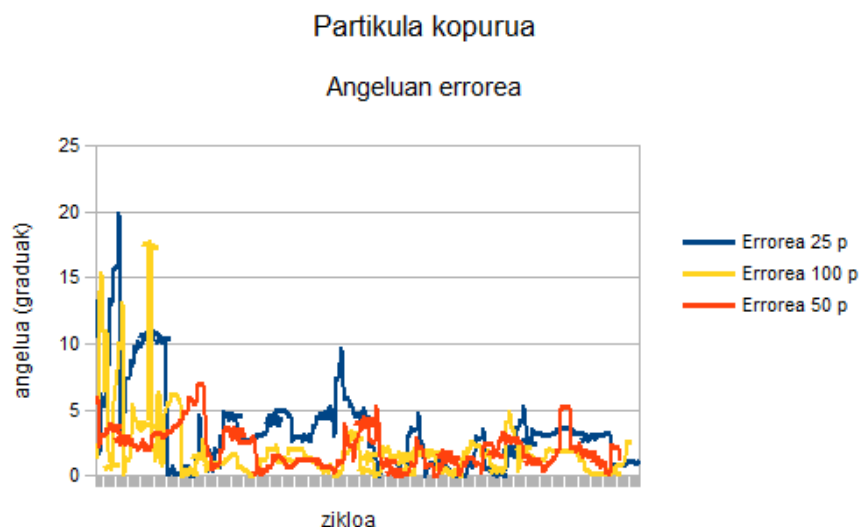
5.17. Irudia: Angeluaren konparaketa partikula kopurua 50 denean.



5.18. Irudia: Angeluaren konparaketa partikula kopurua 100 denean.

5.4. taulan, partikula-filtroak robotaren eta helburu pertsonaren arteko distantziaren estimazioan egindako erroreen estatistikak daude.

Taulan ikus daiteke angeluaren kasuan baino ezberdintasun txikiagoak egon direla hiru kasuetan distantziaren estimazioa egitean baina hala eta guztiz ere, partikula kopuru handiagoarekin emaitza hobekiago lortu dira.



5.19. Irudia: Partikula kopuru ezberdinek angeluaren estimazioan izandako errorearen konparaketa.

Partikula kopurua	Errore Maximoa	Errore Minimoa	Batazbesteko errorea	Desbiderapen estandarra
25	1.38132	0.00000	0.20899	0.15281
50	0.48518	0.00756	0.15775	0.11651
100	0.23461	0.00119	0.10572	0.08572

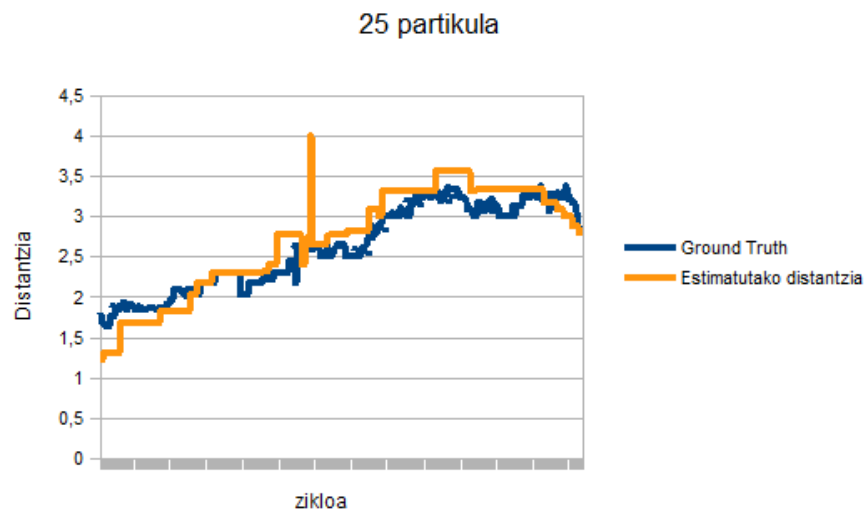
5.4. Taula: Partikula kopuru ezberdinen izandako erroreak metrotan distantziarekiko.

Errore hauen errepresentazio grafikoa ikus daiteke jarraian, [5.20.](#), [5.21.](#) eta [5.22.](#) irudietako grafikoetan. Grafiko bakoitzean, zikloetan zehar, partikula-filtroak egindako distantziaren estimazioa eta benetako balioaren arteko ezberdintasuna errepresentatzen da.

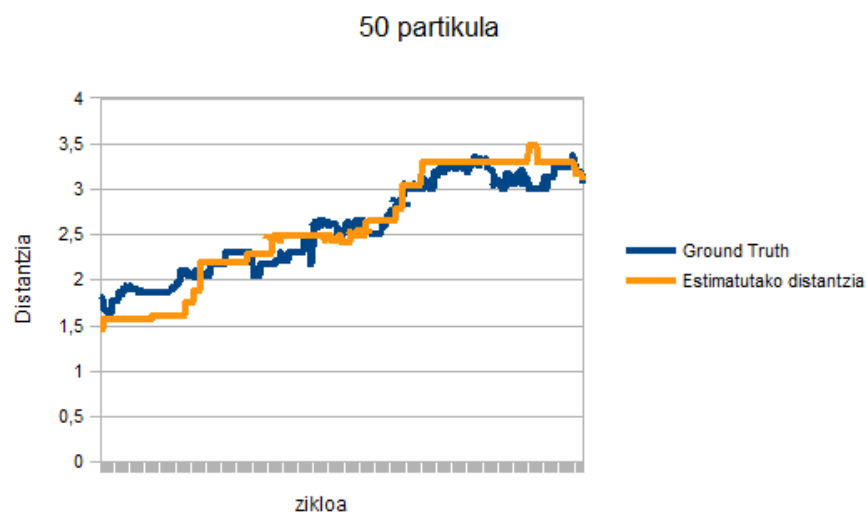
Grafikoetan ikus daiteke estimazio nahiko zuzena egin dela hiru kasuetan, errorea handiagoa da partikula kopuru txikiagoarekin, teoriak dioen bezala.

[5.23.](#) irudiko grafikoan, hiru erroreak errepresentatu dira konparatzeko.

Grafiko horretan ikus daiteke, hiru kasuetan estimazioa nahikoa zuzena izan dela, baina partikula kantitate handiagoa edukita estimazio erroreak gertatzeko arriskua txikitu egiten da. Partikula kopurua 25 denean estimazio errore handi bat gertatzen dela ikusten da.



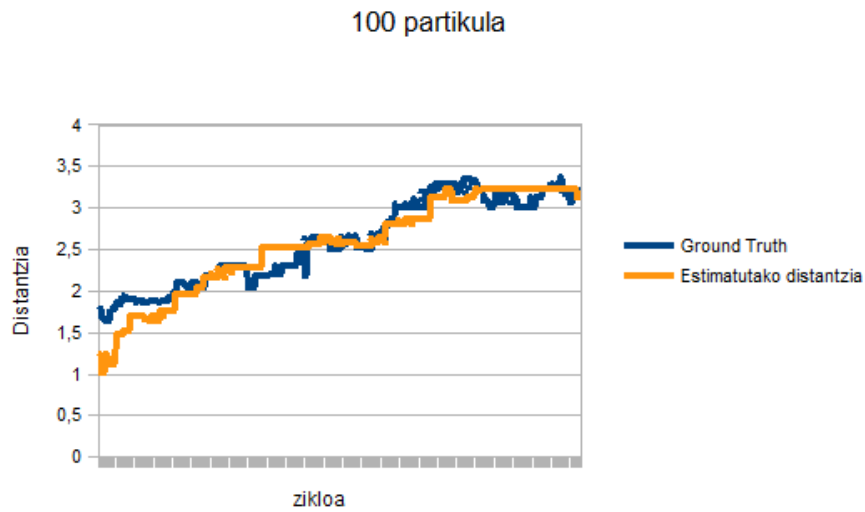
5.20. Irudia: Distantziaren konparaketa partikula kopurua 25 denean.



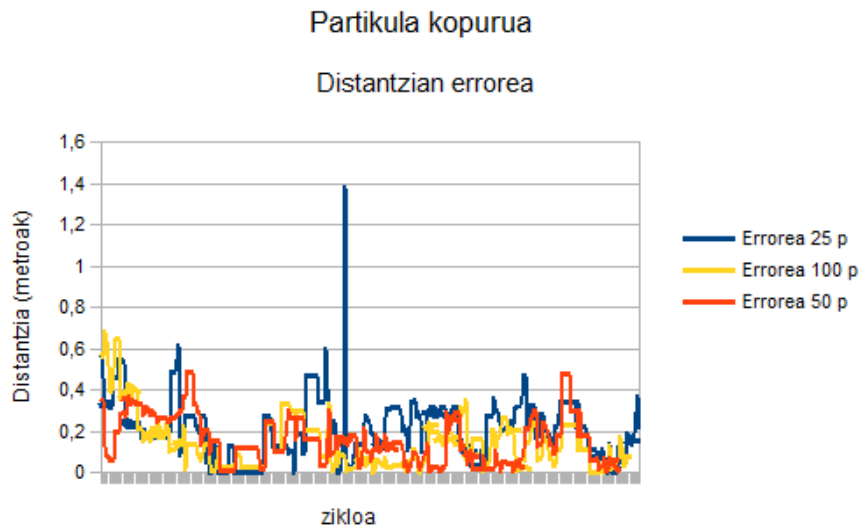
5.21. Irudia: Distantziaren konparaketa partikula kopurua 50 denean.

Ebaluazio kualitatiboa

Robotaren portaera ikusita esan behar da ez dela alde handiegirik antzeman partikula kopuru ezberdinekin egindako probetan. Proba hauek zirkuitu zuzen batean burutu dira eta hiru kasuetan robotak ibilbidea zuzen burutzeko gai izan da. Antzeman da robotak partikula kopuru gehiago zituenean zuzenago joan dela eta duda gutxiago eduki duela hartu beharreko norabidea erabakitzeke unean.



5.22. Irudia: Distantziaren konparaketa partikula kopurua 100 denean.



5.23. Irudia: Partikula kopuru ezberdinek distantziaren estimazioan izandako errorearen konparaketa.

Proben ondorioak

Probetatik ondorio garbiak atera daitezke, partikula-filtroan zenbat eta partikula gehiago ipini emaitza hobea lortuko dela. Baina kontuz horrekin, zenbat eta partikula gehiago ipinita kostu konputazionala asko igotzen da eta honek robotaren errendimenduan eragin dezake estimazio kaxkarragoak egitera bultzatuz.

Ondorioz oreka bat bilatu behar da partikula kopurua eta robotaren errendimenduaren artean, kasu honetan ikusi da 100 partikularekin robotaren errendimendua egokia izan dela eta ez duela eraginik izan robotaren portaeran.

5.3.6. Sentsoreetan pisu ezberdinekin probak

Proba honetan robotak dituen sentsore bakoitzaren eraginkortasuna aztertu nahi izan da. Jakina da sentsore bakoitzak ez duela fidagarritasun berdina pertsonen detekzioan, eskaintzen dituzten datuak ezberdinak baitira eta ondorioz ez da informazio berdina lortzen. Proba hauetan sentsore bakoitzak bakarrik lan eginez duen ahalmena eta hiru sentsoreak fusionatuta lortzen dena konparatu dira. Horretarako detekzio ezberdinei pisu ezberdinak eman zaizkie eta proba ezberdinak egin dira pertsonen jarraipena egiteko konfigurazio egokienaren bila.

Hauek dira probak egiteko aukeratu diren kasuak:

- Txalekoaren detekzioa=1, detekzio termikoa=0, hanken detekzioa=0.
- Txalekoaren detekzioa=0, detekzio termikoa=1, hanken detekzioa=0.
- Txalekoaren detekzioa=0, detekzio termikoa=0, hanken detekzioa=1.
- Txalekoaren detekzioa=0.6, detekzio termikoa=0.2, hanken detekzioa=0.2.
- Txalekoaren detekzioa=0.2, detekzio termikoa=0.6, hanken detekzioa=0.2.
- Txalekoaren detekzioa=0.2, detekzio termikoa=0.2, hanken detekzioa=0.6.
- Txalekoaren detekzioa=0.3, detekzio termikoa=0.3, hanken detekzioa=0.3.

Probetan robotak izan duen abiadura lineala 0.6 m/s-koa izan da, eta partikula kopurua 50.

Ebaluazio kuantitatiboa

5.5. taulan, sentsoreei balio ezberdinak emanda jarraipena egitean partikula-filtroaren estimazioak angeluaren kalkuluan izandako errorearen estatistikak errepresentatu dira. Baliorik ez dagoen kasuetan, robotak ez da gai izan pertsonen jarraipena egiteko detekzioei balio horiek ipinita.

Aurreko taulan ikus daiteke txalekoaren detekzioa izan dela gainontzeko detekzioen laguntzarik gabe pertsonaren jarraipena egiteko gai izan den bakarra.

Sentsorearen pixua	Errore Maximoa	Errore Minimoa	Batazbesteko errorea	Desbiderapen estandarra
Txalekoa=1 Termopila=0 Hanka=0	16.38087	0.00232	3.13559	3.38697
Txalekoa=0 Termopila=1 Hanka=0	--	--	--	--
Txalekoa=0 Termopila=0 Hanka=1	--	--	--	--
Txalekoa=0.6 Termopila=0.2 Hanka=0.2	6.96629	0.00674	2.07927	1.46956
Txalekoa=0.2 Termopila=0.6 Hanka=0.2	21.21907	0.05244	2.83999	2.95958
Txalekoa=0.2 Termopila=0.2 Hanka=0.6	10.97178	0.18505	2.94023	3.05760
Txalekoa=0.3 Termopila=0.3 Hanka=0.3	12.37327	0.05373	2.10003	2.00221

5.5. Taula: Detekzioei balio ezberdinak emanda lortutako erroreak gradutan angeluarekiko.

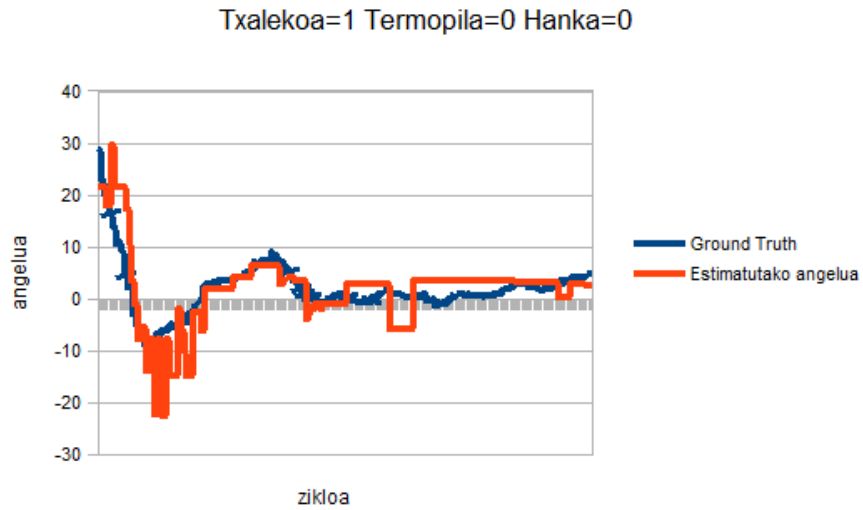
Detekzio termikoa eta hanken detekzioa, beste informaziorik gabe, ez dira jarraipena egiteko gai izan. Taulako emaitzak aztertuta ikus daiteke txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasuan, eta txalekoa=0.3, termopila=0.3 eta hanka=0.3 kasuan lortu direla emaitzarik onenak. Laser edo termopilari balio handiak emateak angeluaren estimazioa kaxkarragoa izatea eragin dute.

Ondoren, grafikoki errepresentatu dira errore hauek [5.24.](#), [5.25.](#), [5.26.](#), [5.27.](#) eta [5.28.](#) irudietako grafikoetan. Grafika bakoitzean partikula-filtroak egindako angeluaren estimazioaren eta ground truth (benetako balioak)-aren bilakaera ikus daiteke zikloek aurrera egin ahala. Jarraipena egiteko gai izan ez diren kasuen grafikorik ez dago.

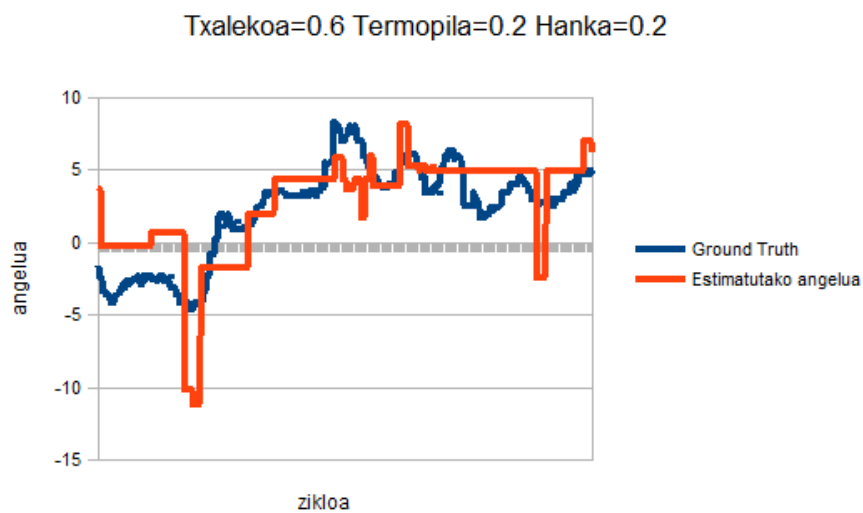
Grafiko hauei begiratuta hanka=0.6 balioa duen kasuan izandako emaitza txarrak ikusi daitezke, ingurunean zeuden gauza ezberdinek eta kristal batzuek sortutako zaratak hanken detekzio faltsu asko eragin ditu eta grafiko horretan errepresentatu da hori.

[5.29.](#) irudiko grafikoan, kasu bakoitzean lortutako erroreak errepresentatu dira konparaketa bat egiteko.

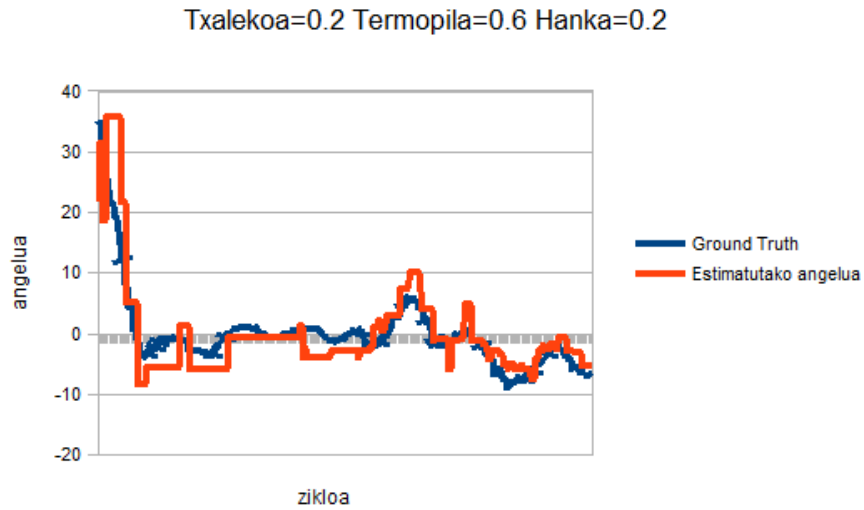
Grafiko horretan ikus daiteke kasu batetik bestera alde haundiak egon direla, eta bertatik detekzio bakoitzaren fidagarritasuna oso ezberdina dela ondoriozta daiteke.



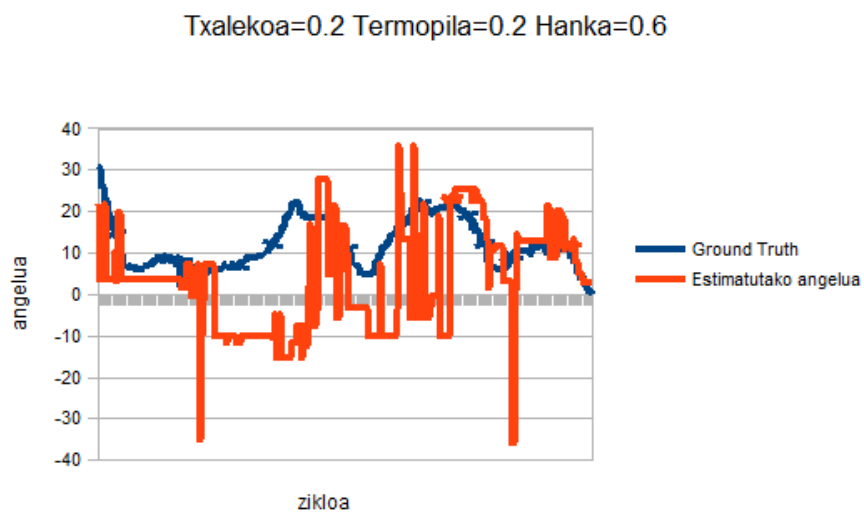
5.24. Irudia: Angeluaren konparaketa Txalekoa=1, Termopila=0 eta Hanka=0 denean.



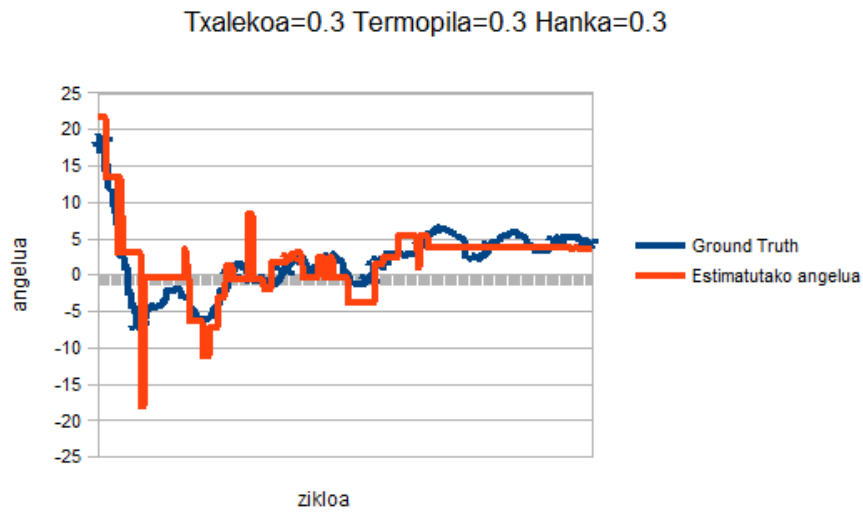
5.25. Irudia: Angeluaren konparaketa Txalekoa=0.6, Termopila=0.2 eta Hanka=0.2 denean.



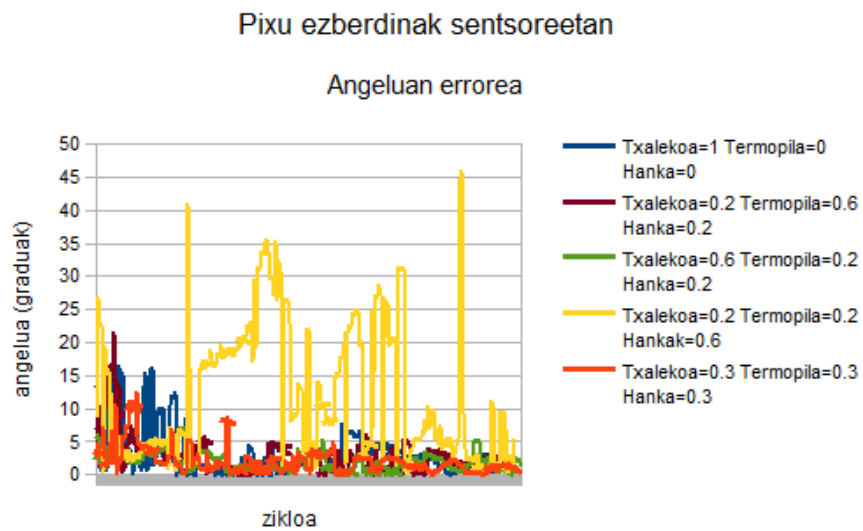
5.26. Irudia: Angeluaren konparaketa Txalekoa=0.2, Termopila=0.6 eta Hanka=0.2 denean.



5.27. Irudia: Angeluaren konparaketa Txalekoa=0.2, Termopila=0.2 eta Hanka=0.6 denean.



5.28. Irudia: Angeluaren konparaketa Txalekoa=0.3, Termopila=0.3 eta Hanka=0.3 denean.



5.29. Irudia: Partikula kopuru ezberdinek angeluaren estimazioan izandako errorearen konparaketa.

5.6. taulan, kasu bakoitzean partikula-filtroak robotaren eta helburu pertsonaren arteko distantziaren estimazioan egindako erroreen estatistikak daude. Estatistikak ez dauden kasuak ez dute jarraipena egitea lortu.

Sentsorearen pixua	Errore Maximoa	Errore Minimoa	Batazbesteko errorea	Desbiderapen estandarra
Txalekoa=1 Termopila=0 Hanka=0	3.03544	0.01495	0.41913	0.36479
Txalekoa=0 Termopila=1 Hanka=0	--	--	--	--
Txalekoa=0 Termopila=0 Hanka=1	--	--	--	--
Txalekoa=0.6 Termopila=0.2 Hanka=0.2	0.48518	0.00757	0.15775	0.11651
Txalekoa=0.2 Termopila=0.6 Hanka=0.2	2.34336	0.18140	0.68922	0.33362
Txalekoa=0.2 Termopila=0.2 Hanka=0.6	2.15026	0.85899	1.16029	0.35461
Txalekoa=0.3 Termopila=0.3 Hanka=0.3	0.63938	0.00218	0.29091	0.15967

5.6. Taula: Sentsoreei balio ezberdinak emanez lortutako erroreak metrotan distantziarekiko.

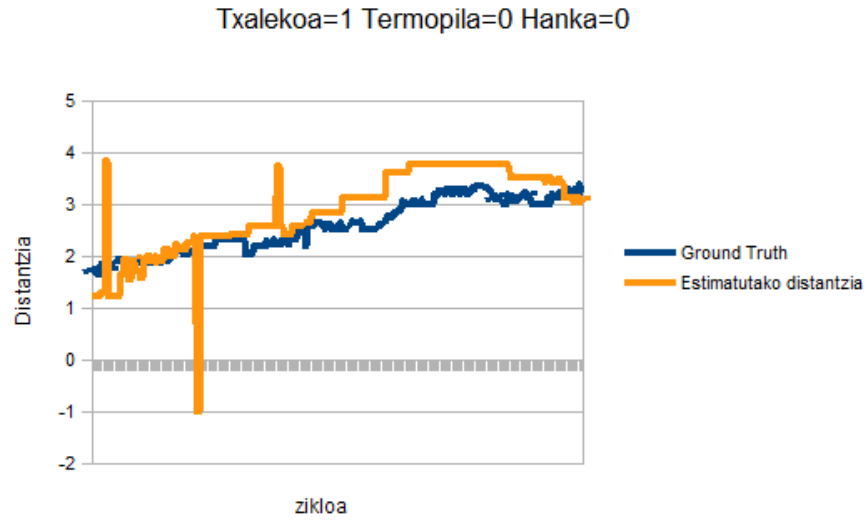
Distantziaren estimazioan ere angeluarenean lortutako emaitza antzekoak lortu dira kasu hoberenei dagokienez.

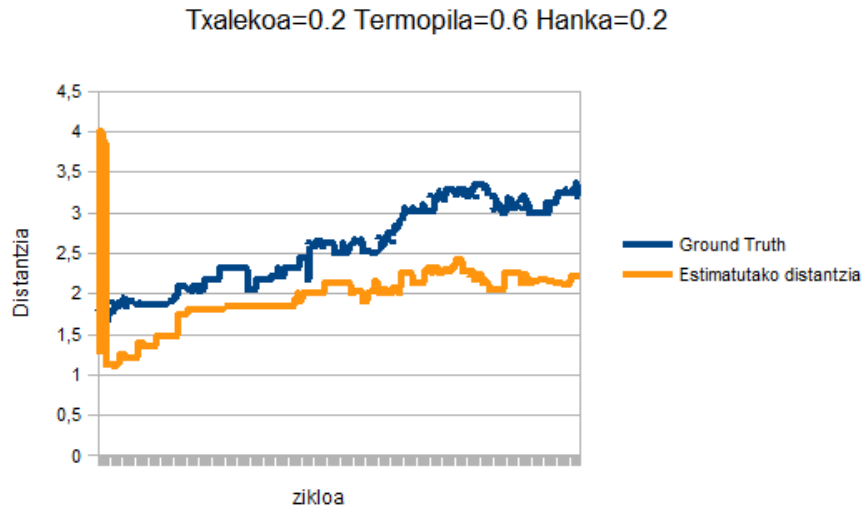
Errore hauen errepresentazio grafikoa ikus daiteke jarraian, 5.30., 5.31., 5.32., 5.33. eta 5.34. irudietako grafikoetan. Grafiko bakoitzean partikula-filtroak egindako distantziaren estimazioaren eta benetako balioa konparatu dira. Jarraipena egin ez duten kasuetarako ez dago grafikorik.

Grafiko hauetan ikus daiteke termopilak balio altua edukitzeak eragina izan duela, termopila=0.6 den kasuak ez du angeluaren estimazioa bezain emaitza onik lortu eta detekzio guztiek 0.3 duten kasuan ere gauza bera gertatu da. Bestalde, garbi ikusten dena da hanken detekzioak balio altua duenean detekzio faltsu asko gertatzen direla.

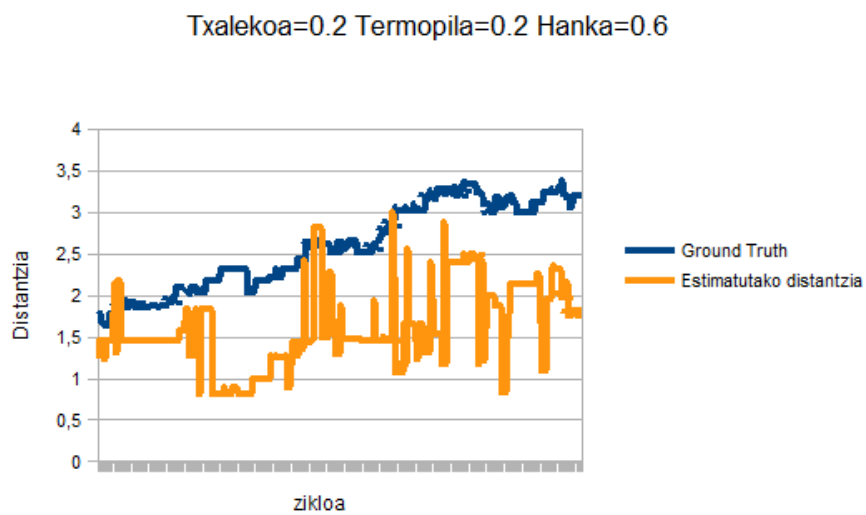
5.35. irudian dagoen grafikoan, lortutako erroreak konparatu dira.

Grafiko horretan ere, aurreko grafikoetan bezala, kasuen arteko ezberdintasun handiak daudela ikus daiteke. Sentsoreei pisu ezberdinak emandako kasu bakoitzean robotaren partikula-filtroak angelu zein distantziaren estimazioan egiten dituen erroreak oso ezberdinak izan dira.

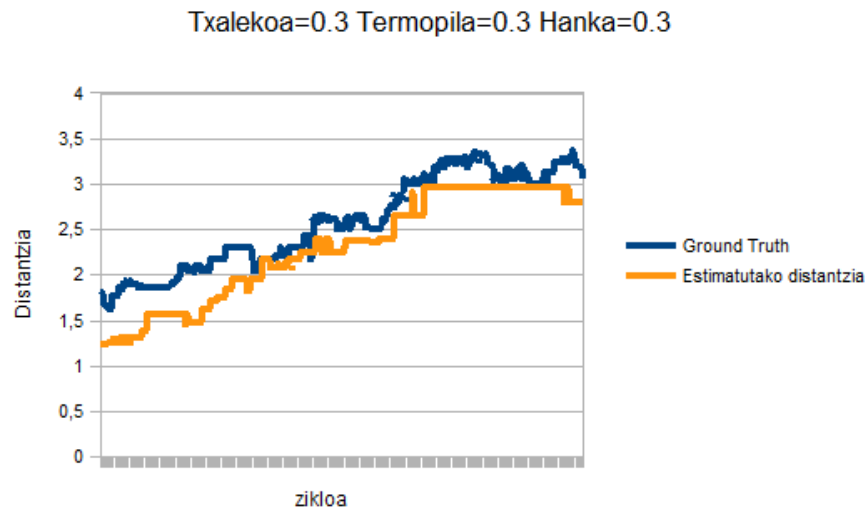




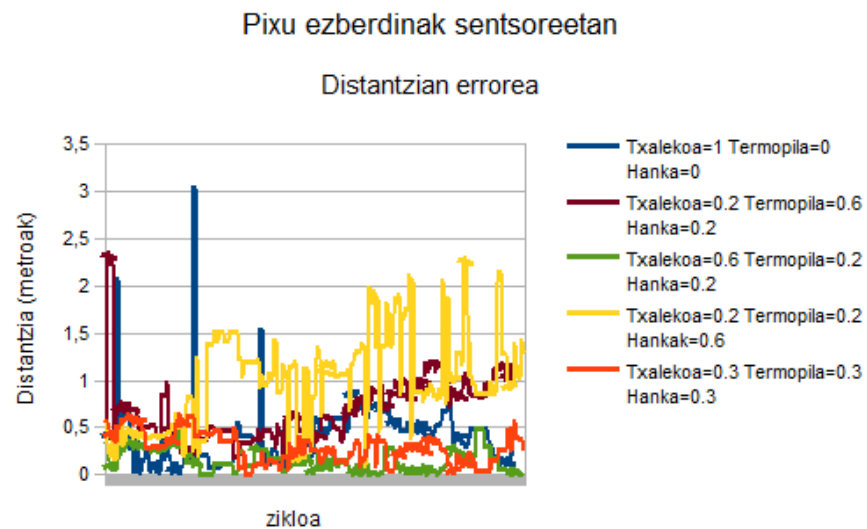
5.32. Irudia: Distantziaren konparaketa Txalekoa=0.2, Termopila=0.6 eta Hanka=0.2 denean.



5.33. Irudia: Distantziaren konparaketa Txalekoa=0.2, Termopila=0.2 eta Hanka=0.6 denean.



5.34. Irudia: Distantziaren konparaketa Txalekoa=0.3, Termopila=0.3 eta Hanka=0.3 denean.



5.35. Irudia: Partikula kopuru ezberdinek distantziaren estimazioan izandako errorearen konparaketa.

Ebaluazio kualitatiboa

Esan behar da kasu batetik bestera alde haundia egon dela. Lehenik eta behin, txalekoa=0, termopila=1 eta hanka=0 kasuek eta txalekoa=0, termopila=0 eta hanka=1 kasuek ez dute jarraipenik egin. Txalekoa=0.2, termopila=0.2 eta hanka=0.6 kasuan jarraipena oso txarra izan da, jarraitzen ari zen pertsona une batzuetan geratu egin behar izan da robotak beste zerbait detektatu duelako. Beste lau kasuetan jarraipena egiteko gai izan da, sensazio onenak ordea txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasuan eta txalekoa=0.3, termopila=0,3 eta hanka=0.3 kasuan eduki dira, robotak beste kasuetan baino bide zuzenagoa egin du eta jarraipena segurua izan da.

Proben ondorioak

Probetatik ateratzen den ondorio nagusia sentsoreen arteko ezberdintasuna da, eta errendimendua handitzeko fusio egoki bat egitea beharrezkoa dela. Probetan garbi ikusi da txalekoaren detekzioa dela fidagarriena. Sentsore termikoak matrize txikia erabiltzen du eta horrek ez ditu datuetan aldaketa haundiegiak eragiten, horregatik, nahiz eta balio handia ipini sentsore horri jarraipena ondo egiten du robotak. Hanken sentsoreak arazo gehiago ditu, ingurunearen arabera asko aldatzen da errendimendua. Probak egin ditudan gunean kristalezko pareta bat dago eta horrek asko nahastu du laserra, emaitzetan ikusi bezala detekzio faltsu asko egin ditu.

Hala eta guztiz ere, emaitzek argi erakusten dute detekzio ezberdinak fusionatuta emaitza hobekak lortzen direla. Emaitza hoberenak lortu diren bi kasuak hauek izan dira: txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasua eta txalekoa=0.3, termopila=0.3 eta hanka=0.3 kasua.

5.3.7. Zirkuitu borobil batean probak

Proba hau egitego aurreko probetatik eskuratutako emaitzak kontuan eduki dira eta proba horietatik ateratako ondorioekin beste bi proba egin dira zirkuitu borobil batetan robotaren portaera egokiena aurkitu asmoz.

Abiadurarekin egindako probetatik ondorioztatu da ez dagoela alde haundirik 0.6 m/s eta 1.2 m/s ko abiaduren artean partikula-filtroaren estimazioa begiraturaz. Baina robotak biraketa azkarrak egiteko arazoa duela ikusirik, proba hauetarako 0.6 m/s-ko abiadura erabiltzea erabaki da robotak abiadura aldaketa handiegirik ez egiteko.

Partikula-filtroari dagokionez, probetan ikusi da 100 partikula ipinita estimazio hobea egiten dela eta konputazionalki ez dela zama handiegia izan robotarentzat, beraz, proba hauek egiteko 100 partikula ipini dira partikula-filtroan.

Sentsoreen konbinaketa aztertu denean, aurreko atalean 5.3.6 proba multzoan, ondorioztatu da 2 konbinazioekin lortzen zela emaitza hobereena, horregatik bi konbinazio horiek erabili dira proba hauetarako:

- Txalekoaren detekzioa=0.6, detekzio termikoa=0.2, hanken detekzioa=0.2.
- Txalekoaren detekzioa=0.3, detekzio termikoa=0.3, hanken detekzioa=0.3.

Ebaluazio kuantitatiboa

5.7. taulan, proba hauetan partikula-filtroaren estimazioak angeluaren kalkuluan izandako erroreen estatistikak errepresentatu dira.

Sentsorearen pixua	Errore Maximoa	Errore Minimoa	Batazbesteko errorea	Desbiderapen estandarra
Txalekoa=0.6 Termopila=0.2 Hanka=0.2	15.65656	0.01645	3.45044	2.86670
Txalekoa=0.3 Termopila=0.3 Hanka=0.3	37.32399	0.00005	3.45219	3.85097

5.7. Taula: Zirkuitu borobilean lortutako erroreak gradutan angeluarekiko.

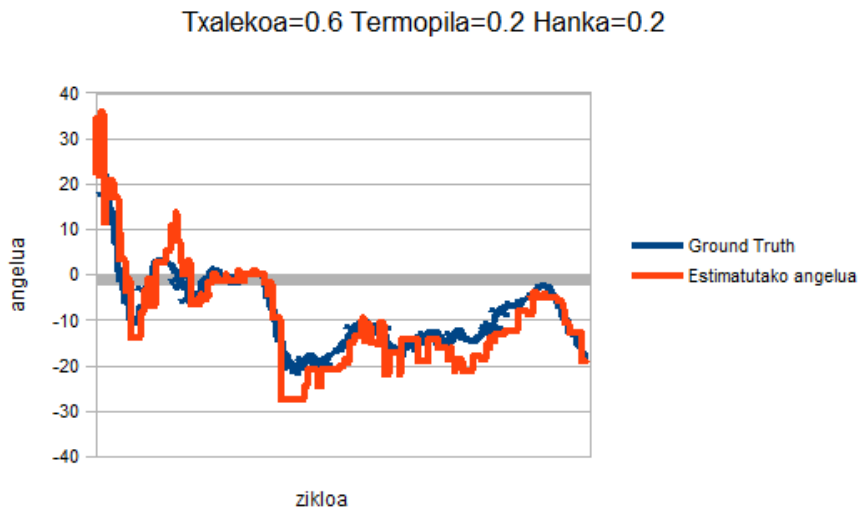
Taulei begiratuta bi kasuen arteko ezberdintasun handiena balio maximoan eta desbiderapen estandarrean dagoela ikusten da.

Ondoren, grafikoki errepresentatu dira errore hauek 5.36. eta 5.37. irudietako grafikoetan.

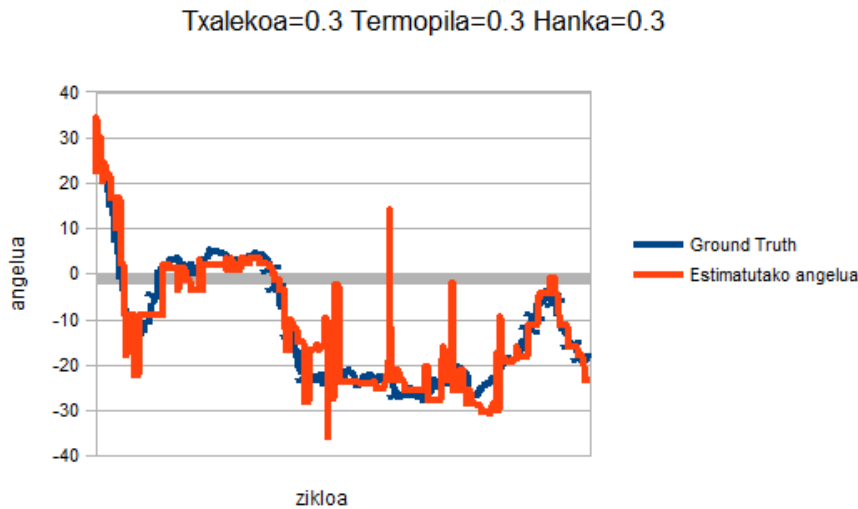
Grafika hauetan ikus daiteke batez ere txalekoa=0.3, termopila=0.3 eta hanka=0.3 kasuan partikula-filtroak estimazio errore batzuk izan dituela estimatutako balioa benetako baliotik urrundu dutena, baina, orokorrean, estimazioak zuzenak izan dira.

5.38. irudian dagoen grafikoan, kasu bakoitzean lortutako erroreak errepresentatu dira konparaketa egiteko.

Grafikoan ikus daiteke txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasuak errore gutxiago izan dituela, orokorrean angeluaren estimazioan izandako errorea antzekoa izan da baina errore puntual handiak gertatu dira txalekoa=0.3, termopila=0.3 eta hanka=0.3 kasuan.



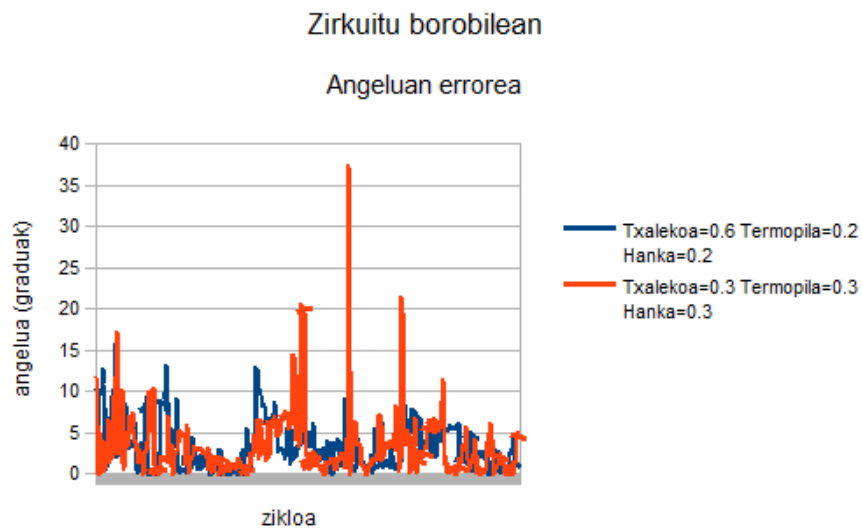
5.36. Irudia: Zirkuitu borobilean kalkulaturako angeluaren eta benetako balioen arteko erlazioa, txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasuan.



5.37. Irudia: Zirkuitu borobilean kalkulaturako angeluaren eta benetako balioen arteko erlazioa, txalekoa=0.3, termopila=0.3 eta hanka=0.3 kasuan.

5.8. taulan, kasu bakoitzean partikula-filtroak robotaren eta helburu pertsonaren arteko distantziaren estimazioan egindako errorearen estatistikak daude.

Taulan ikus daitekeen bezala distantziaren estimazioan ere bi kasuetan nahiko balio antzekoak lortu dira, baina angeluen kalkuluan gertatu ez bezala txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasuan errorearen batez bestekoa txikiagoa izan da.



5.38. Irudia: Zirkuitu borobilean angeluaren estimazioan izandako erroreak konparaketa.

Sensorearen pixua	Errore Maximoa	Errore Minimoa	Batazbesteko errorea	Desbiderapen estandarra
Txalekoa=0.6 Termopila=0.2 Hanka=0.2	2.78422	0.00000	0.15321	0.27759
Txalekoa=0.3 Termopila=0.3 Hanka=0.3	2.39358	0.00176	0.20037	0.31622

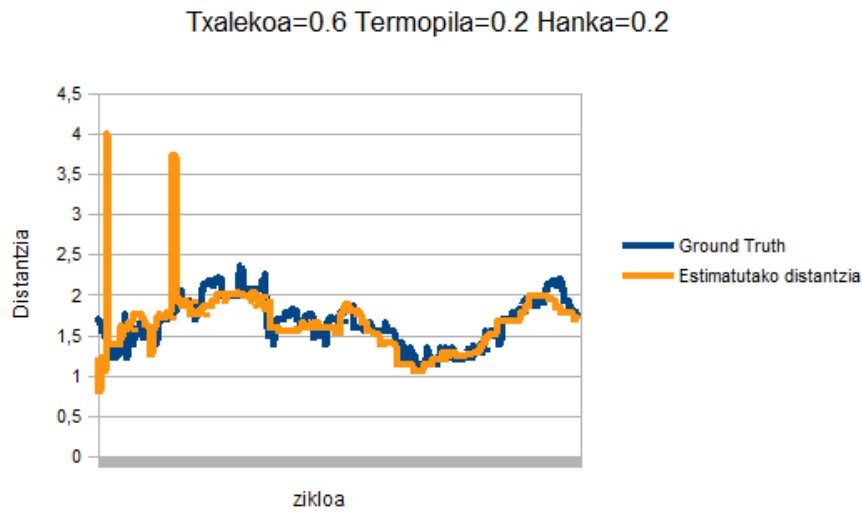
5.8. Taula: Zirkuitu borobilean lortutako erroreak metroan distantziarekiko.

Errore hauen errepresentazio grafikoa ikus daiteke jarraian dauden bi grafikoetan, [5.39.](#) eta [5.40.](#) irudietan. Grafiko bakoitzean, zikloetan zehar, partikula-filtroak egindako estimazioa eta benetako balioaren arteko ezberdintasuna errepresentatzen da.

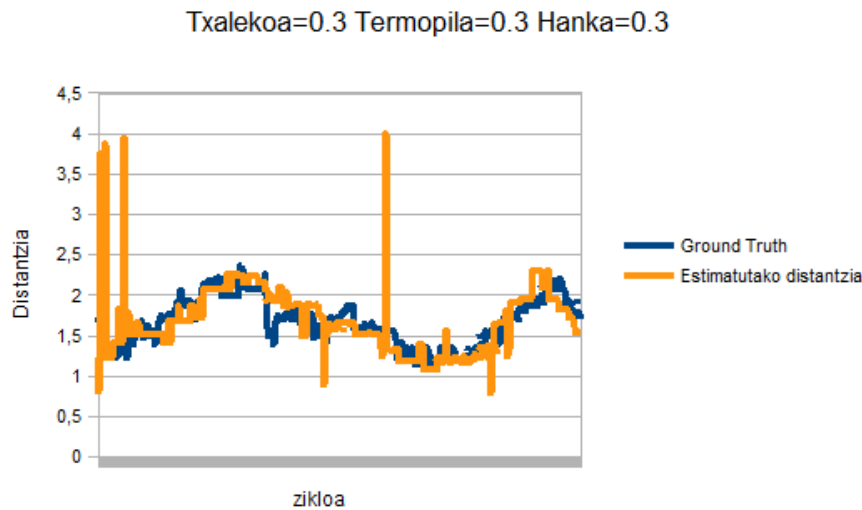
Distantziaren estimazioa, grafiketan ikus daitekeen bezala zuzena izan da eta bi kasuetan errore batzuk gertatu dira, txalekoa=0.3, termopila=0.3 eta hanka=0.3 kasuan erroreak nabarmenagoak dira.

[5.41.](#) irudian dagoen grafikoan, lortutako erroreak konparatu dira.

Grafikoan ikusten da bi kasuetan errore puntual handiak daudela baina, orokorrean, estimazioak onak izan dira.



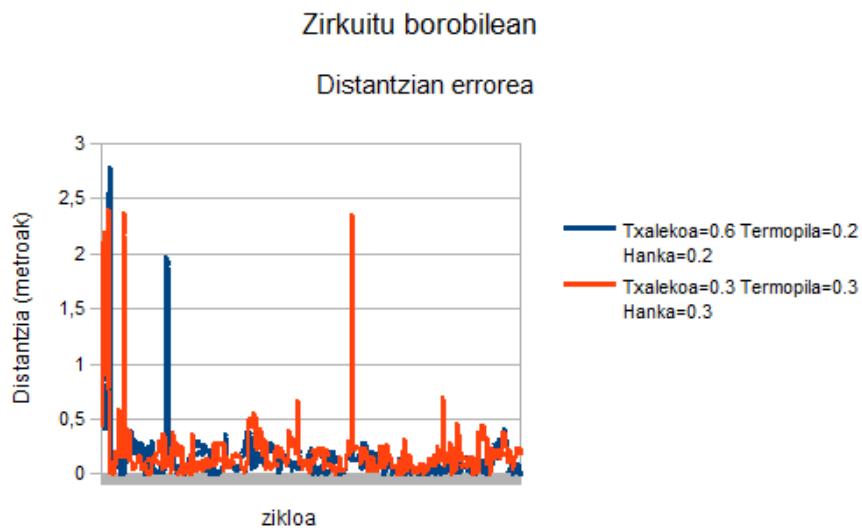
5.39. Irudia: Zirkuitu borobilean kalkulaturako distantziaren eta benetako balioen arteko erlazioa, txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasuan.



5.40. Irudia: Zirkuitu borobilean kalkulaturako distantziaren eta benetako balioen arteko erlazioa, txalekoa=0.3, termopila=0.3 eta hanka=0.3 kasuan.

Ebaluazio kualitatiboa

Proba hauek egiteko zirkuitu borobil bat erabili da. Aipagarria da, bi kasuetan jarraipena ondo egin duela robotak, nahiz eta kurba itxiak hartzeko pertsonak erritmoa jaitsi behar izan duen bi kasuetarako.



5.41. Irudia: Zirkuitu borobilean distantziaren estimazioan izandako errorearen konparaketa.

Proben ondorioak

Proba hauetatik aondorioztatzen, robotak kurba itxiak hartzeko arazoak izan dituela, txalekoa=0.6, termopila=0.2 eta hanka=0.2 kasuan partikula-filtroaren estimazioak hobeak izan dira, bai angelu, bai distantziarentzat, eta horrek robotak kasu horretan pixka bat fidagarriagoa izatea eragin du, batez ere biraketak egiterako orduan. Helburu pertsona robotaren ikusmen eremutik aldentuz gero robota saiaten da estimazioak egiten bere bila, baina helburu pertsona ez badu denbora labur batean berriro aurkitzen, geratu egiten da, berriro detektatu bitartean.

5.3.8. Ondorioak

Proba hauetatik gauza asko ondorioztatu dira. Robotaren abiaduraren kasuan esan daiteke lortutako emaitzetan abiadura aldaketak eragin haundiegirik izan ez arren, jarraipen seguru eta eroso bat egiteko abiadura hau oso garrantzitsua dela.

Partikula-filtroaren kasuan berriz, emaitzek garbi erakutsi dute, partikula kopuru haundiagoarekin robotaren estimazioak hobeak direla.

Sentsoreen kasuan, ikusi da, fusioak emaitza hobeak ematen dituela sentsore bakoitzak bere aldetik lan eginda baino. Sentsoreen fusioa doitzeko ordea, proba asko egin behar dira, eta robot honen kasuan ikusi den bezala, ezberdintasun handia dago sentsore batetik beste batera.

Amaitzeko esan behar da robotak arazoak dituela biraketa itxiak egiteko eta horretan laguntzeko helburu pertsonak erritmoa jaitsi beharra dauka kurbetan, bestela robotak helburu pertsona galtzeko arriskua dago. Robotak helburu pertsona galtzen duenean, hau berriz ere aurkitzen saiatzen da egon behar lukeen posizioa estimatuz, baina ez badu aurkitzen geratu egiten da, bere ikusmen eremuan berriro ere detektatu arte. Hau izan daiteke etorkizunean robotean hobetu daitekeen ataletako bat.

5.4. Proba 3

Atal honetan egin diren probetan IK4-TEKNIKER eko langile ezberdinek hartu dute parte eta proba egin aurretik eta ondoren galdetegi txiki bat bete dute beraien inpresioak azalertzeko. Hau egiteko Ktbot robota erabili da, aurreko probetatik ondorioztatu den konfigurazio hobereana ipini zaio eta robot honen ezagutzarik ez zuten lankide batzuei jarraipena egin die. Jarraipen hauek banaka egin dira eta minutu inguruko iraupena eduki dute, ondoren bakoitzak galdetegi bat bete du.

5.4.1. Helburua

Proba hauen helburua robota maneiatu ez duen jendeak eta bere funtzionamendua ezagutzen ez duenak, jarraipena egiten dionean izan ditzakeen sentrazio, uste edo zalantzak biltzea izan da, etorkizun batetan softwarea hobetzeko lagungarri izango diren asmoarekin.

5.4.2. Ingurunea

Proba hauek IK4-TEKNIKER eko robotika laborategian egin dira eta partaide bakoitza minutu inguru ibili da robota atzetik zuela, partaide bakoitzak joan etorriko ibilbide bat egin du bakoitzak nahi zuen moduan.

5.4.3. Galdetegia

Jarraian dagoen taulan, [5.9.](#) taulan, parte hartzaile bakoitzak bete duen galdetegia ikus dezakegu, bertan ikus daitekeen bezala galdera batzuk proba egin baino lehenago erantzun beharrekoak izan dira, partaidearen aurreiritzia jakin nahian. Beste galderak, berriz, proba egin ondoren erantzun beharrekoak izan dira, izandako sentrazioak adierazteko.

		Galdera	Erantzuna
1	Datu pertsonalak	Adina	
		Sexua	
		Lanbidea	
2	Proba egin aurreko ustea	Roboten erabilera ideia ona dela deritzozu?	Bai Ez
		Robotaren portaera fidagarria iruditzen zaizu?	Bai Ez
3	Proba egin eta gero	Egindako jarraipenak utzitako sentsazioa.	(Ona----Txarra) 5 4 3 2 1
		Jarraipena egiten ari zen bitartean izandako erosotasuna.	(Ona----Txarra) 5 4 3 2 1
		Distantzia egokia mantentzen du?	Handiegia / Egokia / Txikiegia
		Abiadura egokia mantentzen du?	Handiegia / Egokia / Txikiegia
		Espero baina hobeto funtzionatzen du?	Bai Ez
		Jarraipenean hobetuko zenituzkeen gauzak	

5.9. Taula: Proba egin dutenek erantzundako galdetegia.

5.4.4. Emaitzak

Jarraian, proban parte hartu duen jendearen erantzunak ikusi daitezke. Erantzunak ulergarriago egiteko multzokatu eta estatistikak atera dira. Galdetegiaren formatuari jarraituz 3 multzotan banatu dira: datu pertsonalak, proba egin aurreko galderak eta proba egin eta geroak.

Datu pertsonalak:

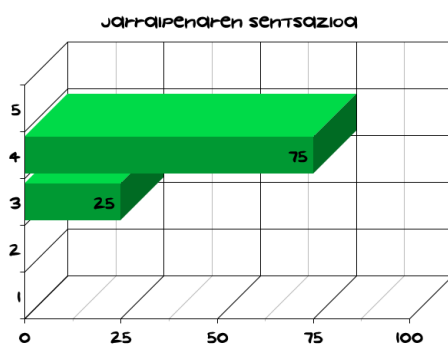
- Proba egin dutenen adin tartea 27 eta 32 urte artekoa izan da.
- Ehuneko 75-a gizonezkoa izan da eta ehuneko 25-a emakumezkoa.
- Guztiak IK4-TEKNIKER eko ikertzaileak izan dira.

Proba egin aurretik:

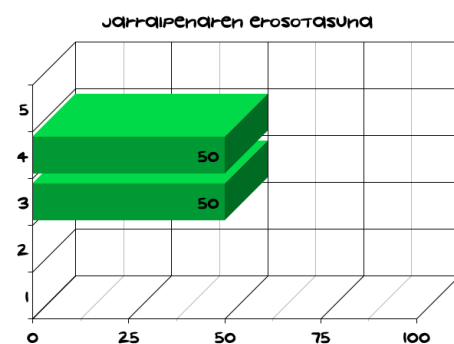
- Proba egin duten guztiak, %100-a, ados egon dira robotaren erabilera ideia ona dela galdetutakoan.
- Guztiek, %100-ak, uste du robota fidagarria dela.

Proba egin eta gero:

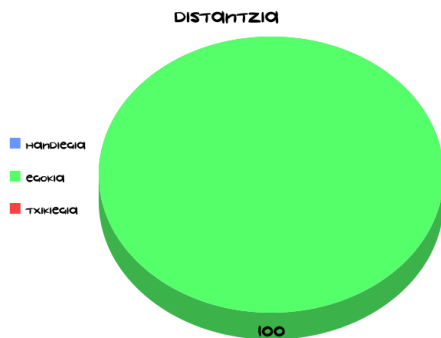
- Jarraipena egin bitartean izandako sentsazioa eta erosotasuna 5 etik 1 era kalifikatu behar izan da, onetik txarrera. Sentsazioari %75-ak 4-ko bat eman dio eta %25-ak 3-koa. Erosotasunari berriz, %50-ak 4-koa eta gainontzeko %50-ak 3-koa. 5.10. eta 5.11. grafikoetan aipatutako emaitzak ikus daitezke.



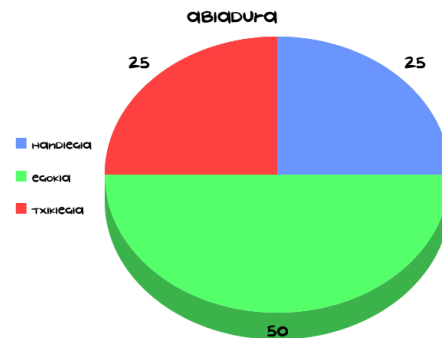
5.10. Taula: Jarraipenaren sentsazioa.



5.11. Taula: Jarraipenaren erosotasuna.



5.12. Taula: Robotaren distantzia.



5.13. Taula: Robotaren abiadura.

- Robotak mantentzen dituen distantzia eta abiaduraren inguruko galderetan 3 aukera daude: handiegia, egokia eta txikiegia. Distantziaren kasuan %100-ak egokia dela iritzi dio. Abiaduraren kasuan berriz, %50-ak egokia, %25-ak handiegia eta %25-ak txikiegia. 5.12. eta 5.13. grafikoetan, aipatutako emaitzak ikus daitezke.
- Proba egin duenaren espektatibak bete diren galderan, %50-ak baiezkoa erantzun du eta gainontzeko %50-ak ezezkoa. Jarraian dago grafikoa, 5.14. .



5.14. Taula: Espektatibaren inguruko galdera.

- Azken galderan parte hartzaileari bere iritzia eskatu zaio, robotaren ahulgune nagusiak identifikatu eta hobekuntzak non egin beharko liratekeen erabakitzeko. Erantzun gehienak hildo beretik joan dira, robotak biraketa azkarrak egiteko duen arazoa azpimarratu dute. Horrez gain, robota jarraitzen ari den pertsona ikusmen-eremutik kanpo geratzen zaionenan

(hau gertatzearen arrazoi nagusia biraketen abiadura motela da) sortzen den arazoa ere aipatzen dute, denbora txiki bat igaro eta gero robota geratu egiten baita. Honi aurre egiteko bilaketa estrategia bat erabiltzea proposatu dute parte-hartzaileek.

5.4.5. Ondorioak

Proba honen helburu nagusia jendeak robotarekiko duen pertzepzioa ezagutzea izan da, robotak dituen ahulguneak ezagunak baitira aurretik egin diren probetatik ondorioztatu direlako.

Galdetegiaren emaitzei begiratuta ondoriozta daiteke orokorrean robotak distantzia egokia mantentzen duela helburu pertsonarekiko eta abiadura ere nahikoa zuzena dela, nahiz eta hemen eztabaida gehiago egon daitekeen, beti ere robotak biraketa zuzenik egin behar ez duenean.

Robota norbaiti jarraitzen ari zaionean pertsona hori nahiko eroso sentitu dela ondoriozta daiteke eta jarraipenaren kalitatea ona dela, hau da, ez dutela jazarpen moduko zerbait sentitu robota atzetik ibili dutenean.

Jakina da robotikan jendearen espektatibak betetzea ez dela batere erreza, jendeak robot humanoide baten jarrera gizaki batenarekin konparatzen baitu, baina askotan ez gara konturatzen gizaki batek oharkabean edo instintiboki egingo lituzkeen zenbait akzio robot batentzat ezinezkoak direla ez baitu gizaki batek duen "adimen" hori. Hau jakinda, galdeketa ikusi den moduan jendearen espektatibak guztiz ez betetzea ez da harrizkeoa.

Jendeak utzitako iritzien inguruan, esan behar da aurretik jakinak zirela gehienak, aurretik egindako probetan ahulgune berak aurkitu baitzaizkio robotari eta, batez ere, ibilbide aldaketa bortitzak egitean izaten ditu arazoak jarraitzen ari den pertsonak ikusmen-eremutik ihes egiten baitu eta robota zer jarraitu ez dakiela geratzen baita. Horri aurre egiteko, parte-hartzaile batek proposatu bezala, interesgarria litzateke bilaketa estrategia bat definitu eta helburu puntua galtzean martxan jartzea, etorkizun batean garatu beharreko atal bat izango da.

Bestalde, esango nuke pozgarria dela jendeak erabili den robotarekiko eta orokorrean robotekiko erakutsi duen aurreiritzia. Galdetegiaren arabehera jendeak positibo ikusten du roboten erabilera eta fidagarriak iruditzen zaizkio. Uste dut hau gaur egungo gizartean dugun egoerarekin bat datorrela, geroz eta ohikoagoak dira robotak gure ingurunean eta lan ezberdinak egiteko edo laguntzeko erabiltzen ditugu. Iruditzen zait gizaki eta roboten arteko hurbiltasuna lortzen ari garela baina robot eta gizakien arteko elkarbizitza seguru baten alde lan handia dago oraindik egiteko.

6. Kapituluia

Ondorioak eta etorkizuneko lana

Gaien Aurkibidea

6.1. Sarrera	104
6.2. Nabigazioa robot mugikorretan	104
6.3. Pertsonen jarraipena robot mugikorretan	104
6.4. Etorkizuneko lana	105

6.1. Sarrera

Atal honetan proiektuan zehar egindako lana laburtuko da, bertatik atera diren ondorioak azalduko dira eta etorkizunean, ebaluatu den sistema hobetzeko har daitezkeen neurrien iradokizunak egingo dira.

Ekarpena robot mugikorren nabigazioaren inguruan eta pertsonen jarraipenaren inguruan egingo da.

6.2. Nabigazioa robot mugikorretan

Lan honetan robot mugikorren nabigaziorako zenbait teknika deskribat dira. Robot batek bere burua lokalizatzeko erabiltzen diren teknikak: partikula-filtroa edo kalmanen filtroa. Horrez gain, robot batek hasiera puntu batetik, helburu puntu batera joateko egin behar duen bidea planifikatu eta egiten dituen mugimenduak zuzenak direla kontrolatzeko erabiltzen diren teknikak ere deskribatu dira.

Robotaren nabigazioaren puntu kritikoa robotaren lokalizazioa da. Robotak, denbora guztian, bere burua ingurune batean non dagoen jakin behar du, eta horrek zailtasun handiak ditu. Ingurunearen mapa eguneratu bat edukitzea ezinbestekoa da eta, gainera, mapa horretan kokatuko duen sistema ere beharko du. Pentsa daiteke egungo GPS sistemekin kokapen hori lor daitekeela, baina ez da egia, sistema hauek 2-10 metro arteko erroreak izan baititzakete eta hori onartezina da. Horregatik beste sistema batzuk erabili behar dira. Lan honetan, hau teorikoki aztertu den arren, roboten nabigazioan lanean ibili diren lankideekin hitz egiteko aukera ere izan da. Beraien proiektuak ikusi eta izan dituzten arazoez hitz egitean, garbi dute, lokalizazioa dela nabigazioan buruhauste gehien sortzen duen atala.

6.3. Pertsonen jarraipena robot mugikorretan

Lan honetan pertsonen jarraipena egiteko sistema bat deskribatu eta ondoren bere funtzionamendua ebaluatu da. Proposaturiko sistemak sentzore ezberdinetatik jasotako informazioa fusionatzen du partikula-filtro bat erabiliz, helburu pertsonaren posizioaren estimazioak egin eta jarraipena burutzeko.

Sistemeekin egindako esperimentazioan ondorioztatu da, laserra eta termopila soilik informazio iturri moduan hartzea ez dela ideia ona jarraipena egiteko. Kamera, laserra eta termopila erabiliz, ordea, bakoitzari garrantzi egokia emanda, sistema sendo eta fidagarri bat lortu daitekeela ere ikusi da.

Robotaren ahulgune nagusia biraketa itxiak hartzeko unean dagoela ere ondorioztatu da esperimentazioan. Kasu horretan gertatzen dena da, robotak jarraitzen ari den pertsona bere ikusmen-eremutik galtzen duela, robotak pertsona batek baino denbora gehiago behar baitu biraketak egiteko. Robotak helburu pertsona galtzen duenean geratu egiten da, nora joan ez dakiela, berriro ere bere ikusmen-eremuan pertsona detektatu bitartean.

Hau guztia kontuan izanda, gizakiei erreskate lanetan laguntzeko diseinatutako sistema honek, hobekuntza batzuk beharko lituzke gizakiekin elkarlanean modu seguru batean lan egin ahal izateko. Hurrengo atalean hobekuntza posible batzuk aipatuko dira.

6.4. Etorkizuneko lana

Behin memoria honetan aurkeztu den lana amaituta eta egindako esperimentazio-tik ondorioak aterata, ikertu daitezkeen bide ezberdinak zabaldu dira:

- Robotaren nabigazioaren inguruan arlo ezberdinak uztartu dira, baina SLAM (*Simultaneous localization and mapping*) sistemak ez dira landu. Sistema hauek robotek ingurunearen maparik ez daukaten lekuetan lan egitea ahalbidetzen dute, robotaren lokalizazioa eta maparen sorrera batera eginez. Interesgarria litzateke sistema hauen funtzionamendua aztertu eta azaltzea.
- Ebaluatu den robotean ere hainbat hobekuntza egin litezke. Esperimentazioan ondorioztatu den bezala, robotaren ahulgune nagusia biraketa itxietan helburu pertsona galdu eta robota noraezean gera daitekeela da. Hau ekiditeko bilaketa sistema bat inplementatzea aukera bat litzateke. Robotak helburu pertsona galtzen duenean zein aldetatik galdu duen jakinda eta bilaketa estrategia bat erabiliz arazo hau konpon liteke.
- Beste hobekuntza posible bat robotean nabigazio sistema bat txertatzea litzateke. Helburu pertsonara abiatu aurretik bide egokiena kalkulatu eta bidean egon litezkeen oztopoak ekiditen lagunduko lukeena. Bigarren hobekuntza hau konplexuagoa da eta seguruenik robotean sentsoire gehiago jartzea behartuko luke.

Bibliografía

- [1] Simon Godsill Arnaud Doucet and Christophe Andrieu. On sequentian monte carlo sampling methods for bayesian filtering. *Signal Processing Group, Department of Engineering, University of Cambridge*, 2000.
- [2] Arnaud Doucet and Adam M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *The Institute of Statistical Mathematics, Tokyo and Department of Statistics, University of Warwick*, 2004.
- [3] Patrick Goebel. *ROS By Example*. Lulu, 2013.
- [4] Jeroen D. Hol, Thomas B. Schön, and Fredrik Gustafsson. On resampling algorithms for particle filters. *Division of Automatic Control, Department of Electrical Engineering, Linköping University Sweeden*, 2006.
- [5] Aaron Martinez and Enrique Fernández. *Learning ROS for Robotics Programming*. Packt Publishing, September 2013.
- [6] Maikel O. Torres Pineiro and Valery Moreno Vega. Un estudio de dos métodos de planificación de trayectorias en entornos estáticos. *Instituto Superior Politécnico J.A. Echeverría, Ciudad de La Habana, Cuba*, 2009.
- [7] Loreto Susperregi, Jose Maria Martinez-Otzeta, Ander Ansuategui, Aitor Ibar-guren, and Basilio Sierra. Rgb-d, laser and thermal sensor fusion for people following in a mobile robot. *Autonomous Smart Systems Unit, IK4-TEKNIKER, Spain*, 2013.
- [8] Greg Welch and Gary Bishop. An introduction to the kalman filter. *Department of Computer Science, University of North Carolina*, 2006.